



RAPPORT INTERMÉDIAIRE

NUAGE POUR SITES WEB

DÉCEMBRE 2015

THIBAUT SCHOLAERT

JÉRÉMIE DENÉCHAUD

Table des matières

Introduction

Cahier des charges

Travail réalisé

Choix technologiques

Première couche virtuelle : Docker

Couche de supervision

Deuxième couche virtuelle : Xen

Un nouveau serveur

Benchmark de performance

Réseau

Prochains objectifs

Architecture réseau

Fonctionnalités

Maquette du site

Conclusion

Introduction

Dans le cadre de notre projet de dernière année à l'école d'ingénieur Polytech Lille, nous avons l'opportunité de travailler sur une solution info-nuagique pour créer et héberger les sites web de l'école.

L'objectif est de permettre à tout étudiant, association ou personnel encadrant de créer facilement un site web sur les serveurs de l'école. L'architecture utilisée doit être légère, robuste et documentée pour remplacer, à terme, l'infrastructure existante.

Le cahier des charges précis est développé dans une première partie. Ensuite nous aborderons le travail effectué jusqu'à présent et enfin nous présenterons nos prévisions pour la suite du projet.

Cahier des charges

Le but du projet est de mettre en place une infrastructure info-nuagique permettant de créer rapidement et simplement un site web pour un utilisateur n'ayant pas nécessairement de connaissance en informatique.

La solution recherchée doit pouvoir remplacer celle existante et donc au moins supporter la charge actuelle des sites existants. Plus précisément, les serveurs hébergent actuellement environ 2000 sites web dont 200 sont considérés comme actifs. Le matériel à utiliser n'est pas imposé, ni défini. Nous nous efforçons donc de garder une infrastructure qui s'adapte à différentes échelles. Le choix des technologies à utiliser est libre mais l'architecture finale doit remplir des contraintes de légèreté et de performance.

L'accès à chaque serveur doit se faire à travers une unique adresse IPv4 ou IPv6 routée. L'utilisateur doit pouvoir gérer son site et y ajouter du contenu avec au moins autant de facilité que ce que propose le système actuel (FTP). La gestion des certificats pour un accès en HTTPS doit aussi être prise en compte. Différents modules doivent être mis à la disposition de l'utilisateur comme par exemple un module php. Le nom du site est fixé par l'utilisateur et doit être implémenté très simplement.

Travail réalisé

Choix technologiques

Première couche virtuelle : Docker

Concernant nos choix technologiques, l'idée initiale était l'utilisation d'une technologie récente utilisant la conteneurisation : Docker. Cette technologie offre une grande légèreté dans son utilisation mais également dans son déploiement. Dans ses applications on peut trouver des serveurs Web tel qu'Apache ou Nginx, des modules additionnels (PHP, Ruby on rails, etc) mais aussi des solutions de serveur virtuels complets (OVH). Cette solution proposée en open source nous a semblé un choix pertinent pour embarquer un serveur Web dans un conteneur virtuel.

Nous sommes néanmoins en lieu de nous demander quelles vont être les avantages de Docker par rapport à une virtualisation classique. Tout d'abord, le conteneur n'embarque pas de système d'exploitation supplémentaire, il utilise le noyau du système sur lequel il s'exécute. Nous avons donc une grande rapidité d'exécution mais également une perméabilité accrue entre la machine hôte et le conteneur. Docker étant récent et attrayant, on trouve beaucoup de documentation à son sujet. Il existe par exemple une base de données de conteneurs pour diverses applications. On peut citer quelques entreprises ayant intégré ce système comme Amazon sur AWS (Amazon Web Services), Google sur Google Compute ou encore OVH et DigitalOcean.

Nous avons donc rapidement décidé, en suivant les recommandations de nos encadrants, d'utiliser Docker. Durant les premières semaines de travail, nous nous sommes renseignés sur cet outil et sur son fonctionnement. Nous l'avons installé sur une machine et réalisé de multiples tests. Docker utilise donc le noyau de la machine hôte, les conteneurs Linux LXC notamment pour leur système d'isolation.

Couche de supervision

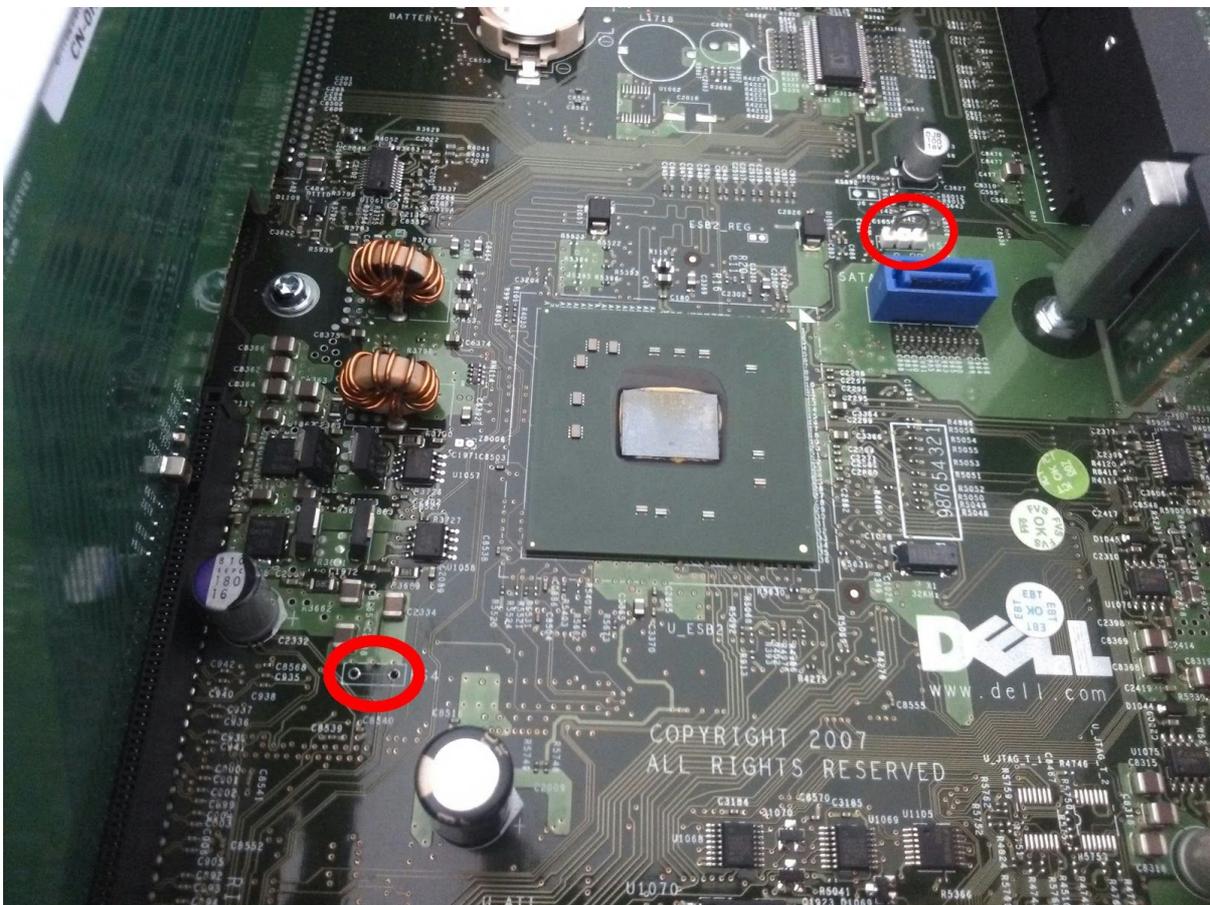
Le but était ensuite de savoir comment utiliser au mieux Docker, l'adapter à nos besoins et l'intégrer dans notre système. Pour cela l'utilisation d'Openstack nous a été initialement suggérée pour la supervision de notre système. Nous nous sommes donc renseigné à ce sujet auprès de Monsieur Redon, Alexandre Chojnacki un intervenant extérieur avec qui nous avons eu cours et Thomas Maurice, IMA diplômé l'année dernière utilisant Docker et Openstack quotidiennement. De ces différentes entrevues nous en avons conclu que l'utilisation d'Openstack était trop lourde pour être pertinente pour notre utilisation. Nous avons donc arrêté nos recherches à ce sujet et décidé de mettre cet outil de côté. Nous allons créer nous-mêmes les scripts nécessaire au déploiement automatique de sites.

Deuxième couche virtuelle : Xen

Suite à ces discussions et à d'autres recherches nous avons également parlé de l'utilisation d'une couche intermédiaire de virtualisation pour une meilleure isolation et résilience en cas de défaillance. Il existe plusieurs outils tels que xen, kvm ou vmware. Nous nous sommes néanmoins rapidement tournés vers une solution à base de para-virtualisation, l'hyperviseur de machine virtuelle (VM) Xen. Cet outil nous permet de créer plusieurs VMs sur lesquels nous avons des conteneurs Docker.

Nous avons initialement travaillé sur les machines de la salle projet pour réaliser nos différents tests sur Docker et Xen séparément. Un nouvel élément est venu changer notre façon de tester.

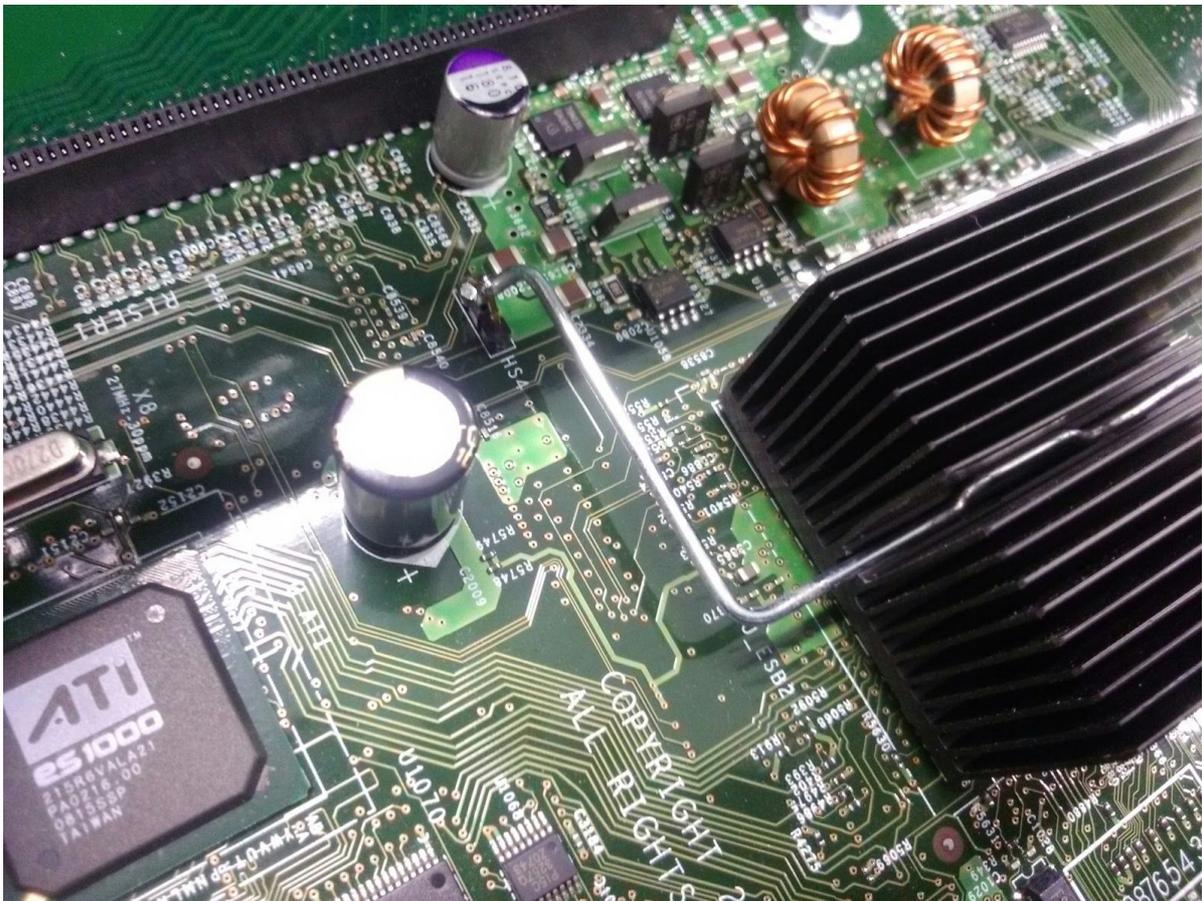
Un nouveau serveur



Nous avons reçu une machine DELL Poweredge 2950. Ce serveur était à notre entière disposition à la condition de réparer le dissipateur thermique d'un des composants.

En effet on peut voir sur l'image ci-dessus que l'une des anses aidant à maintenir le dissipateur en place s'est détachée. Après avoir évoqué plusieurs solutions, nous avons créé une anse similaire à celle déjà existante en soudant deux broches. Le résultat est des plus satisfaisants et le dissipateur est à nouveau maintenu correctement.

Nous avons procédé ensuite à la réinstallation complète de la machine sous Debian (Jessie).

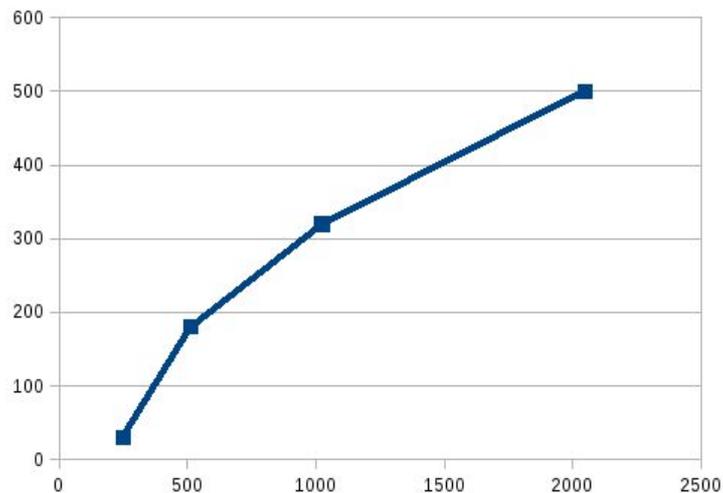


Benchmark de performance

Une fois le serveur en place nous avons pu réaliser nos premiers essais avec les machines virtuelles Xen et les conteneurs. Nous avons un test rudimentaire afin de dimensionner correctement les machines virtuelles.

Nous avons fait le choix de déterminer de manière statique le nombre de machines Xen sur le serveur. Ceci permet d'éviter de nombreux problèmes liés à l'allocation dynamique des ressources. Il faut donc bien dimensionner les machines pour garder une adaptabilité maximum, en anticipant la charge.

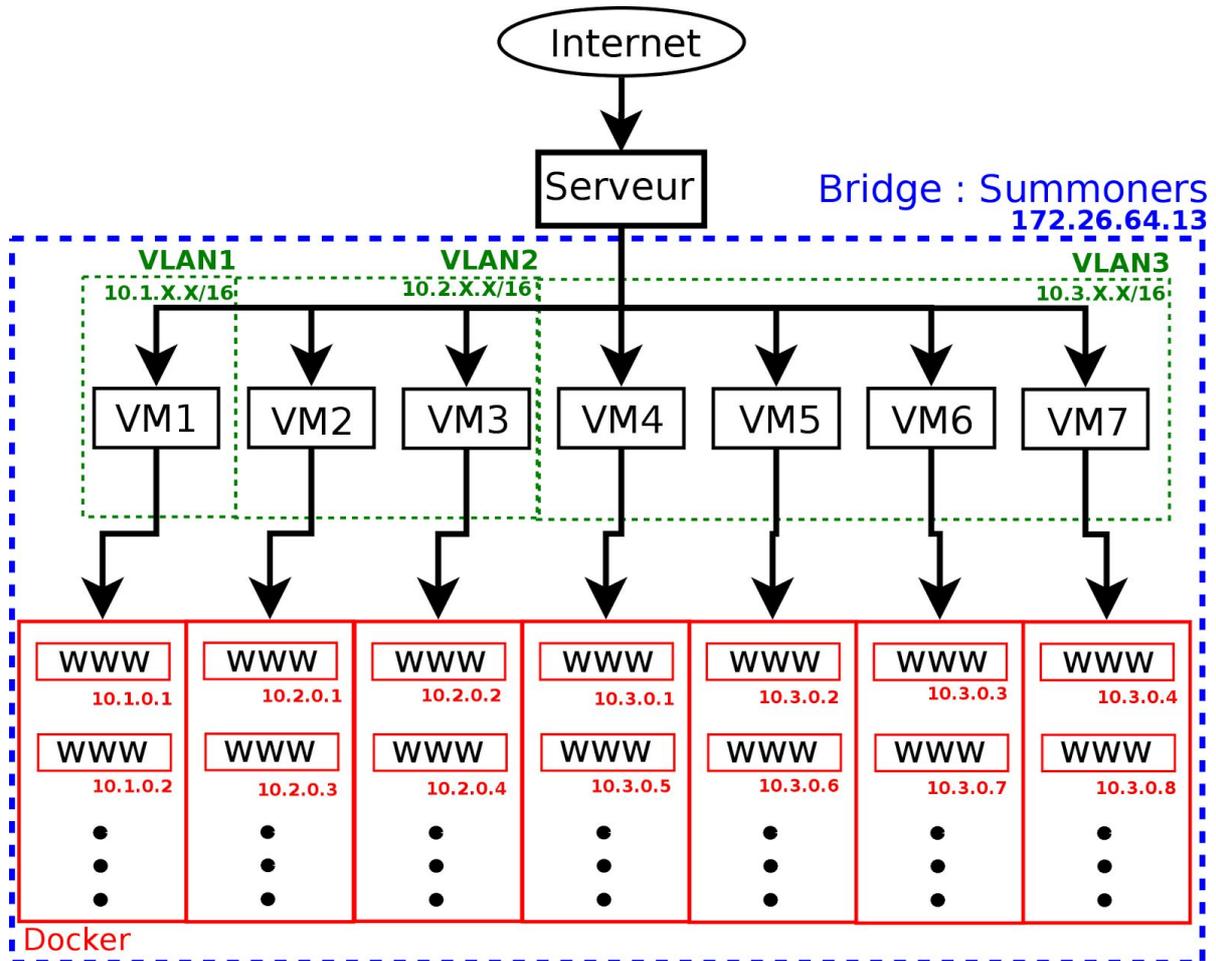
Pour cela, nous avons essayé plusieurs configurations pour voir comment réagissent les conteneurs. On charge le maximum de conteneurs sur une machine virtuelle, jusqu'à saturation de la mémoire disponible. Les conteneurs utilisés font tourner un serveur Nginx avec un site léger, sans aucun client. On établit alors un maximum théorique de conteneur par VM en fonction de la mémoire :



Le résultat de ces tests montre que la rentabilité idéale se situe aux alentours de 512Mo par VM. Une répartition de la mémoire selon ces résultats serait pertinente.

Réseau

Nous devons faire communiquer les conteneurs entre eux pour qu'ils répondent au cahier des charges exigé. En effet, pour qu'une seule adresse routée ne soit utilisée, nous avons choisis d'utiliser les fonctionnalités des sous-réseaux propre à Docker. Voici une maquette illustrant la hiérarchie à atteindre.



Prochains objectifs

À long terme, le but est évidemment d'arriver à respecter notre cahier des charges. Nous avons cependant divisé le travail en tâche plus petites afin d'avoir des objectifs à court terme.

Architecture réseau

A l'heure actuelle, nous avons deux machines virtuelles Xen qui communiquent par le bridge Docker configuré pour un réseau **172.17.0.0/16**. Notre objectif pour la partie réseau est de déplacer ce réseau vers celui décrit dans la maquette (10.0.0.0/8).

Le benchmark décrit plus haut a également permis de vérifier le comportement de l'attribution automatique des adresses IP dans le sous réseau. Nous avons bien un remplissage consécutif et sans conflit, même lorsque deux conteneurs sont créés sur des VMs différentes.

Nous devons ensuite créer les VLANs et y mettre les VMs concernées. Si le temps le permet, il serait intéressant d'utiliser une machine de projet PRA, qui possède une adresse IP routée, pour établir une connexion à internet.

Fonctionnalités

Un de nos principaux objectifs sera la configuration d'un serveur DNS, notamment la partie reverse DNS afin d'associer correctement les noms de domaines aux bonnes sous-adresses.

Nous devons ensuite penser à un point très important, à savoir le transfert de fichier vers le conteneur. L'utilisateur doit pouvoir envoyer ses fichiers sur son conteneur associé de manière facile et rapide. La solution actuellement utilisée est le FTP (File Transfer Protocol), et ce protocole présente de nombreux avantages qui nous ont parus intéressants. Cependant le transfert **sécurisé** est très important. Heureusement, il existe plusieurs solutions qui utilisent FTP par dessus une connexion sécurisée. Nous nous tournons actuellement vers **SFTP** utilisant le protocole SSH avec toutes les fonctionnalités du reverse SSH.

Maquette du site

L'utilisateur doit pouvoir créer et gérer son site. Nous proposons donc une interface web qui réunit les fonctionnalités auxquelles un utilisateur a droit.

Après identification avec les identifiants Polytech, nous récupérons les informations d'un utilisateur. Grâce à ces informations, nous savons quel VLAN sélectionner. Lors d'une création de site, il devra spécifier un nom de domaine et les modules qu'il désire installer (php, blog, etc). De là, un script lance la création dans le VLAN adéquat d'un conteneur avec les services demandés. Idéalement, une documentation informe l'utilisateur de la procédure à suivre pour transférer des fichiers vers son site. Il aura bien entendu en début de page la liste des noms de domaines liés à son compte.

Nous devons également voir comment gérer les certificats SSL pour une sécurisation des sites et donc un accès par HTTPS.

Conclusion

Ce projet est une très bonne occasion de découvrir en détail une technologie actuelle du milieu de la virtualisation : Docker. Le retour des ingénieurs en poste qui confirment l'utilité quotidienne de cet outil nous motive beaucoup.

La prise de connaissance sur la première partie de ce projet fût énorme, nous attendons autant de stimulation pour la suite. Dans le volume horaire qui nous est donné, les objectifs fixés paraissent réalistes et devraient nous permettre d'aboutir ce projet.