Dossier d'architecture technique

WhatSup au service de la performance

Qwirk

|  |  |  |  |
| --- | --- | --- | --- |
| **Diffusion interne** | | **Code** | |
| **Commercial** | Adrien MONTAGU | | I |
| **Avant-vente** | - | | I |
| **Chef de Projet** | Pierre-Eric TESSA | | ACR |
| **Consultant(s)** | Equipe projet | | C |

|  |  |  |  |
| --- | --- | --- | --- |
| **Diffusion client** | | **Code** | |
| **DSI** | - | | AC |
| **Chef de Projet** | - | | I |
|  |  | |  |

***Code : A****pprobation,* ***I****nformation, a****C****tion, a****R****chivage*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Informations qualité** | | | | | |
| **Date de sortie** | 11/12/2017 | **Numéro de la dernière page** | | | 21 |
| **Rédigé par** | Guillaume LOYER et Pierre-Eric TESSA | | | | |
| **Vérifié par** | Guillaume LOYER et Pierre-Eric TESSA | | **Visa** | 09/06/2017 | |
| **Approuvé par** | Pierre-Eric TESSA et Adrien Montagu | | **Visa** | 09/06/2017 | |

| **Version** | **Date** | **Auteur** | **Nature des modifications** |
| --- | --- | --- | --- |
| 0.1 | 15/05/2017 | Guillaume LOYER | Création du document |
| 0.2 | 07/06/2017 | Guillaume LOYER | Description des modules |
| 0.3 | 08/06/2017 | Pierre-Eric TESSA | Révision |
| 0.4 | 09/06/2017 | Adrien Montagu | Correction orthographique et grammaticale intégrale |
|  |  |  |  |

Table des matières

[1. Présentation 3](#_Toc484772071)

[2. Architecture générale 3](#_Toc484772072)

[2.1. Rappel du besoin 3](#_Toc484772073)

[2.2. Objectifs du projet 3](#_Toc484772074)

[2.3. Schéma de principe d'architecture 4](#_Toc484772075)

[3. Architecture physique 5](#_Toc484772076)

[3.1. ESXi 5](#_Toc484772077)

[3.2. Actif directory 5](#_Toc484772078)

[3.3. CDN 6](#_Toc484772079)

[3.4. Serveurs Web 6](#_Toc484772080)

[3.5. Serveur MySQL 7](#_Toc484772081)

[4. Architecture logique 7](#_Toc484772082)

[4.1. Versions logicielles 7](#_Toc484772083)

[4.2. Virtualisation 7](#_Toc484772084)

[4.3. NGINX 9](#_Toc484772085)

[5. Sécurité 10](#_Toc484772086)

[5.1. Renforcement de la sécurité ESXi 10](#_Toc484772087)

[5.2. Sécurité applicative pfsense 11](#_Toc484772088)

[5.3. Sécurisation de la couche Données 11](#_Toc484772089)

[6. Administration 12](#_Toc484772090)

[6.1. Accès à l'équipement 12](#_Toc484772091)

# Présentation

Ce document constitue le Dossier d'Architecture Technique (DAT) concernant le projet d'intégration et de déploiement de l'infrastructure de haute disponibilité avec répartition de charge sur le site Web de l'entreprise Qwirk. Ce document a pour but de faire l'adéquation entre les besoins spécifiques de l'entreprise et la façon dont ceux-ci sont intégrés à l'infrastructure déployée. Ce document décrit l'ensemble du paramétrage que préconise **WhatSup** pour l'intégration de cette infrastructure.

# Architecture générale

*Cette section définit le contexte du projet haute disponibilité*

## Rappel du besoin

*Ce chapitre présente le problème à résoudre, tel que nous l'avons perçu pour établir cette spécification*

Qwirk est une startup dédiée aux applications de communication personnelles et professionnelles. Leur nouveau produit est un système complet de messagerie instantanée, tel que Skype ou Slack.

Notre équipe a été sélectionnée pour écrire cette application qui fonctionnera sur plusieurs systèmes d’exploitation tels que Windows, MacOS ou Linux.

## Objectifs du projet

Qwirk ++ est un projet de chat à la frontière entre Skype et Slack. Ces deux technologies ont leur avantage et leurs inconvénients.

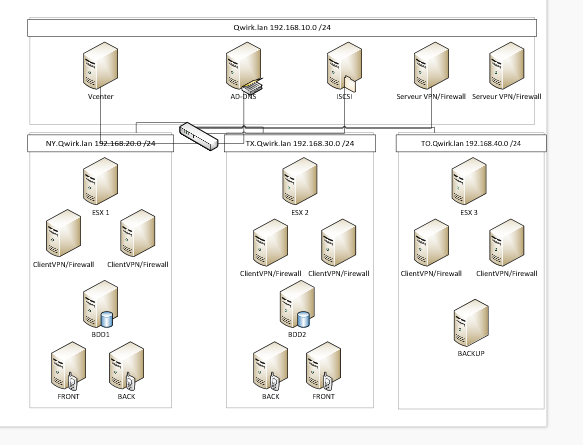
Skype est de moins en moins utilisés en entreprises, a des problèmes de compatibilité sur les différentes plateformes et entre ces différentes versions.

Slack est un outil spécialement conçu pour les entreprises et ne permet pas d’ajouter n’importe qui se trouvant sur sa plateforme pour engager une conversation avec lui.

Qwirk ++ arrive à la croisée des deux chemins pour offrir une expérience aussi bien web que desktop ou n’importe quel utilisateur peut en ajouter un autre et commencer une conversation avec lui.

## Schéma de principe d'architecture

*Voici le schéma d'architecture général de l'infrastructure mise en place*



# Architecture physique

*Cette section définit l'architecture physique de la solution mise en place dans le cadre du projet haute disponibilité du site Qwirk*

Tous nos serveurs physiques ont été achetés chez dell pour bénéficier d'une garantie en 24h00 ouvré lorsque vous rencontrez un problème technique vous empêchant d'utiliser vos serveurs.

## ESXi

|  |  |  |  |
| --- | --- | --- | --- |
| ***Nom*** | ***ESX1*** | ***ESX2*** | ***ESX3*** |
| ***Hyperviseur*** | *ESXi* | *ESXi* | *ESXI* |
| ***Mémoire*** | *16* | *16* | *16* |
| ***Nombre de vCPU*** | *4* | *4* | *4* |
| ***Disques*** | *300 go 10000 tr/mn* | *300 go 10000 tr/mn* | *300 go 10000 tr/mn* |
| ***Interface réseau*** | *eth0* | *eth0* | *eth0* |

*Caractéristiques techniques des hyperviseurs ESXi*

*Caractéristiques techniques du vCenter sur site primaire*

|  |  |
| --- | --- |
| **vCenter** | **Site primaire** |
| Mémoire | 8GB |
| Nombre de vCPU | 4 |
| Disque | 1 000 go 7200 tr/mn |
| Interface réseau | eth0 |

Pour le site secondaire, on utilise le client vSphere version 6 pour administrer les serveurs.

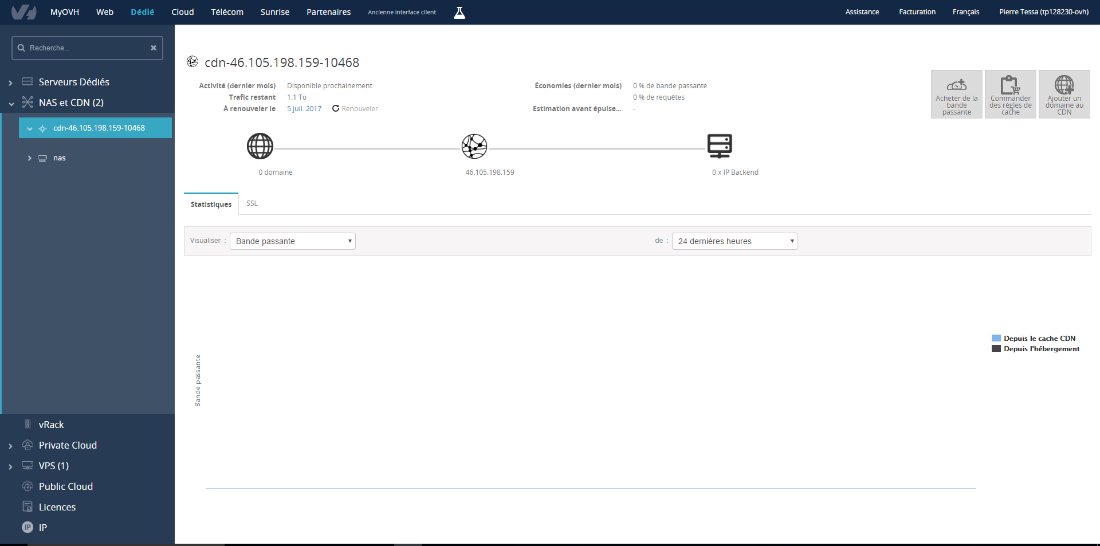
## Actif directory

*Caractéristiques techniques du serveur actif directory*

|  |  |
| --- | --- |
| **Actif directory** | **Site principal** |
| Mémoire | 4GB |
| Nombre de vCPU | 4 |
| Disque | 1 000 go 7200 tr/mn |
| Interface réseau | eth0 |

## CDN

Nous avons jugé plus utile de sous-traiter ce serveur à OVH qui est un groupe spécialisé dans l'hébergement. Notre offre nous revient à 10 € /mois avec un nom de domaine offert.



Nous pourrons ajouter nos serveurs en faisant du port forwarding et en mettant un nom de domaine à notre CDN. Pour plus de renseignement la procédure de d'ajout de serveurs back end, nous vous recommandons d'aller dans la FAQ du site OVH.

## Serveurs Web

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Site New York** | | **Site Dallas** | |
| Nom | Ny-webserver1 | Ny-webserver2 | Tx-webserver1 | Tx-webserver2 |
| Hyperviseur | ESXi1 | ESXi2 | ESXi1 | ESXi2 |
| Mémoire | 4GB | 4GB | 4GB | 4GB |
| Nombre de vCPU | 2 | 2 | 2 | 2 |
| Disque | 40 GB | 40 GB | 40 GB | 40 GB |
| Interface réseau | eth0 | eth0 | eth0 | eth0 |

*Caractéristiques techniques des serveurs web nginx*

## Serveur MySQL

|  |  |  |
| --- | --- | --- |
| **Nom** | **MySQL** | **tymysql** |
| **Mémoire** | 4GB | 4 GB |
| **Nombre de vCPU** | 2 | 2 |
| **Disque** | 1 disque de 16 go + 1to (database) | 1 disque de 16 go + 1to (database) |
| **Interface réseau** | eth0 | Eth0 |

# Architecture logique

*Cette solution définit l'architecture logique de la solution mise en place dans le cadre du projet haute disponibilité du site Qwirk*

## Versions logicielles

*Ce chapitre présente les caractéristiques logicielles des équipements utilisés*

Les équipements sont installés avec les versions logicielles suivantes :

|  |  |  |
| --- | --- | --- |
| **Service** | **Version** | **Site** |
| Hyperviseur Type 1 | ESXi 5.5 | Tous |
| Serveur Web | Nginx **1.6.2-5** | New York / Dallas |
| vCenter Server | 5.5 | Primaire |

## Virtualisation

*Ce chapitre présente les caractéristiques des plateformes de virtualisation mises en place*

Nous mettons en place une plateforme de virtualisation réparties sur trois sites physiquement séparés.

Celle-ci est de type 1, c'est-à-dire installée directement sur la couche matérielle du serveur. Le choix s'est porté sur l'ESX Server de VMware.

La virtualisation a plusieurs avantages par rapport aux machines physiques :

* Elle permet la gestion de plusieurs systèmes d'exploitation.
* Elle facilite les migrations de données entre sites.
* Elle permet de répartir la charge sur plusieurs serveurs en fonction du trafic.

Sur ces plateformes, nous avons mis en place trois Datacenters. Avoir 2 serveurs sur 2 sites permet donc une redondance de l'information, avec un risque de panne divisé par 2 x 2 = 4.

Sur deux des sites physiques, nous faisons travailler les deux serveurs web ensemble pour optimiser le temps de chargement des utilisateurs.

Puis pour sauvegarder l'intégralité des machines virtuelles nous virtualisons une station dédiée à la sauvegarde des machines virtuelles via le logiciel VEAM.

Nous verrons par la suite en détail les solutions utilisées.

**Capacités des machines virtuelles sous ESXi**

Les machines virtuelles s’exécutant sur ESXi présentent les caractéristiques suivantes :

* Les machines virtuelles peuvent prendre en charge jusqu’à 128 CPU virtuels.
* Les machines virtuelles peuvent prendre en charge jusqu’à 4 To de RAM.
* Le contrôleur xHCI prend en charge les périphériques USB 3.0.
* L’interface AHCI (Advanced Host Controller Interface) prend en charge jusqu’à 120 périphériques par machine virtuelle.
* La taille maximum d’un disque de machine virtuelle est de 62 To.
* La fonctionnalité de récupération d’espace de stockage de l’OS client rend au pool de stockage l’espace disque libéré au sein de l’OS client.
* La virtualisation du CPU est optimisée, grâce à la présentation d’un plus grand nombre d’informations sur l’architecture du CPU hôte aux machines virtuelles. Cette meilleure exposition du CPU permet de déboguer, d’affiner et de dépanner plus facilement les systèmes d’exploitation et les applications dans la machine virtuelle.
* L’efficacité du CPU est accrue par la fonctionnalité LRO (Large Receive Offload) qui regroupe les paquets TCP entrants en un paquet unique plus important.

**Utilisation de vCenter Server**

 vCenter Server est le composant de gestion centralisé d’un Datacenter virtuel VMWare. Il joue le rôle d’administrateur des hôtes ESX/ESXi connectés au réseau, et des machines virtuelles présentes sur ces différents hôtes.

vCenter Server utilise un Système de Gestion de bases de données relationnelles de manière à stocker les informations relatives aux hôtes et configurations. Ici nous utilisons la version allégée du SGBDR Microsoft SQL Server 2005 Express packagée avec les fichiers d’installation de vCenter Server.

Ces composants supplémentaires sont installés sans notification lors de l'installation de vCenter Server :

* vCenter Inventory Service
* PostgreSQL
* vSphere Web Client
* vSphere ESXi Dump Collector
* VMware vSphere Syslog Collector
* Service VMware Syslog
* vSphere Auto Deploy

**Datastore**

Les datastores, ou banque de données, sont des conteneurs logiques qui masquent les caractéristiques propres à chaque périphérique de stockage. Elles sont caractérisées par une partition d’un volume formaté en VMFS. Les limitations suivantes doivent-être respectées :

* 1 banque de données = 1 seul système de fichiers.
* Jusqu’à 256 banques de données VMFS par système.
* 8 Banques de données NFS par système.

Une banque de données peut contenir les fichiers des machines virtuelles (disques, méta datas, etc.) et également des images ISO de disques (pour l’installation d’OS par exemple).

Ici nous nous servons des disques durs par défaut intégré sur les serveurs ESXi.

## NGINX

*Ce chapitre présente les caractéristiques des reverse-proxies mis en place*

Le principal but de notre serveur est d'héberger des sites web. Pour cela on a besoin d'installer un serveur HTTP qui va être capable de comprendre les requêtes faites par le navigateur et de renvoyer les bonnes informations en fonction

**Pourquoi nginx ?**Nginx a été conçu conçu dans le but d'être capable de répondre à plus de 10 000 requêtes simultanées. Nginx est du coup plus performant lors d'un trafic importan. Là où Apache crée un processus par connexion, Nginx lance une série de workers qui vont chacun être capable de gérer de multiples connexions d'une manière non bloquante.

Les Content Delivery Network sont assurés par des serveurs Nginx, qu’on appelle aussi dans ce cas des reverse proxy, ou serveur frontal.

Un CDN permet de stocker des ressources sur de multiples serveurs externes. Le but est de pouvoir fournir à chaque internaute les fichiers dont il a besoin de la manière la plus rapide possible. Pour cela, le CDN se base sur plusieurs éléments :

* Chaque fichier est présent sur plusieurs serveurs, permettant de réduire l’encombrement du site lors de pics de trafic.
* Les CDN sont équitablement répartis géographiquement, réduisant la distance serveur-ordinateur.
* Ces serveurs sont sans cookies, permettant de réduire le poids de chaque transfert en enlevant ces données inutiles pour les ressources statiques.
* Certains fichiers étant sur des serveurs différents, cela augmente le nombre de connexions simultanées autorisées entre l’ordinateur et les fichiers à télécharger.

Ces serveurs sont installés sur une base linux, Debian pour le site primaire et CentOS pour le site secondaire. La configuration des Nginx est la même sur les deux distributions.

A la première requête au serveur, le CDN cache les ressources statiques dans un répertoire préalablement défini.

# Sécurité

## Renforcement de la sécurité ESXi

L’optimisation de l’accès et du contrôle basés sur des rôles supprime la dépendance vis-à-vis d’un compte root partagé. Les utilisateurs et les groupes peuvent être dotés de droits d’administration complets. Il n’est pas nécessaire de partager un accès ni un compte super utilisateur commun pour pouvoir effectuer des tâches administratives.

**Journaux et audits exhaustifs**

vSphere ESXi enregistre toute l’activité des utilisateurs depuis le Shell et depuis l’interface utilisateur de console directe au titre du compte utilisateur. Cette journalisation garantit la responsabilisation et facilite les activités d’audit utilisateur.

**vMotion**

La migration à chaud de VMware vSphere vous permet de déplacer toute une machine virtuelle active d’un serveur physique à un autre, sans interruption de service. Les clients peuvent migrer les machines virtuelles actives entre les clusters, les commutateurs distribués et les instances vCenter sur de longues distances avec un temps de latence aller-retour jusqu’à 100 ms.

**Pare-feu sans état**

ESXi inclut un pare-feu sans état, orienté services. Ce pare-feu permet de définir des règles d’accès aux ports pour les services. Il permet aussi d’indiquer des plages d’adresses IP ou des adresses IP individuelles, qui peuvent se connecter à des services d’hôte.

## Sécurité applicative pfSense

La solution pfSense, étant le point d’entrée de l’infrastructure de chaque site, agit comme un véritable garant de la sécurité. Elle assure plusieurs niveaux de protection :

* Validation protocolaire (HTTP)
* Protection contre les attaques DoS, DDoS, vers, injections SQL, etc.
* Gestion des ACLs
* Gestion des listes noires / blanches
* Filtrage des URLs et la possibilité de restreindre l'accès par un système d'authentification
* Blocage des fuites d'information
* Filtrage des requêtes et des réponses HTTP/HTTPS

## Sécurisation de la couche Données

La base de données SQL Microsoft Azure offre des mesures de sécurité à tous niveaux :

* **Sécurité de la connexion :**

On crée une règle de pare-feu au niveau du portail de Microsoft Azure pour refuser toute tentative de connexion d’une adresse IP qui ne serait pas celle du site primaire ou celle du site secondaire.

Toutes les données en transit depuis et vers la base de données sont chiffrées (SSL/TLS). Dans la chaîne de connexion de l’application, on spécifie les paramètres **Encrypt = True** et **TrustServerCertificate = False,** permettant de chiffrer la connexion et de ne pas accepter le certificat du serveur. Sans cela, la connexion ne vérifie pas l’identité du serveur et peut faire la cible d’attaques de types PITM.

* **Authentification :**

Il s’agit du processus de validation de son identité lorsqu’on se connecte à la base de données. Pour le master, on utilise l’authentification SQL qui utilise un nom et un mot de passe. On ajoute un autre administrateur de serveur qui utilise des identités gérées par Azure Active Directory et est prise en charge pour des domaines gérés et intégrés. Cet administrateur peut effectuer toutes les opérations de l’administrateur master. On utilise cet utilisateur depuis l’application. Cela permet de limiter les autorisations accordées à l’application et de réduire les risques d’activité malveillante comme les attaques par injection de code SQL.

* **Autorisation :**

Les actions que l’on peut exécuter dans la base de données sont contrôlées par les permissions et l’appartenance au rôle du compte utilisateur. Pour limiter le nombre d’actions que peut réaliser un utilisateur, nous définissons des autorisations granulaires, c’est-à-dire que certaines colonnes sont accessibles en lecture seule, d’autres en lecture écriture et d’autres en lecture écriture et modification.

* **Chiffrement :**

Vos données « au repos » ou stockées dans des fichiers de base de données et des sauvegardes sont chiffrées en AES 256B. Il est possible aussi de chiffrer vos données au niveau des cellules ou bien via un module de sécurité matériel.

* **Audit :**

La fonction d’audit permet d’enregistrer les évènements survenus dans la base de données dans un journal d’audit au sein de votre compte Microsoft Azure Storage. Elle permet aussi d’en sortir des rapports et des analyses.

* **Conformité :**

Elle participe à des audits de sécurité régulier et est certifiée conforme avec de nombreuses normes.

* **Backup intégré :**

Les bases de données sont automatiquement sauvegardées dans les partitions partagées des deux serveurs mysql. Cette sauvegarde à lieux tous les jours à minuit pour des raisons de sécurité.

# Administration

*Cette section définit les modes d'administration de la solution dans le cadre du projet haute disponibilité du site Qwirk*

## Accès à l'équipement

Les serveurs sont stockés dans des environnements adaptés et sécurisés dont l’accès est restreint aux personnes responsables.

Pour toute action nécessitant une intervention physique sur nos serveurs, nous faisons appel à nos équipes sur place.

Pour l’accès à distance, nous nous connectons en RDP (port standard sécurisé 3389) sur le serveur VCenter.