efrei

PARIS PANTHÉON-ASSAS UNIVERSITÉ

TP01 Infrastructures Cloud

Par David TEJEDA, Vincent LAGOGUE, Tom THIOULOUSE, Thomas PEUGNET, Alexis PLESSIAS

I3 – 2024-2025

Table des matières

Questions préliminaires

Outillage

1. **Expliquez la fonction remplie par *jq*, donnez un exemple.**

Jq est un outil en ligne de commande utilisé pour traiter des données sous le format JSON. Il permet de :

- Lire des données JSON
- Filtrer une structure pour récupérer ce qui nous intéresse
- Combiner des données JSON

Par exemple si nous avons notre fichier JSON *data.json* qui contient les données de M.Plexiglass :

```
{
    "nom": "Plexiglass",
    "age": 29,
    "activites": ["Course_a_pied", "Cuisine", "Lecture"],
    "localisation": {
      "ville": "Limoge",
      "pays": "France"
    }
}
```

On rend le fichier plus lisible grâce à la commande *:*

```
jq . data.json
```

On obtient ce résultat :

```
{
    "nom": "Plexiglass",
    "age": 29,
    "activites": [
      "Course_a_pied",
      "Cuisine",
      "Lecture"
    ],
    "localisation": {
      "ville": "Limoge",
      "pays": "France"
    }
  }
```

Une fois le formatage du fichier réalisé on peut extraire des informations du fichier.

Par exemple, on peut récupérer le nom de l'utilisateur :

```
jq '.nom' data.json
$> "Plexiglass"
```

On peut également récupérer des informations plus précises en précisant l'arborescence du fichier. Par exemple, pour récupérer la ville de *Plexiglass* :

```
jq '.localisation.ville' data.json
$> "Limoge"
```

Jq permet beaucoup de fonctionnalités et est un outil très intéressant lorsqu'il s'agit de traiter des fichiers JSON.

2. **Expliquez la fonction remplie par *yq*, donnez un exemple.**

Yq est un outil similaire à *jq* mais qui permet de manipuler des fichiers YAML. Ainsi l'avantage de cet outil est qu'il est également capable de traiter les fichiers JSON, on peut donc le voir comme une version évoluée de *jq.*

Imaginons maintenant que nous avons toujours notre fichier JSON avec les informations de Plexiglass. Pour le passer du format JSON au format YAML il nous suffit de faire la commande :

```
yq . data.json
```

Et nous aurons comme résultat :

```yaml
nom: Plexiglass
age: 29
activites:
  - Course_a_pied
  - Cuisine
  - Lecture
localisation:
  ville: Limoge
  pays: France
```

Nous avons donc un fichier YAML bien construit. Les commande de base de *yq* vont être similaire à celle de *jq*.

On peut voir comme autre fonctionnalité la possibilité de modifier les informations du fichier. Si on veut ajouter un champ « age » avec la valeur « 30 » on va pouvoir mettre à jour notre fichier avec la commande :

```
yq '.age = 30' data.json
```

La sortie sera donc mise à jour :

```
nom: Plexiglass
age: 30
activites:
  - Course_a_pied
  - Cuisine
  - Lecture
localisation:
  ville: Limoge
  pays: France
```

3. **Expliquez trois avantages d'utiliser *tmux* lors de connexions SSH, notamment en travaux de groupe.**

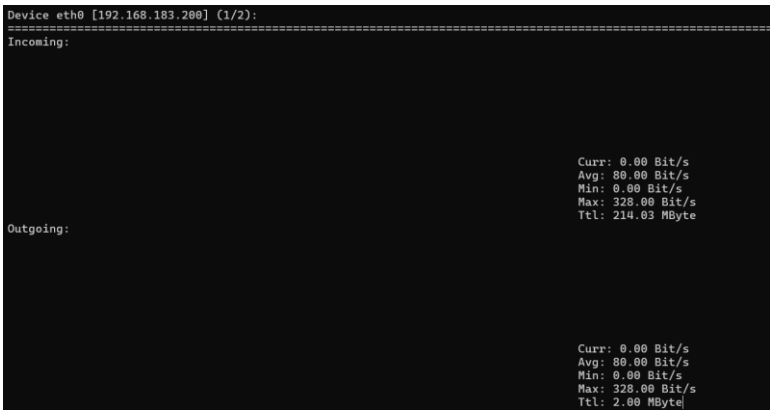
Il existe plusieurs avantages à l'utilisation de *tmux* pour des connexions SSH :

- Il permet d'avoir accès à plusieurs terminaux virtuels sur une seule fenêtre et donc d'éviter d'ouvrir plusieurs terminaux.
- On peut naviguer et créer facilement de nouvelles sessions
- Les sessions sont persistantes, elles restent actives même après une interruption SSH.

Cet outil est très similaire à *Terminator* mais le fait qu'il propose des sessions SSH persistante le rend bien plus intéressant dans notre cas.

4. **Expliquez la fonction remplie par *nload*, donnez un exemple.**

nload est un outil en ligne de commande pour surveiller la bande passante réseau en temps réel. Il affiche les débits d'entrée et de sortie pour chaque interface réseau.



On a un rapport en temps réel des ressources réseaux utilisées.

Fiche d’architecture fonctionnelle

1. Explications textuelles du fonctionnement et des liens

Services rendus par la DSI de l'EFREI :

La DSI de l’EFREI fournit les services nécessaires au bon fonctionnement de l’infrastructure. Elle propose un accès réseau sécurisé via un *VPN*, permettant aux utilisateurs d’accéder aux machines virtuelles. Elle gère également l’hébergement des ressources sur une plateforme cloud, en provisionnant les machines virtuelles nécessaires (Trois dans notre cas). Enfin, la DSI centralise la *gestion des identités* pour authentifier et autoriser les utilisateurs tout en assurant la supervision et la maintenance de l’infrastructure.

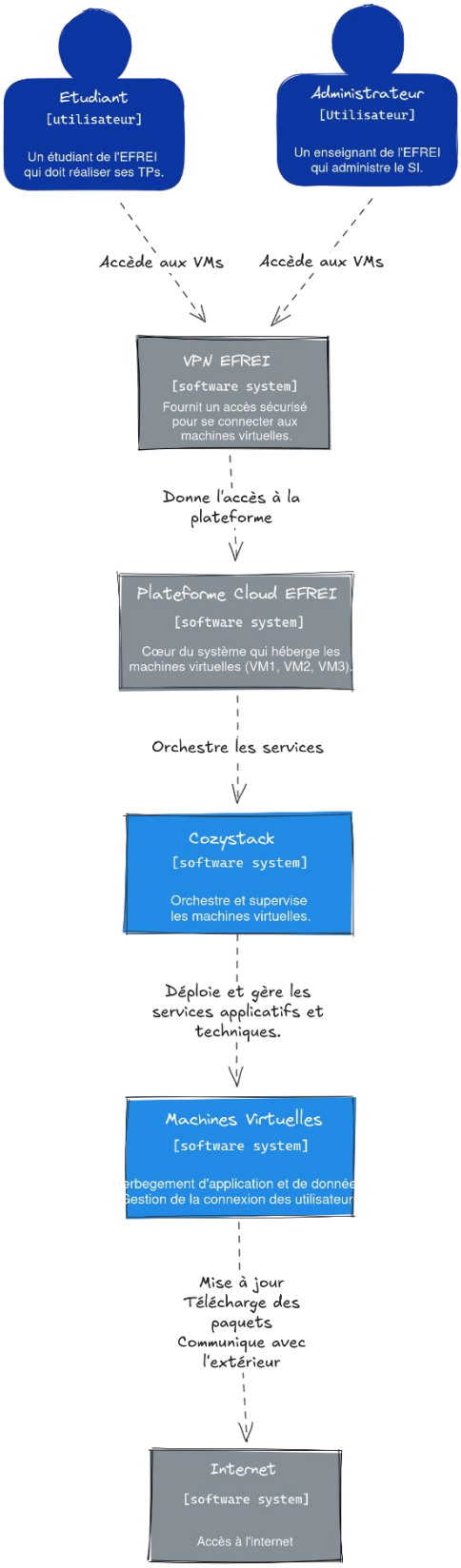
Services rendus par le gestionnaire de clés et le gestionnaire d’identités :

Le *gestionnaire de clés* ***(PKI)*** permet une gestion sécurisée des accès en fournissant des clés SSH aux utilisateurs pour se connecter aux machines virtuelles, cela permet d’éliminer les risques liés aux mots de passe faibles. Le *gestionnaire d’identités* ***(IDP)*** assure l’authentification des utilisateurs et définit leurs permissions pour garantir un contrôle centralisé des accès. PKI et IDP sécurisent donc les connexions et facilitent également la gestion des droits dans l’infrastructure.

Valeur ajoutée du Cozystack :

Cozystack simplifie la gestion des environnements cloud pour les développeurs, plus particulièrement pour la gestion des ressources et l’orchestration des services. Il permet de déployer des applications dans des environnements de conteneurs de manière transparente, tout en intégrant des outils d'automatisation et de collaboration.

2. Schéma d’architecture fonctionnelle



Fiche d’implémentation

1. Explications textuelles du fonctionnement et des liens

Espaces contigus d’adresses IP de publication de services

Les espaces contigus d’adresses IP de publication de services sont :

172.16.18.36

172.16.19.27

172.16.19.29

Informations sur les machines

On vérifie les adresses IP des clients et leur capacité. Pour la première machine :

```
root@1124BUBUSTD25:/home/studentlab# ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b6:16:58 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 172.16.18.36/23 metric 100 brd 172.16.19.255 scope global dynamic ens33
       valid_lft 6126sec preferred_lft 6126sec
    inet6 fe80::250:56ff:feb6:1658/64 scope link
       valid_lft forever preferred_lft forever
3: tailscale0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1280 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 100.127.217.69/32 scope global tailscale0
       valid_lft forever preferred_lft forever
    inet6 fd7a:115c:a1e0::aa01:d947/128 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::226a:5aeb:b85:cfb3/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

On lance un **htop** sur les clients pour vérifier ses capacités.

```
  0[||                                                    1.1%] Tasks: 55, 84 thr, 148 kthr; 1 running
  1[|                                                    0.4%] Load average: 0.07 0.02 0.00
Mem[||||||||||||||||||||||||||||||||||||||||||||||||  449M/7.76G] Uptime: 7 days, 10:51:06
Swp[                                                     0K/4.00G]
```

Pour la deuxième machine :

```
root@1124BUBUSTD26:/home/studentlab# ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group def
ault qlen 1000
    link/ether 00:50:56:b6:a6:9a brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 172.16.19.27/23 metric 100 brd 172.16.19.255 scope global dynamic ens33
       valid_lft 18801sec preferred_lft 18801sec
    inet6 fe80::250:56ff:feb6:a69a/64 scope link
       valid_lft forever preferred_lft forever
3: tailscale0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1280 qdisc fq_codel
 state UNKNOWN group default qlen 500
    link/none
    inet 100.98.204.70/32 scope global tailscale0
       valid_lft forever preferred_lft forever
    inet6 fd7a:115c:a1e0::fd01:cc47/128 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::3bb0:5a1b:4fa6:2436/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

```
  0[|||                      5.9%] Tasks: 56, 84 thr, 160 kthr; 1 running
  1[|||                      4.0%] Load average: 0.07 0.06 0.06
  2[                         0.0%] Uptime: 7 days, 09:28:19
  3[|                        0.7%]
Mem[||||||||||||||||     474M/7.75G]
Swp[                       0K/4.00G]
```

Enfin pour la dernière machine :

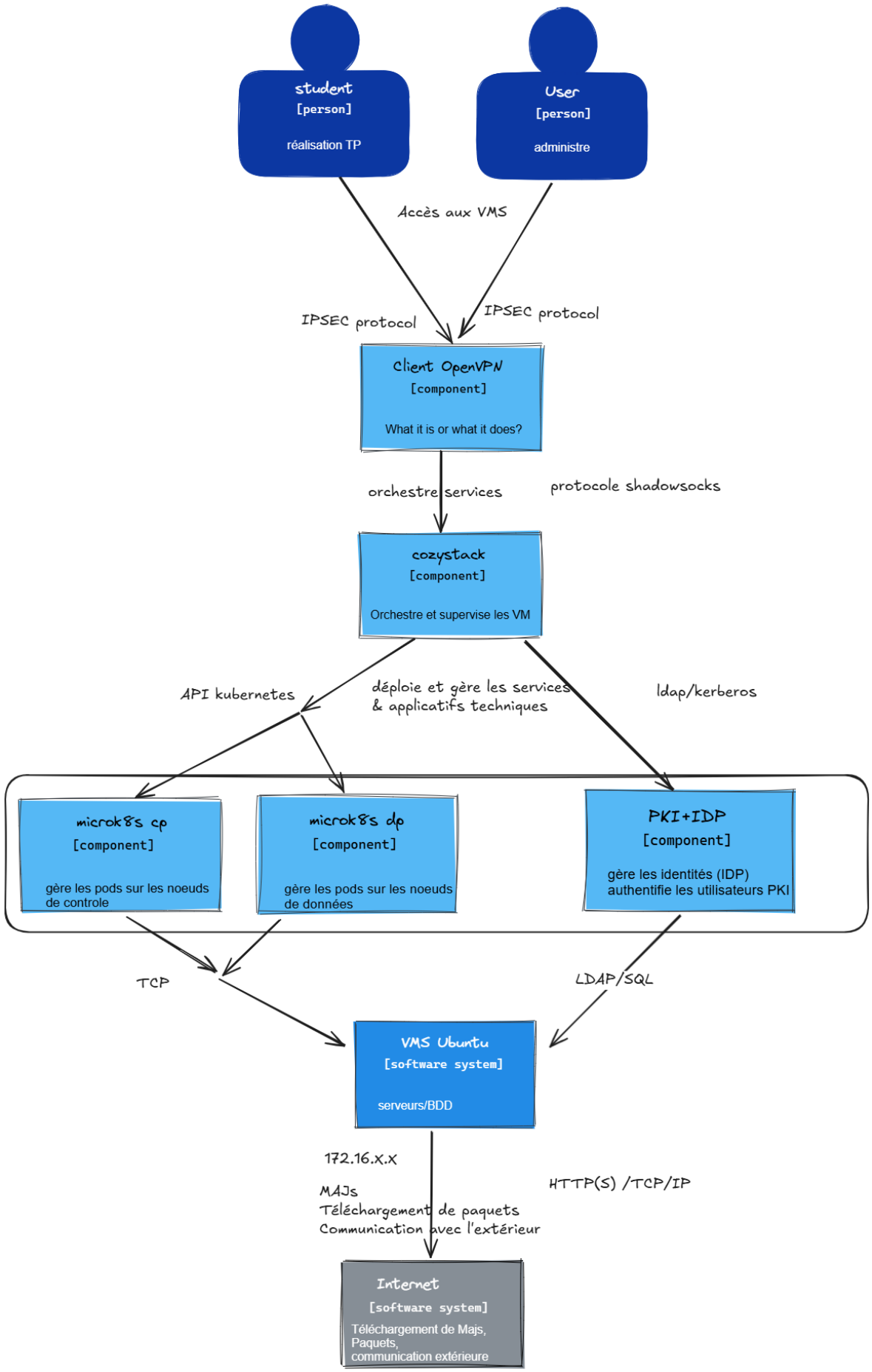
```
root@1124BUBUSTD27:/home/studentlab# ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:50:56:b6:44:9b brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 172.16.19.29/23 metric 100 brd 172.16.19.255 scope global dynamic ens33
       valid_lft 8638sec preferred_lft 8638sec
    inet6 fe80::250:56ff:feb6:449b/64 scope link
       valid_lft forever preferred_lft forever
3: tailscale0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1280 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 100.79.188.4/32 scope global tailscale0
       valid_lft forever preferred_lft forever
    inet6 fd7a:115c:a1e0::7c01:bc04/128 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::4827:42cc:a221:4ba9/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

On voit que la machine est plus puissante :

On regarde aussi le stockage qui est le même que les autres :

```
root@1124BUBUSTD27:/home/studentlab# df -h
Sys. de fichiers                 Taille Utilisé Dispo Uti% Monté sur
tmpfs                              1,6G    1,9M  1,6G   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv   95G     12G   79G  13% /
tmpfs                              7,9G       0  7,9G   0% /dev/shm
tmpfs                              5,0M       0  5,0M   0% /run/lock
efivarfs                           256K     63K  189K  25% /sys/firmware/efi/efivars
/dev/sda2                          2,0G    190M  1,6G  11% /boot
/dev/sda1                          1,1G    6,2M  1,1G   1% /boot/efi
tmpfs                              1,6G    108K  1,6G   1% /run/user/1000
```

2. Schéma d’architecture technique


student
[person]
réalisation TP
User
[person]
administre
Accès aux VMS
IPSEC protocol
IPSEC protocol
Client OpenVPN
[component]
What it is or what it does?
orchestre services
protocole shadowsocks
cozystack
[component]
Orchestre et supervise les VM
API kubernetes
déploie et gère les services
& applicatifs techniques
ldap/kerberos
microk8s cp
[component]
gère les pods sur les noeuds
de controle
microk8s dp
[component]
gère les pods sur les noeuds
de données
PKI+IDP
[component]
gère les identités (IDP)
authentifie les utilisateurs PKI
TCP
LDAP/SQL
VMS Ubuntu
[software system]
serveurs/BDD
172.16.x.x
MAJs
Téléchargement de paquets
Communication avec l'extérieur
HTTP(S) /TCP/IP
Internet
[software system]
Téléchargement de Majs,
Paquets,
communication extérieure