

Mini Projet de synthèse

I. Objectifs

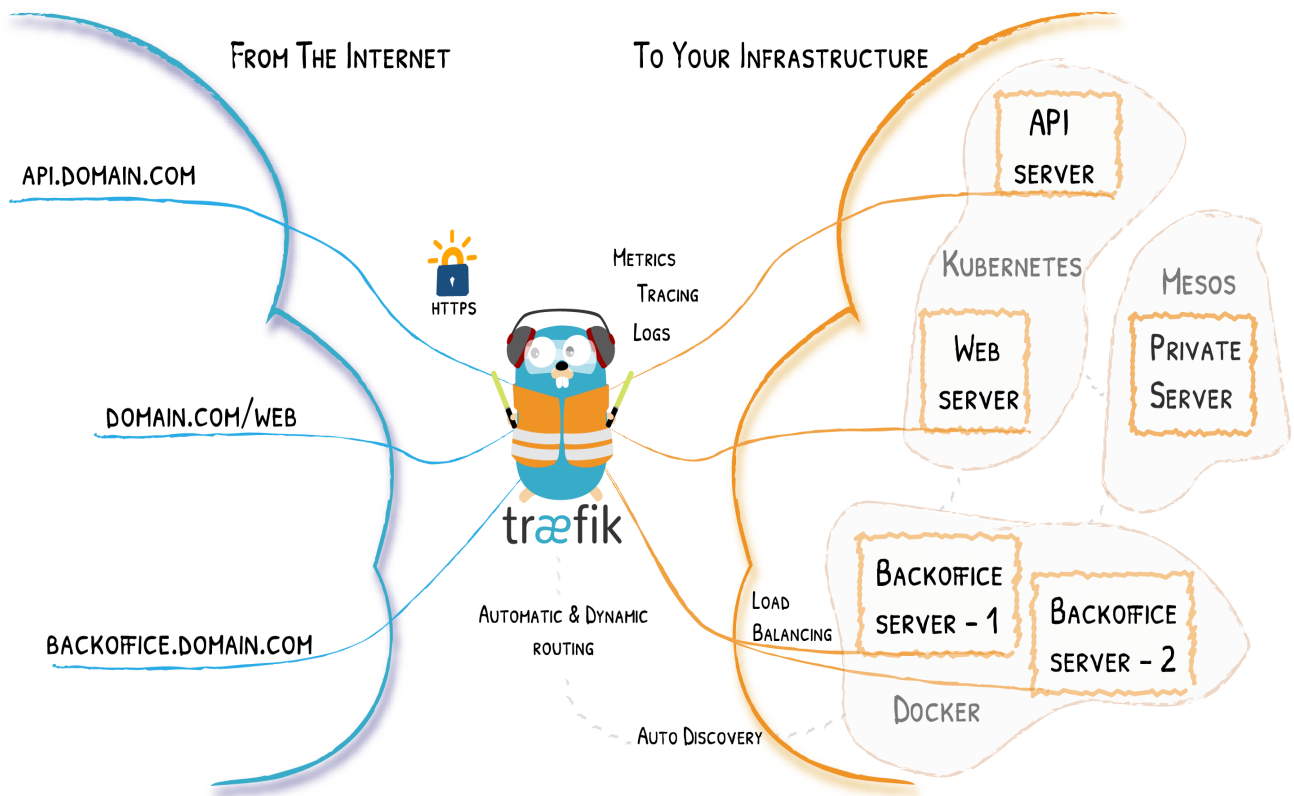
L'objectif de ce mini projet est de mettre en place sur votre cluster swarm plusieurs services accessibles via un seul point d'entrée, le port 80 du manager de votre cluster. À cette fin, vous devrez mettre en place des fichiers de configuration swarm, docker compose, dockerfile pour déployer facilement vos services. Il s'agira de déployer des services que vous avez abordés en TD/TP. Le point d'entrée sera un ingress. Attention tous ceci est gourmand en ressources il vous faudra prendre soin de limiter/paramétrer les ressources allouées aux services.

II. Point d'entrée

notion d'ingress:

Un ingress (ou une entrée réseau) expose les routes HTTP et HTTPS de l'extérieur du cluster à des services au sein du cluster. Le routage du trafic est contrôlé par des règles définies sur la ressource ingress.

[Internet] <-> [Ingress] <-> [[Service1] , [Service2] , [Service3] , ...]



Un ingress peut être configuré pour donner aux services des URLs accessibles de l'extérieur, un équilibrage du trafic de charge externe, la terminaison SSL/TLS et un hébergement virtuel basé sur le nom.

Vous utiliserez en guise de point d'entrée l'ingress Traefik. Voir: [Doc Traefik](#)

Vous en donnerez les fichiers de configuration afin de le lancer avec swarm. Il donnera accès aux services suivants qui tournent tous en même temps.

Non obligatoire mais utile pour comprendre ce que l'on fait: utiliser le dashboard (de préférence de manière authentifiée avec basicauth, voir la fin du mini projet)

Au sujet de Traefik, voir:

- Le cours 2:
https://webem.iuto.ovh/coursvirtualisation/virtualisation_avancee_C3V0.html

Docker Compose

- Le cours 3:
https://webem.iuto.ovh/coursvirtualisation/virtualisation_avancee_C4V0.html

Docker Swarm

- Doc dashboard : <https://doc.traefik.io/traefik/operations/dashboard/>



The Dashboard

See What's Going On

The dashboard is the central place

III. Les Services

1. Site web de « Reprise de contact »

Il s'agit du site web en php/apache qui utilise mysql avec redis entre les deux. Le service affiche la liste des utilisateurs qui sont dans la base de donnée. Il vous faut distribuer le service web sur les trois noeuds du cluster swarm.

L'url de ce site web doit être: `http://utilisateurs.<nom de votre VM>`

2. Site web de « Cluster Swarm »

Reprenez l'application web/redis/mysql de cette partie et créez 5 instances de site web, 2 serveurs redis et placez 2 redis sur 2 volumes gluster. Le serveur mysql reste lui unique mais peu sollicité grâce à redis. Débrouillez-vous pour afficher autre chose que pour le site de la question précédente.

L'url de ce site web doit être: `http://clusterswarm.<nom de votre VM>`

3. Application Fortune Images

Vous trouverez les explications ici dans l'archive. Attention il faut optimiser l'image utilisée comme il l'est expliqué.

 [archive](#)

L'url doit être: `http://fortune.<nom de votre VM>`

4. Site web en flask pris dans la partie « Rappel de 2ième année... »

Ici c'est le site web en flask qui permet de peupler une base de donnée redis.

L'url doit être: `http://flask.<nom de votre VM>`

IV. Recherche perso

5. Ajoutez un site web apache simple mais qui est accessible via non par une url avec un sous domaine mais au contraire avec un chemin.

L'url doit donc être: `http://<nom de votre VM>/chemin`

V. Bonus

6. Ajouter une authentication basique (basicauth) pour l'accès à votre site web apache simple. Le site web demande un couple login/password. Vous devez utiliser Traefik pour gérer cela.

Voir: [Doc basicauth](#)

VI. Défi Authentication Commune

On veut protéger l'ensemble des services par un même login / mot de passe, sans que ceci soit précisé dans les labels du fichier de paramétrage du service. Ceci ouvre la voie à la création d'un serveur de sites web qui réclamerait une authentication basicauth systématique pour accéder aux services. Par exemple pour vos SAE il serait possible de les porter, dockerisées, vous-même sur un serveur sans vous préoccuper d'authentification mais que celle-ci soit exigée par l'ingress.

7. Trouver un paramétrage de traefik afin que tout service qui passe par lui réclame un mot de passe comme décrit ci-dessus.

Modalités de rendu

Vous rendrez une archive avec une arborescence avec vos fichiers de paramétrage ET aussi un court texte en markdown (et son équivalent pdf) expliquant ce que vous avez fait et ce qui vous a posé problème. Vous devez aussi et **impérativement** fournir en tête de ce fichier **les urls de vos services** sur vos serveurs proxmox. **ATTENTION** Votre archive doit être en tgz et créée en ligne de commande (l'interface graphique à l'IUT pose problème car elle omet certains fichiers). Donc utilisez une commande de type `tar -czf mon_repertoire.tgz mon_repertoire` et **vérifiez en décompressant votre archive pour voir si elle contient quelque chose**