

# Document de Référence

## I. Le cahier des charges

### Les besoins :

Dans une entreprise en quête d'évolution et de veille sur les dernières technologies utilisées dans le monde informatique, le client veut pouvoir détruire et créer son infrastructure le plus rapidement possible et qu'elle soit identique à chaque fois.

Pour faciliter la mise à jour des configurations, le client veut centraliser le tout sur un outil de versionning.

Dans cette infrastructure, le client veut avoir une gestion automatisée des services conteneurisé, il veut pouvoir monitorer ses serveurs et ses conteneurs.

Il veut également pouvoir gérer les utilisateurs ainsi que leurs permissions, avoir une registry pour stocker ses images docker, et pour finir, il veut sécuriser un minimum son infra avec un Haproxy.

### Les ressources et le temps

Pour ce qui est du temps, le client veut pouvoir utiliser cette infrastructure à partir du 9 Juin 2020. Ce jour-là, nous lui ferons une démo du produit, et lui apprendrons comment s'en servir. Nous avons donc 2 mois devant nous à raison de 2 jours de travail par semaines environ.

Lors de notre démo, nous allons utiliser des machines virtuelles stocker sur nos PC, cependant tout pourra être utilisé dans un environnement de production sur des machines physiques ou virtuelles lié à un ESX.

Nous utiliserons plusieurs logiciels open-sources tels qu'Ansible qui va nous permettre de faire la gestion de configuration des serveurs, Github pour gérer le versionning et stocker toute la conf sur internet, Docker pour la conteneurisation, Swarm pour le clustering avec docker, Docker Registry pour héberger les images sur un repository privé, HaProxy pour le LoadBalancing ainsi que Netdata pour la supervision des machines.

### Le coût

Tous les logiciels qu'on utilise sont open-source et donc gratuit. Les seuls coûts qui seront à la charge du client, seront les machines sur lesquels ils vont faire tourner les serveurs.

Dans cette infra, nous utiliseront 4 machines (2 pour le swarm, 1 pour la registry et 1 pour l'HaProxy). Le coût dépendra si le client fait appelle à une provider Cloud tel qu'AWS ou Azure, s'il utilise des VM dans son propre Datacenter, ou encore si met ça sur des machines physiques. Le prix peut donc varier de 5 000€ l'année à 10 000€

### Le planning :

	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24
Choix des technologies & Cahier des charges										
Schéma réseau, création du Git, installation & configuration des logiciels										
Création de la partie Users sur Ansible										
Création de la partie docker & Swarm sur Ansible										
Création de la partie Registry sur Ansible										
Création de la partie HaProxy sur Ansible										
Création et configuration Docker-Compose Netdata										
Phases de tests										
Test et Validation Final										

### La solution technique :

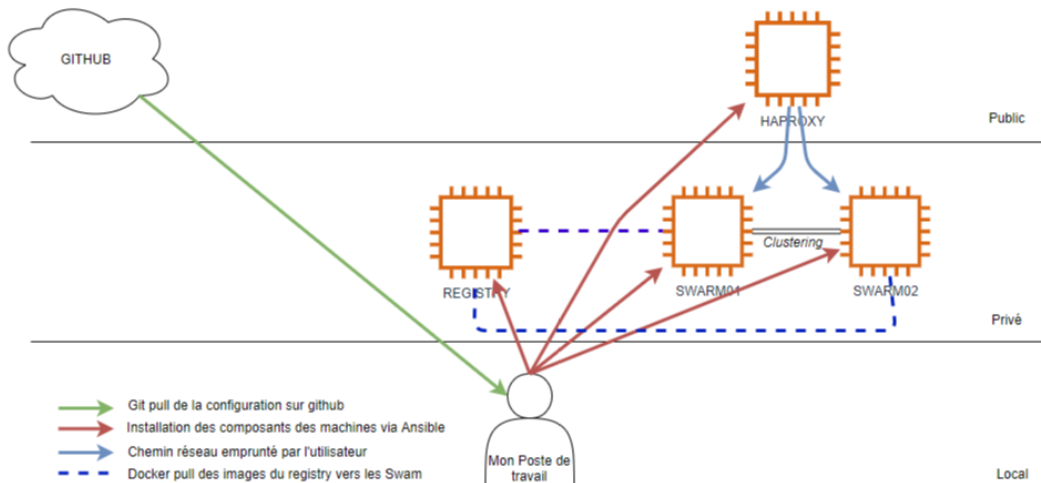
La solution est d'utiliser Ansible comme outil de gestion et d'automatisation. Tout est modifiable et simple de compréhension. Le tout stocker sur Github, il sera simple de télécharger et de déployer une nouvelle infrastructure en passant une ligne de commande. La seule installation à faire est celle d'ansible pour pouvoir déployer l'infra.

Nous utiliserons aussi un HaProxy qui va permettre le LoadBalancing entre les deux nœuds du cluster. Grâce à lui, nous pouvons laisser les deux machines non accessible directement depuis internet, donc dans le réseau privée, et l'HaProxy fera une redirection en interne. Cela est beaucoup plus sécurisant pour l'infrastructure, et va aussi permettre d'équilibrer la charge sur les différents nœuds du cluster. Il va aussi empêcher les indisponibilité de service si un des deux nœuds du cluster tombe.

Le mieux est de proposer un second nœud HaProxy qui prend la relève si le premier nœud est KO. Ceux-ci se verront configurer d'un **heartbeat** permettant de voir si l'autre nœud est HS ou non pour prendre la relève. Cependant, par manque de temps, nous ne pouvons pas mettre cette solution en place.

## II. Rapport de votre solution technique :

Architecture de la solution :



Extrait de code sur git pour la création des users :

Branch: master [infra-project / ansible / roles / users / tasks / users.yml](#) Find file Copy path

Louispautasso Add users roles for deploy ssh 98e2e4a on 4 May

1 contributor

32 lines (29 sloc) | 779 Bytes Raw Blame History

```
1 ---
2 # Tasks file for users creation
3 - name: Ensure nominative account exists
4   user:
5     name: "{{ item.name }}"
6     uid: "{{ item.uid }}"
7     shell: /bin/bash
8     group: "{{ item.group }}"
9     state: "{{ item.state }}"
10    force: no
11    remove: yes
12    with_items:
13      - "{{ project_users.users }}"
14
15 - name: Deploying their pub key
16   authorized_key:
17     user: "{{ item.name }}"
18     state: "{{ item.state }}"
19     key: "{{ item.id_key_pub | default('') }}"
20    with_items:
21      - "{{ project_users.users }}"
22    when: item.state == "present"
23
24 - name: Allowing admins members group to log as root
25   authorized_key:
26     user: root
27     state: "{{ item.state }}"
28     key: "{{ item.id_key_pub | default('') }}"
29    with_items:
30      - "{{ project_users.users }}"
31    when: item.group == "admins"
32
```

## Commande de lancement d'ansible :

```
[20-06-08 16:35]PC-FIXE: /ynov/project/infra-project/ansible@master***** lpautasso% ansible-playbook -i hosts playbook.yml
PLAY [all] *****
TASK [Gathering Facts] *****
[WARNING]: Platform linux on host swarm02 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [swarm02]
[WARNING]: Platform linux on host swarm01 is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [swarm01]
[WARNING]: Platform linux on host haproxy is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [haproxy]
[WARNING]: Platform linux on host registry is using the discovered Python interpreter at /usr/bin/python, but future installation of another Python interpreter could change this. See
https://docs.ansible.com/ansible/2.9/reference_appendices/interpreter_discovery.html for more information.
ok: [registry]
TASK [users : Ensure Sudo packages is installed] *****
[WARNING]: Updating cache and auto-installing missing dependency: python-apt
ok: [registry]
ok: [swarm01]
ok: [swarm02]
ok: [haproxy]
```

En annexe le lien du repo GIT, où tout le projet y est stocké. Les schémas, le cahier des charges, ainsi que les instructions nécessaires à l'exécution d'ansible

<https://github.com/Louispautasso/infra-project>

## III. Conclusion

Le projet nous a permis d'apprendre énormément, sur des choses qu'on ne connaissait pas forcément. Travailler sur ces technologies est un plus dans le sens où beaucoup d'entreprises partent sur techniques d'automatisation et de conteneurisation. On est dans un projet qui se situe bien dans l'ère du temps et qui sera bénéfique à l'ensemble des personnes qui ont travaillé sur ce projet.

Cependant, nous n'avons pas eu le temps de faire tout ce qu'on aurait voulu, il manquait encore un peu de temps, même si avec le confinement, nous avons eu plus de temps libre.

SUGAC Mihail

PAUTASSO Louis