Beni MULUMBA

par

Présenté le 29/09/ 2022

Cloudification de l’infrastructure de  
Medicarche

Déploiement de l’infrastructure

Encadrant : Frédéric DIOLEZ

Superviseur : Frédéric DIOLEZ et Gerard LAMBION

Table des matières

[I. Introduction 3](#_Toc114314630)

[II. Environnement de travail 3](#_Toc114314631)

[III. Procédure de déploiement de l’infrastructure Medicarche 5](#_Toc114314632)

[1. Installation d’Openstack 7](#_Toc114314633)

[IV. Déploiement des applications de l’entreprise Medicarche 16](#_Toc114314634)

[1. Installation d’Odoo 16](#_Toc114314635)

[2. Installation de Nextcloud 17](#_Toc114314636)

[3. Installation de Syncthing 19](#_Toc114314637)

[4. Site web Medicarche 20](#_Toc114314638)

[V. Description du plan des tests 20](#_Toc114314639)

[1. Test de montée en charge 20](#_Toc114314640)

[2. Test de stress du système 21](#_Toc114314641)

[3. Test de performance 21](#_Toc114314642)

[VI. Information d’accompagnement au changement 23](#_Toc114314643)

[VII. Conclusion 24](#_Toc114314644)

[VIII. Webographie 24](#_Toc114314645)

# Introduction

A titre de rappel la société Medicarche spécialisée dans la recherche sur les virus, est située sur 2 sites, à LA DÉFENSE dans la Grande Arche (site dénommé ARCHE), et à PARIS rue

de l’Arcade, dans le 8ème arrondissement (site dénommé ARCADE). Sa croissance est

actuellement importante, en raison des demandes liées à la crise sanitaire. Aussi elle envisage d’ouvrir de nouveaux sites, pour répondre aux besoins du marché. Pour répondre au cahier de charge de l’entreprise Medicarche concernant la migration dans le cloud, nous mettons en place le déploiement automatisé de l’infrastructure dans le cloud.

En entrée de jeu nous avons déployés 3 applications web et un site web qui sont Syncthing, Nextcloud et Odoo. Les différentes applications sont hébergées dans un sous réseau privé dont seuls les collaborateurs de l’entreprise auront accès. Pour rendre cela possible nous avons développé deux types de script. Le premier script se déploie de bout en bout sans interaction humaine en exécutant la commande de déploiement du système et le second script qui est considéré comme le cœur du projet peut se déployer indépendamment du premier. Dans les lignes qui suivent nous développerons l’environnement de travail de l’entreprise Medicarche, le processus de déploiement de l’infrastructure, le plan de test, l’accompagnement au changement et enfin nous pourrons conclure.

# Environnement de travail

L’environnement de travail qui a permis de mettre en place le déploiement de l’infrastructure de l’entreprise Medicarche est composé d’une machine hôte dont le système d’exploitation est Microsoft Windows 10. La machine hôte est une machine dont le type de processeur est un i5 composé de 8 cœurs physique et 4 cœurs logique, d’une mémoire RAM de 16 gigabyte et d’un disque dur de 300 gigabyte. Comme hyperviseur nous avons utilisé VirtualBox qui est un hyperviseur open source. Pour faciliter la communication au niveau réseau nous utilisons deux cartes réseaux. La première carte réseau est configuré en utilisant l’option NAT qui permettra la communication de la machine virtuel hôte vers internet et la deuxième carte réseau est configuré avec l’option hôte privée pour faciliter la communication être la machine virtuelle et la machine physique.

Le déploiement a été réalisé grâce à Vagrant qui est un outil de création et de configuration des environnements de développement virtuels qui facilite aussi l’automatisation. Avec Vagrant on peut déployer des systèmes d’exploitation légère, moins gourmande en ressource en quelques minutes. A l’aide de Vagrant nous avons déployé la machine virtuelle avec les caractéristiques suivantes :

* Système d’exploitation Linux Ubuntu Focal 20.04
* Disque dur de 100 gigabytes
* Mémoire RAM de 10 gigabytes
* 12 vCPUs
* Adresse IP 192.168.33.16

Graphical user interface, application, email

Description automatically generated

Figure 1 – caractéristique machine virtuelle

Visual Code est un environnement de développement intégré puissant qui facilite la création de scripts. Pour administrer la machine hôte qui héberge le cloud privé de l’entreprise Medicarche nous avons installé des plugins. Parmi les extensions que nous avons installées nous avons RemoteSSH une extension qui permet de se connecter en mode SSH dans les machines virtuelles se trouvant dans le cloud privé de l’entreprise Medicarche pour pouvoir effectuer des actions d’administration si cela s’avère nécessaire.

Packer est une solution open source permettant de construire des images machine pour de multiples plateforme cloud. Pour construire les images que nous allions utiliser dans Openstack premièrement nous avons récupérer une image Linux Ubuntu Bionic 18.04 sur le site officiel de Openstack avec le format QCOW2. Pour personnaliser l’image nous avons utiliser Packer en définissant un utilisateur par défaut et le mot de passe, la capacité maximale de mémoire RAM, de disque dur, le format, le format du clavier, la langue du système d’exploitation et le nombre de processeur pouvant être utilisé.

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre et gratuit. Elle nous a été utile pour faire du versioning des scripts utilisé tout au long du déploiement de l’infrastructure. Nous mettrons à votre disposition tous les scripts utilisés pour déployer le cloud privé de Medicarche.

Logo, company name

Description automatically generated

Figure 2 - environnement

# Procédure de déploiement de l’infrastructure Medicarche

Le déploiement de l’infrastructure se fait en utilisant le concept de l’infrastructure as a code. De la mise en place de l’infrastructure Openstack au déploiement des applications nous avons utilisé des scripts bash pour automatiser l’installation.

Le script est écrit de sorte à automatiser :

* Le déploiement de l’infrastructure Openstack
* La création des instances
* La création des pairs de clé RSA pour les instances
* La création des règles de sécurité
* Le clonage et le téléversement de l’image Linux Ubuntu et son déploiement
* La configuration de la connexion SSH de chaque instance pour assurer le déploiement des applications qui sont Syncthing, Nextcloud, Odoo et le site de l’entreprise Medicarche.

Nous avons pensé à la mise en place d’un serveur mandataire pour permettre aux employés de Medicarche d’avoir accès aux applications de manière sécurisée.

Avant d’entrer dans le vif du sujet nous aimerons faire une piqure de rappel concernant l’architecture de base présenté lors de la conception de l’infrastructure Medicarche. Cette architecture représente les besoins réels effectué lors de la rédaction du cahier de charge. Lors de la conception de la plateforme de Medicarche, nous avons proposé deux types de solution cloud qui sont AWS un fournisseur public et Openstack un Cloud de type privée. L’entreprise Medicarche avait opté pour Openstack lors de l’appel d’offre.

Maquette originale

Description automatically generated

Figure 3 – maquette de base

À la suite des difficultés rencontrées en rapport avec les performances de la machine virtuelle hôte où s'exécute Openstack nous sommes obligé d'utiliser que des services qui nous permettront d’avoir une maquette qui fonctionne correctement à l’aide du script de déploiement et par après une fois que le problème résolus nous pourrons revenir sur la maquette de base.

Diagram, schematic

Description automatically generated

Figure 4 – maquette déployé

Le déploiement de l’infrastructure Medicarche a été scripté de bout en bout. Pour commencer le déploiement on a juste exécuté la commande vagrant up une commande de l’outil vagrant qui permet de créer une machine virtuelle. Nous avons provisionné le script de déploiement dans le fichier vagrantfile afin d’automatiser l’installation de Openstack ainsi que le déploiement des applications et du site web.

Text

Description automatically generated

Figure 5 – provisionnement vm

## Installation d’Openstack

OpenStack est un ensemble de logiciels [open source](https://fr.wikipedia.org/wiki/Open_source) permettant de déployer une infrastructure de type cloud privé en suivant le model [infrastructure en tant que service](https://fr.wikipedia.org/wiki/Infrastructure_as_a_service) (Iaas). Une fois l'environnement Openstack installé, les credantials nous sont fournis afin d'accéder à la console en tant qu’administrateur.  Par défaut Openstack configure la partie réseau en créant un sous réseau privé, un sous réseau public, la table de routage et d'autres services pour ne citer que ceux-là. La commande iptables nous permet d’autoriser le trafic venant de Openstack vers internet.

Text

Description automatically generated

Figure 6 – script d’installation

1. Tableau de bord

 Un tableau de bord permet de visualiser l’ensemble de ressources de l’infrastructure. Cela nous permet aussi de garder un œil sur l’évolution de l’infrastructure. Nous pouvons avoir ­­facilement une vue sur les instances en cours d’utilisation, la quantité des vCPUs, la quantité de la RAM alloué aux instances en cours d’utilisation, les volumes, les groupes de sécurités, les sous réseaux, les ports, les routeurs ainsi les adresses IP publique.

Graphical user interface, chart

Description automatically generated

Figure 7 – Tableau de bord

1. Topologie réseau dans Openstack

Lors de l’installation de Openstack plusieurs fonctionnalités sont installées par défaut. Une topologie réseau représentant l’architecture du réseau. Nous avons utilisé un sous réseau privé et un sous réseau public. Pour de raison de sécurité, nous avons déployer les machines virtuelles contenant les applications web dans le réseau privé et nous avons pensé à les attribuer une adresse IP privé dont la passerelle est 192.168.222.1. Pour permettre la communication avec internet, un routeur virtuel fait office de passerelle entre le réseau privé et le sous réseau public. L’adresse IP de la passerelle est 10.20.20.224.

Diagram, timeline

Description automatically generated

Figure 8 – Topologie réseau

1. Les gabarits

Openstack permet de personnaliser des gabarits qui sont utilisés pour déployer les instances. Le plus petit *gabarit* par défaut consomme 512 Mo de mémoire par instance. Dans le contexte du déploiement de la plateforme Medicarche nous avons créé différent type de gabarit qui permettrons d’accueillir les applications. Lors de la création du gabarit on définit les paramètres suivants :

* Le nom
* Le VCPU
* La RAM
* Le disque dur principal
* Possibilité d’ajouter un disque dur éphémère
* La portée public ou privée
* Le swap

On a aussi la possibilité d’ajouter des metadatas pour mieux identifier le gabarit.

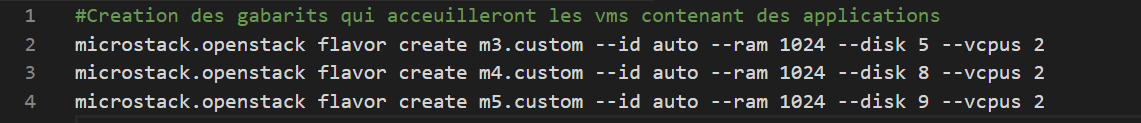


Figure 9 – script création gabarit

Graphical user interface, application

Description automatically generated

Figure 9 – Gabarit

1. Les images

Par défaut Openstack offre une image par défaut Cirros qui permet d’effectuée des tests au niveau de la couche réseau. Nous avons automatisé la création et le téléversement de l’image dans Openstack. Une image Linux Ubuntu Bionic sera utilisée par tous les instances. Lors du téléversement de l’image on définit le nom du système d’exploitation, la visibilité et le format du disque (QCOW2).

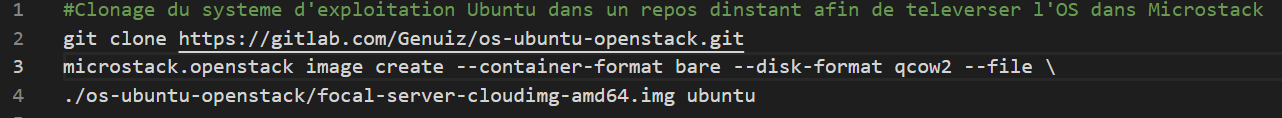


Figure 10 – script installation image

Pour pouvoir mettre en place cette solution nous avons créé un dépôt git pour stocker les images que nous allons utiliser lors de l’automatisation. On commence par cloner le dépôt à la racine puis on exécute la commande qui microstack.openstack pour crée l’image ubuntu depuis le dossier os-ubuntu-openstack.

Graphical user interface, text, application, email

Description automatically generated

Figure 11 – images dans Openstack

1. Les instances

Les instances dans Openstack sont des machines virtuelles qui s’exécute dans un sous réseau. Nous avons déployé des applications web Odoo, Nextcloud, Syncthing et le site web de l’entreprise Medicarche. Pour chaque application nous avons utilisé de gabarit spécifique pour chaque machine virtuelle afin de respecter le minimum requis dans la documentation des applications cité ci-haut.

Nous avons pensé à mettre les machines virtuelles dans un même sous réseau privée et avons utilisé un groupe de sécurité pour apporter une couche au niveau de la sécurité. Cela à pour objectif de ne mettre en place le principe de moindre privilège c’est-à-dire autoriser que les ports nécessaires en entrée. Parmi les ports autorisés pour atteindre les machines virtuelles nous avons :

* Le port 8088 qui permet d’atteindre le site web de l’entreprise
* Le port 8087 qui permet d’atteindre l’application Odoo
* Le port 8806 qui permet de communiquer avec l’application Nextcloud
* Le port 8885 qui permet de communiquer avec l’application Syncthing
* Le port 22 pour assurer une connexion sécurisée en SSH
* Le Port 80 pour autoriser la communication en http

Les machines virtuelles ont tous une adresse IP privée et une adresse IP public afin de communiquer en internet dans le réseau privée communication entre les machines virtuelles et en externe pour la communication avec l’extérieur. Openstack propose un outil pour le log toutes les actions effectuées sur la machine virtuelle peuvent être audité grâce à cette fonctionnalité. Les machines virtuelles utilisent une clé privée crée lors du déploiement cela permet de mettre un accent sur la sécurité. Le script suivant donne un aperçu de la création de la machine virtuelle.

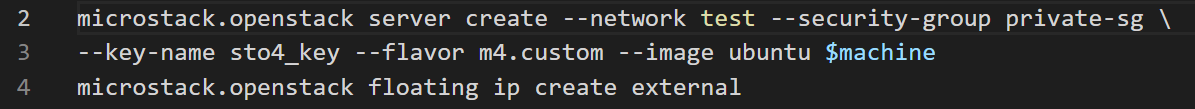


Figure 12 – script de création de la vm

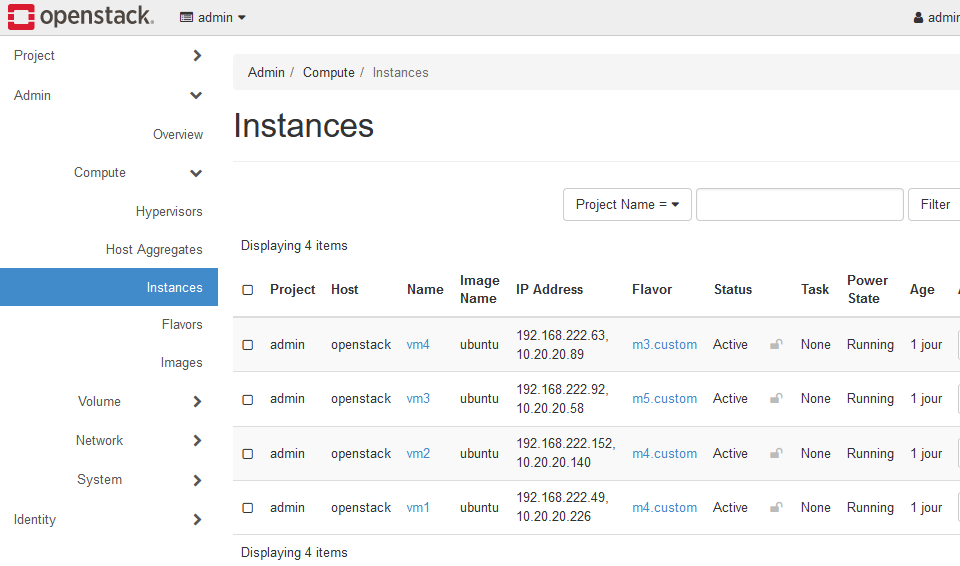


Figure 13 – les instances

1. Les groupes de sécurité

Un groupe de sécurité agit en tant que pare-feu virtuel pour les instances afin de contrôler le trafic entrant et sortant. Nous avons mise en place un groupe de sécurité nommé private-sg afin d’autoriser que les ports et les adresses IP nécessaire à la communication des instances. Afin de resserrer le trafic entrant nous avons appliqué la stratégie de moindre privilège en autorisant que le strict nécessaire.

Graphical user interface, text

Description automatically generated

Figure 14 – création de groupe de sécurité

Openstack offre une interface d’administration des groupes de sécurité, on peut ajouter, modifier ou supprimer le trafic entrant ou le trafic sortant. En cas de dépannage rapide cela permettra à la DSI de Medicarche de pouvoir intervenir pour pouvoir régler des problèmes.

Table

Description automatically generated

Figure 15 – les groupes de sécurité

1. Les pairs de clé

Pour avoir accès à la machine virtuelle et pouvoir faire les opérations d’administration nous avons lors de la création des instances mise en place un script qui permet de crée un pair de clé RSA et de l’associer à l’instance. Ce type de connexion permet de renforcer la sécurité ainsi seuls les administrateurs de Medicarche pourrons avoir accès aux instances pour effectuer les taches de maintenances. Le script suivant permet d’automatiser la création de clé pour se connecter en SSH.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 16 – script création de pair de clé

Graphical user interface, text, application, email

Description automatically generated

Figure 17 – Génération de la clé publique

1. Gestions des utilisateurs

Concernant la création des utilisateurs nous avons juger de le faire manuellement et non en passant par un script. Openstack offre une interface pour administrer les utilisateurs. La stratégie que nous avons adoptée est celle du moindre privilège. L’administrateur ayant tous les droits sera utiliser qu’en cas de nécessité. Nous créerons les utilisateurs les utilisateurs de la DSI qui auront les autorisations pour gérer l’infrastructure. Les rôles pourront être créé en ligne de commande afin d’autoriser les utilisateurs à accéder à des ressources dont ils ont le droit. 3 types de rôle sont créé par défaut :

* Membre
* Admin
* Reader

Graphical user interface, text, application, email

Description automatically generated

Figure 17 – les utilisateurs

1. Les adresses IP flottante

Les adresses IP flottante permettent aux instances de pouvoir accéder à internet. Elles ont été créées lors du déploiement en ligne de commande. Une interface de gestion des adresses IP flottantes permet de manager les adresses IP en mode graphique. On peut les attribuer ou les retirer à une instance à tout moment. La commande suivante permet de crée une adresse IP flottante en ligne de commande : microstack.openstack floating ip create external. Chaque instance ne peut avoir qu’une adresse IP flottante pour en avoir une nouvelle l’administrateur système devra libérer l’ancienne adresse IP avant d’en utiliser une nouvelle.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 18 – scripts de création adresse ip flottante

Graphical user interface, text, application, email

Description automatically generated

Figure 19 – gestions adresse ip flottante

# Déploiement des applications de l’entreprise Medicarche

Pour assurer le partage des documents, la synchronisation des données entre les services, les gestions des activités de l’entreprise nous avons mise en place un script qui permet de procéder à l’installation, la configuration ainsi que le déploiement des applications dans les machines virtuelles.

Pour répondre à la demande de l’entreprise nous avons proposé les applications suivantes :

* Odoo
* NextCloud
* Syncthing
* Wordpress

Que nous développerons dans les lignes qui suivent en donnant un aperçu sur ce que chacun des applications pourra apporter. L’architecture de déploiement reste le même pour toutes les applications. Le déploiement de la base de données et du serveur web se fait dans l’instance. Cela n’est pas une solution qui permet de rendre l’architecture hautement disponible et tolérante aux pannes mais à la suite des contraintes rencontré en termes de performance de la machine et le nombre réduite des machines virtuelles pouvant être utilisé pour le déploiement nous avons préférer déployer les applications en favorisant une architecture monolithe.

Graphical user interface, application

Description automatically generated

Figure 20 – déploiement dans la VM

## Installation d’Odoo

Odoo est initialement un progiciel open-source de gestion intégré comprenant de très nombreux modules permettant de répondre à de nombreux besoins de gestion des entreprises, ou de gestion de la relation client.

Dans le contexte Medicarche, l’application Odoo utilise une base de données PostgreSQL installé en local dans la machine virtuelle. Pour exposer l’application web et permettre l’accès aux collaborateurs, l’utilisation du reverse proxy ou serveur mandataire a été mise en place afin d’exposer l’application web de manière sécurisée.

Text

Description automatically generated

Figure 21 – script reverse proxy

Graphical user interface, application, Teams

Description automatically generated

Figure 21 – application Odoo

## Installation de Nextcloud

Nextcloud est un logiciel libre de site d'hébergement de fichiers et une plateforme de collaboration. La solution NextCloud apporte un haut niveau de protection et de contrôle des informations et communications dans l'entreprise. Les utilisateurs peuvent conserver toutes leurs données sur leurs serveurs internes, y compris les métadonnées.

Dans le contexte Medicarche, NextCloud utilise en local un serveur Apache avec une base de données MySQL les tout dans la machine virtuelle.  De même, nous avons mis en place un serveur mandataire pour exposer l’application web afin de la rendre accessible sur internet de manière sécurisé.

Text

Description automatically generated

Figure 22 – extrait script installation nextcloud

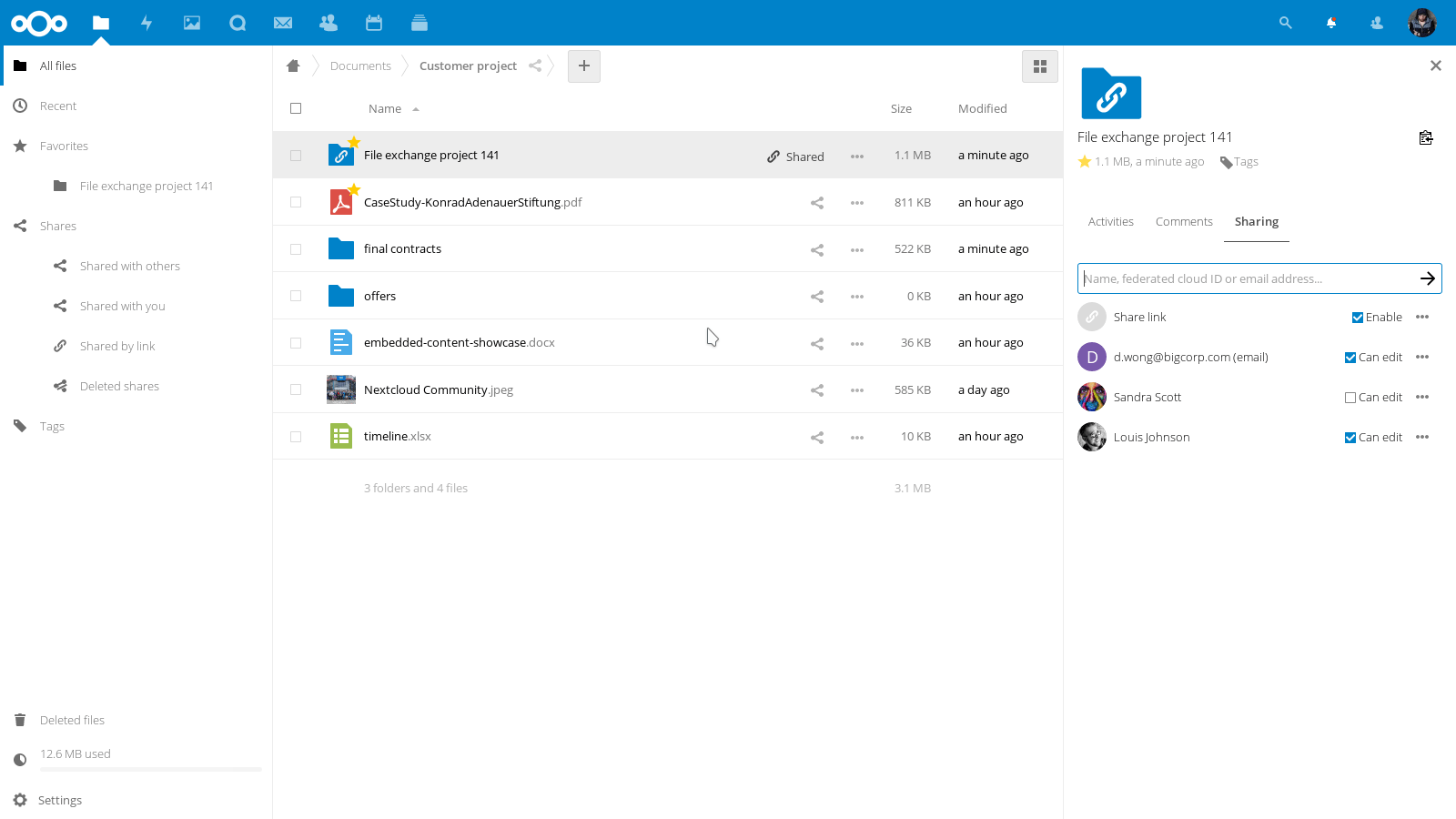


Figure 23 – interface nextcloud

## Installation de Syncthing

Syncthing est une application de synchronisation de fichiers pair à pair open source disponible pour Windows, Mac, Linux, Android, Solaris, Darwin et BSD. Aucun compte ni enregistrement préalable à l'utilisation auprès d'un tiers n'est nécessaire, ni même optionnelle.

L’application tourne localement dans la machine virtuelle en utilisant l’adresse IP 127.0.0.1. Pour avoir accès depuis l'extérieur de la machine virtuelle, on doit utiliser le reverse proxy pour pouvoir exposer l’application à tous les collaborateurs de l’entreprise Medicarche.

Text

Description automatically generated

Figure 24 – script reverse proxy syncthing

Graphical user interface, text, email

Description automatically generated

Figure 25 – interface syncthing

## Site web Medicarche

Wordpress est un système de gestion de contenu gratuit, libre et open-source. Ce logiciel écrit en PHP repose sur une base de données MySQL. Le site de l’entreprise est exposé sur internet permettant à tout le monde le désirant d’y avoir accès. Comme pour les autres applications web le site web de l’entreprise est déployé sur un serveur Apache et utilise une base de données MySQL Nous avons veillé à appliquer aussi le reverse proxy afin d’exposer le site sur internet.

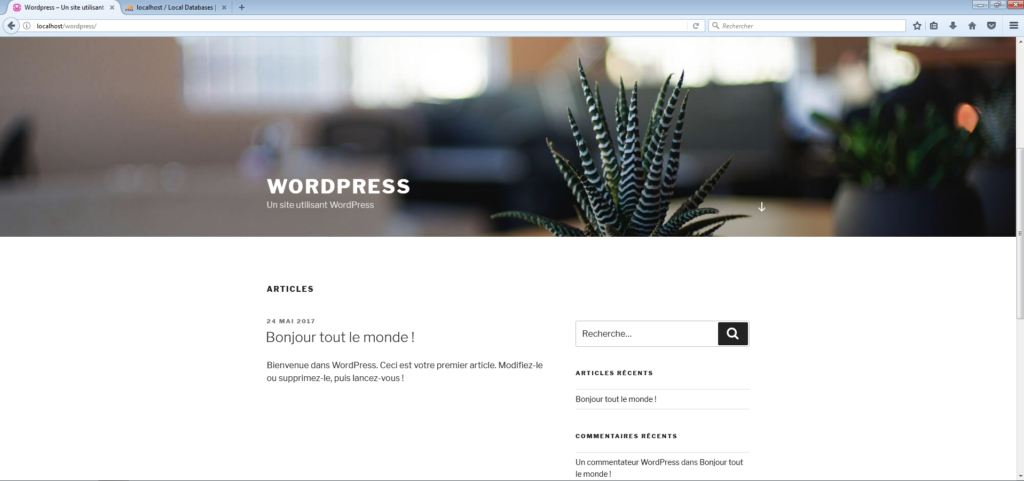


Figure 26 – site web medicarche

# Description du plan des tests

Dans le cadre du projet de déploiement il y a des points importants à effectuer des tests. Nous avons effectué un plan de test de capacité du système, un test de montée en charge des applications ou du site web de l’entreprise Medicarche et un test de stress du système que nous allons développer dans les lignes qui suivent.

## Test de montée en charge

Le test de montée en charge que nous prévoyons d’effectuer est un test au cours duquel nous aurons simulé un nombre d'utilisateurs sans cesse croissant de manière à déterminer quelle charge limite le système est capable de supporter sans tomber. Nous effectuerons un test sur l’application Odoo, Nextcloud, Syncthing et le site web de l’entreprise Medicarche. Nous prévoyons de mettre en place un script en utilisant l’outil h2load. Une fois l’utilitaire installé on aura juste à exécuter la commande suivante que l’on prévoit de le scripter pour toutes les autres applications de la plateforme Medicarche. Dans ce script nous simulons 100 clients qui produiront au total 10000 requêtes HTTP.

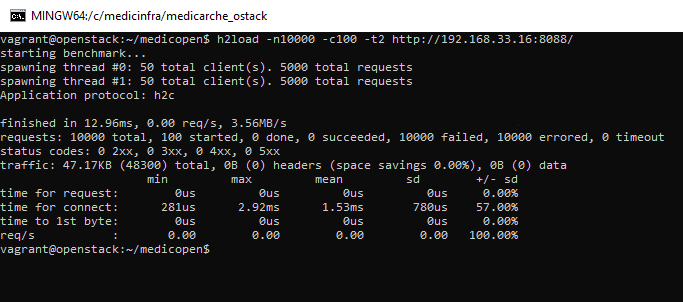


Figure 26 – simulation de la montée en charge

## Test de stress du système

Pour s’assurer que les gabarits choisis pour accueillir les applications web de l’entreprise Medicarche répondent bien au minimum des capacités requises, nous mettons en place ce type de test qui aura pour objectif de stresser le CPU, la mémoire RAM, le I/O et le disque. Pour cela nous utilisons l’outil stress and stress-ng. L’installation s'effectue en ligne de commande et on pourra voir le résultat sur l’interface CLI.

## Test de performance

Après avoir installé les applications web de l’entreprise Medicarche, on doit vérifier que le système répond normalement aux attentes du client. Pour s’assurer que c’est réellement le cas, nous avons choisi d'utiliser l’outil Tsung qui est approprié pour ce type de test. Tsung est un outil de test de performances permettant de réaliser des benchmarks massifs.

Tsung est un outil qui nous permettra de faire un rapport sur le statistique du système, des transactions, du réseau, du débit et même des statuts liés au requête HTTP.

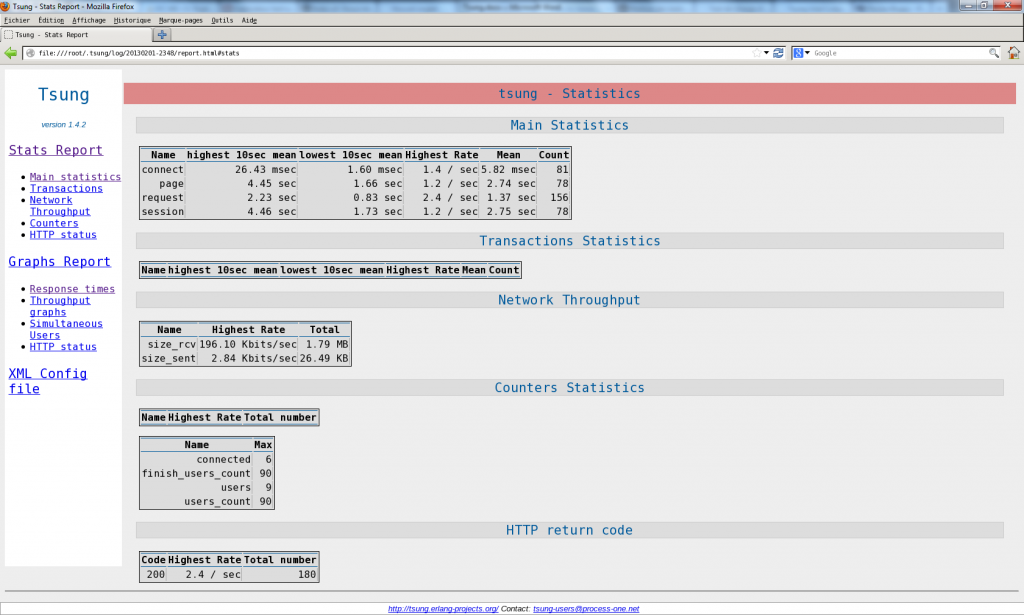


Figure 26 – tableau de bord Tsung

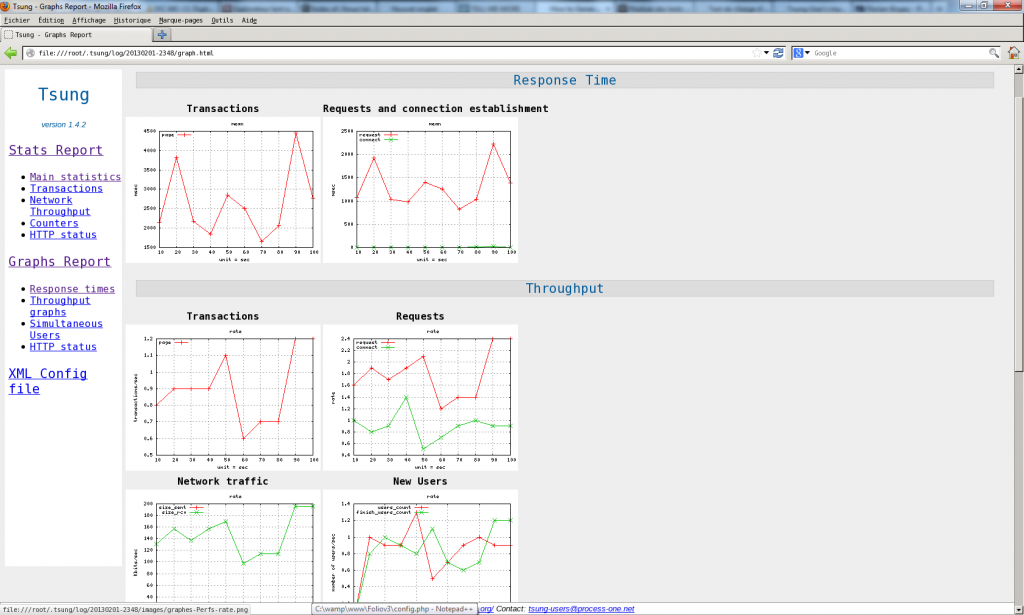


Figure 27 – résultat graphique

# Information d’accompagnement au changement

L’accompagnement au changement se fait en plusieurs étapes pour éviter toute forme de rejet auprès des collaborateurs de l’entreprise Medicarche afin d’accepter facilement le changement. En effet, les employés sont déstabilisés car ils se retrouvent hors de leur zone de confort.

L’annonce du changement de la migration de l’infrastructure Medicarche dans le cloud suscitera un éventail de réactions émotionnelles diverses :

* Le déni
* La colère
* La peur et la tristesse
* La négociation, le chantage
* La résignation

En tant que manager, il est essentiel de savoir reconnaître ces réactions, de ne pas les ignorer et d'y apporter une réponse adaptée.

La première étape est d'un changement est de communiquer en expliquant les raisons aux collaborateurs de Medicarche en étant transparent et cohérent.

Le manager doit réunir tout le personnel concerné et impacté par ce changement en exposant les différentes raisons ; les avantages de la migration dans le cloud ou du nouveau logiciel. Les bénéfices pour les salariés, sans pour autant négliger de citer les aspects moins attractifs. Une série de questions réponses sera organisée afin de répondre honnêtement aux questions des collaborateurs.

La deuxième étape est une proposition de formation qui permettra aux collaborateurs de prendre en main la nouvelle infrastructure (DSI). Certains collaborateurs refusent le changement simplement parce qu'ils craignent de ne pas être à la hauteur, ou parce qu'ils doutent de leur capacité d'adaptation. Ces craintes concernent tous les niveaux de l'entreprise, du manager à l'opérateur.

Pour y remédier, nous proposons du coaching, de l'accompagnement et de la formation aux salariés concernés. Le coaching du changement peut aider grandement le manager à motiver et à engager son équipe. Quant aux collaborateurs, ils sont accompagnés dans leur processus d'adhésion au changement.

La troisième étape est de les accompagner en encourageant les collaborateurs, reconnaître leurs efforts, être proche d’eux pour qu'ils ne se sentent pas délaissé. Un compliment est tellement plus fort qu'une critique.

La quatrième étape est de mettre en place une cellule de suivi pendant une certaine période afin de s'assurer que le système fonctionne correctement.

# Conclusion

Pour conclure, OpenStack est une bonne infrastructure open source en tant que service (IaaS) pour déployer un cloud privé, qui offre un énorme potentiel en évolutivité, en permettant à un grand nombre de nœuds interconnectés de fournir les services nécessaires. Aussi, offrant de la flexibilité en ayant des composants modulaires qui interagissent pour former l'infrastructure finale. Ces modules des composants peuvent être ajoutés ou supprimés lorsque la nécessité s'en fait sentir. Dans ce document, il montre le déploiement de OpenStack utilisant Microstack, à travers lequel on peut facilement déployer l'infrastructure. Micostack se déploie OpenStack avec des exigences système minimales et gère également le fardeau de la configuration d'OpenStack et son réseau avant le déploiement. L'intention principale de Microstack est de fournir un environnement OpenStack dans un système du développeur à des fins de test ou de développement. Microstack fait partie de Canonical et cela ne fonctionne que sur Ubuntu. Il existe divers outils disponibles qui aident au déploiement comme Devstack. Qui est fourni par OpenStack pour le déploiement. L'inconvénient est que cela prend beaucoup de temps le temps de déploiement et les configurations doivent être effectuées manuellement. Le projet nous a permis de simuler le déploiement d’une infrastructure en mettant en place l’automatisation le seul inconvénient réside au niveau de la performance de la machine hôte qui ne correspond pas à ce type de projet.

# Webographie

### *Rédiger un plan de test* -  <https://www.ingenieurtest.fr/2018/04/rediger-un-plan-de-test.html>

*Tsung* - <https://blog.admin-linux.org/administration/tsung-outils-de-benchmark-multi-protocole>

*Microstack* - <https://microstack.run/>

*Odoo* - <https://fr.wikipedia.org/wiki/Odoo>

*Nextcloud* - <https://nextcloud.com/>

*Syncthing* -<https://doc.ubuntu-fr.org/syncthing>

*Repository* *medicarche* - <https://gitlab.com/Genuiz/medicarche_ostack.git>

*Repository* *medicarche core* - <https://gitlab.com/Genuiz/medicopen.git>