

CÉLIAN JOUNIN

# Rapport de Stage BTS SIO 1<sup>ère</sup> année

Réseau et développement

**AMBITION**  
CONSULTING & SERVICES

31 Mai – 2 Juillet 2021

# Sommaire

Remerciements .....	1
Introduction .....	2
Entreprise .....	3
Activités .....	3
Organigramme .....	4
Importance de l'informatique au sein de l'entreprise.....	4
Projet.....	5
Besoin.....	5
Contexte.....	5
Choix des outils .....	5
Thème du stage .....	5
Vue globale du stage.....	6
Création de réseaux.....	6
Création d'une page HTML.....	7
Repérage du tracé d'une fibre optique .....	9
Automatisation de l'ajout et de la suppression de client via une API Rest.....	10
Détails et approfondissement du projet d'automatisation.....	10
Conclusion .....	13

# Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, je tiens à remercier vivement mon maître de stage, Mr Mehdi GHEZOUARY, ingénieur IT chez Ambition Consulting & Service, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Il m'a permis de développer de précieuses compétences notamment par l'indépendance qu'il m'a accordé mais en restant néanmoins présent dans les moments où j'avais besoin d'aide.

Je pense également à Mr Coumar SANDAMOURTHY, qui a cru en mon potentiel et m'a accueilli au sein de son entreprise.

Je remercie également toute l'équipe d'Ambition Consulting & Service pour leur accueil, leur esprit d'équipe et en particulier Mr Jean-Luc BON, qui m'a fait découvrir un autre univers du réseau : l'aspect physique de celui-ci.

Enfin, je tiens à remercier mon père qui m'a conseillé et relu lors de la rédaction de ce rapport de stage

# Introduction

Actuellement toute entreprise a besoin d'un réseau et donc d'un réseau avec un minimum de sécurité. Il faut donc des personnes à même de pouvoir leur en créer un, de pouvoir le surveiller et réagir en cas de besoin.

Dans le cadre de ma formation et dans le but de me préparer à la suite de mes études, j'ai souhaité réaliser mon stage d'une durée de 1 mois dans une entreprise qui vend des prestations en réseau et en cybersécurité. C'est pourquoi je suis allé chez Ambition Consulting & Services, une entreprise de réseau très récente au format humain fondée en 2019 mais néanmoins très performante et possédant une rapide croissance, passant de 0 à 80 employés en 3 ans. Dont le siège se trouve à Lyon 7 juste à côté de la halle Tony Garnier

J'effectue mon stage dans le département IT de l'entreprise ce qui me permet de voir plusieurs facettes dans ces domaines. Mon tuteur de stage est ingénieur IT ce qui me permet d'améliorer rapidement mes capacités auprès d'un professionnel possédant beaucoup d'expérience dans le domaine.

Dans ce rapport nous verrons dans un premier temps l'entreprise, une description de celle-ci ainsi que plus précisément quelles sont ses sources de revenus. Puis dans un second temps nous étudierons les différentes missions que j'ai pu effectuer durant ce stage.

# Entreprise

## Activités

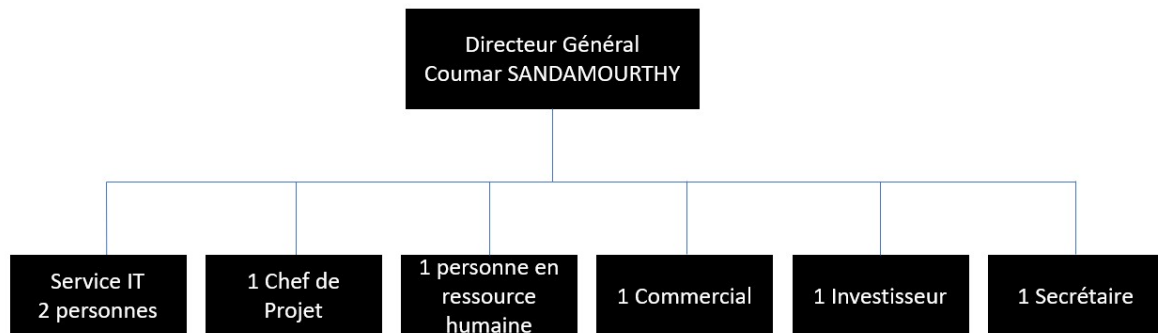
L'entreprise se situe comme un opérateur de liaisons Fibres FTTH et 4G. Elle propose comme service 4 gammes de connexions, 3 par fibre et 1 par 4G possédant toutes différentes options. L'entreprise propose également une installation d'un réseau de téléphone IP professionnel et propose sur toutes ses offres une supervision par un centre de service dédié.

Mais l'installation de la fibre et la mise en place du wifi ne sont pas les seuls services proposés par l'entreprise, elle propose aussi la mise en place de data center ainsi que des services de stockages. Elle propose des solutions de MetroCluster<sup>2</sup>, de stockage de données mais aussi de déploiement de serveur AD et d'architecture adaptée au parc ainsi que de serveurs Linux.

<sup>2</sup> Les configurations NetApp® MetroCluster™ combinent la mise en cluster de baies et la réplication synchrone pour assurer une disponibilité continue et une duplication immédiate de l'ensemble de vos données stratégiques, transaction par transaction. Les configurations MetroCluster optimisent les opérations intégrées de haute disponibilité et de continuité de l'activité du logiciel de gestion du stockage ONTAP® et du matériel NetApp. L'espace de stockage et l'environnement hôte bénéficient ainsi d'une couche de protection supplémentaire. (<https://www.netapp.com/fr/support-and-training/documentation/metrocluster/>)



# Organigramme



## Importance de l'informatique au sein de l'entreprise

L'entreprise étant une entreprise informatique cette technologie est donc omniprésente dans la société et ne sert pas juste comme support de travail mais sont les prestations réalisées par l'entreprises.

# Projet

## Besoin

Au début de mon stage, je n'ai pas réalisé de choses réellement utiles pour l'entreprise, le but étant plus de tester mon niveau et mes capacités pour l'entreprise et de valider des acquis ainsi que d'en obtenir de nouveaux pour moi. Dans un second temps j'ai réalisé des tâches profitables à l'entreprise tel que permettre la visualisation des réseaux déjà existant sur une carte. Et j'ai également contribué à l'automatisation de l'entreprise en créant des scripts.

## Contexte

En première partie de mon stage, on m'a demandé de réaliser des laboratoires réseaux avec des équipements Huawei ainsi qu'une application web permettant de faire des paris en interne dans un but ludique au sein de l'équipe.

Dans la seconde partie j'ai réalisé un travail d'appui lors du répertoriage des différents réseaux. Ainsi qu'un travail de simplification de la tâche des employés et des futurs techniciens avec la création de scripts automatisant l'ajout et la suppression des hôtes et des clients sur le logiciel de supervision de l'entreprise.

## Choix des outils

Afin de réaliser les missions qui m'ont été confiées j'ai utilisé le logiciel eNSP pour les laboratoires, une variante de Cisco Packet Tracer mais pour des équipement Huawei. Visual Studio Code combiné à WampServer64 pour réaliser la page html. Mais aussi IDLE afin de réaliser des scripts python. Ce choix pour les logiciels de codage a été fait car J'avais déjà une expérience dessus et ainsi je n'ai pas perdu de temps à découvrir l'interface. Et pour finir Géoportail dans le but de réaliser les tracés de fibre optique

## Thème du stage

Le thème du stage a été l'approfondissement de mes compétences, ainsi que la découverte du monde IT dans une petite entreprise par le biais de plusieurs petits projets.

# Déroulé des Projets

Dans cette partie je vais présenter plusieurs projets parmi tous ceux réalisés et je vais faire un focus sur l'un d'entre eux et expliquer plus en détail le déroulé de ce projet. En effet durant mon stage je n'ai pas réalisé qu'un seul gros projet mais plusieurs petits que je vais présenter dans cette partie.

## Vue globale du stage

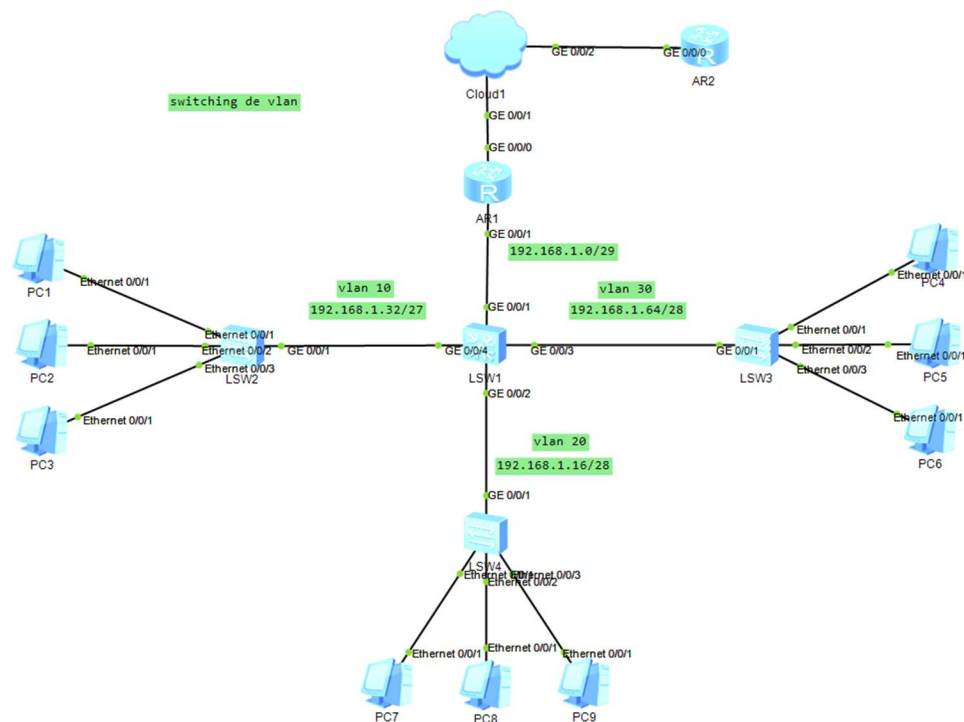
J'ai débuté mon stage par une série de labs (simulation de réseau émulé sur un logiciel) sur le logiciel eNSP. Puis j'ai développé un site internet en html CSS php. Avant d'accomplir des tracés de fibre optique. Et enfin j'ai réalisé des scripts python afin d'automatiser des saisies de données.

## Création de réseaux

Les labs eNSP n'avaient pas pour but d'être utiles à l'entreprise mais étaient faits pour jauger mon niveau ainsi que revoir une partie réseau de mon programme scolaire. La consigne m'était donnée par mon tuteur qui me disait « une entreprise a tant d'employés elle possède la plage IP XXX-XXX-XXX-XXX et veut du switching de VLAN en DHCP » ensuite c'était à moi de faire l'adressage IP de positionner les machines et de les configurer afin que le réseau fonctionne selon les critères demandés.

Voici ci-dessous un exemple d'un réseau que j'ai pu créer.





## Création d'une page HTML

Dans un deuxième temps on m'a demandé de réaliser un site de paris sportifs afin que les personnes au sein de l'entreprise puissent parier sur les résultats de l'euro avec comme récompense un t-shirt de l'équipe de France pour le gagnant.

Ci-dessous vous pouvez voir l'en tête de la page et encore en dessous les boutons afin de sélectionner les vainqueurs de chaque match.



*\*pour choisir un vainqueur cliquez sur le bouton de l'équipe*

les règles :

- Vous avez 36 points au début de vos paris. Pour chaque équipe sur laquelle vous pariez vous utilisez 1 point.

Lors de ce projet j'ai dû réaliser l'authentification ainsi que les différentes pages de navigation.

Et ainsi gérer la base de données MySQL.

Je montre ici les pages pour s'authentifier ou créer un compte.

Nom de la première équipe	affronte	Nom de la seconde équipe
Groupe A		
LA TURQUIE	contre	L'ITALIE
LE PAYS DE GALLES	contre	LA SUISSE
LA TURQUIE	contre	LE PAYS DE GALLES
L'ITALIE	contre	LA SUISSE
L'ITALIE	contre	LE PAYS DE GALLES
LA SUISSE	contre	LA TURQUIE
Groupe B		

localhost/euro/login.php

localhost/euro/login.php

Pas en cours de synchronisation

## Authentification

Votre Pseudo

Mot de passe

[Créer votre Compte](#)

localhost/euro/inscription.php x +

localhost/euro/inscription.php

Pas en cours de synchronisation

## Inscription

Jounin

Célian

Supersite

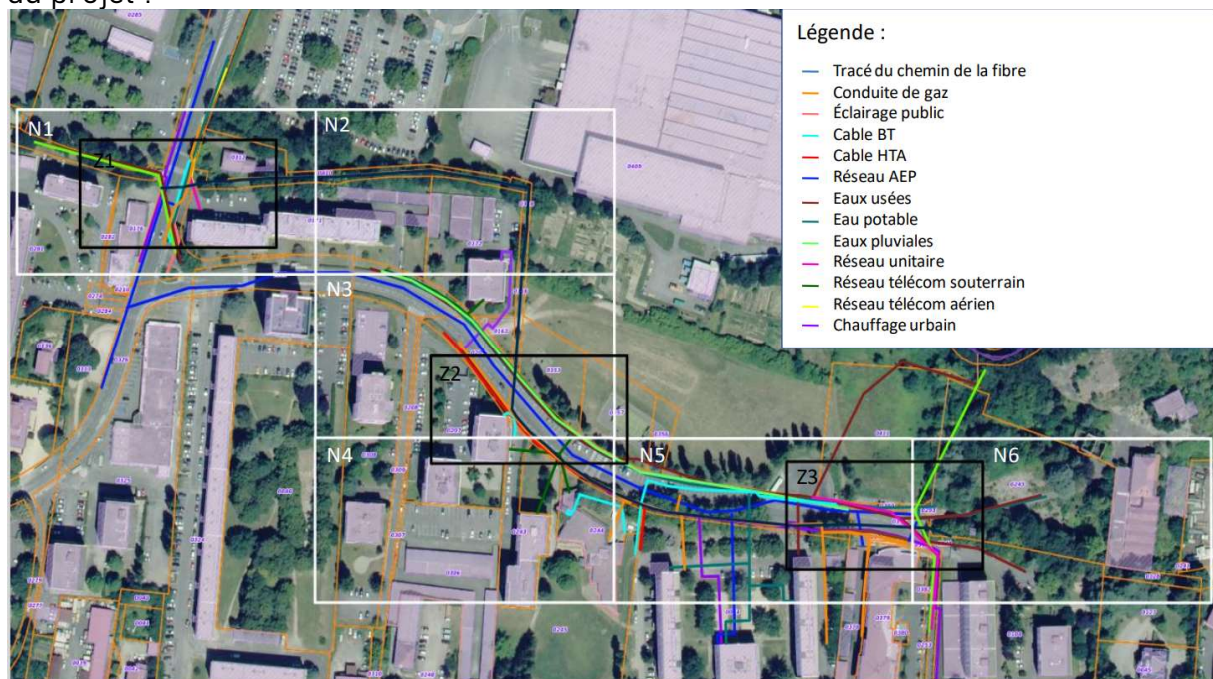
\*\*\*\*\*

\*\*\*\*\*

[Déjà un compte](#)

## Repérage du tracé d'une fibre optique

On m'a ensuite demandé de repérer et de consigner tous les réseaux coupant le tracé d'une future fibre optique. Le but étant d'indiquer à l'équipe des travaux de voirie où ils risquent de rencontrer des tuyaux et conduites. Voici un exemple du rendu global du projet :



Chaque sous-partie a été aussi enregistrée pour permettre une visualisation optimale des différents réseaux. Comme le montre l'exemple ci dessous



## Automatisation de l'ajout et de la suppression de client via une API Rest

Enfin lors de la dernière semaine de stage j'ai pu réaliser avec l'aide d'un alternant en 3<sup>ème</sup> année d'école d'ingénieur la conception de deux scripts permettant pour le premier de rajouter des clients et leurs équipements sur le site de supervision utilisé par l'entreprise et pour le second d'en supprimer. J'ai réalisé ce travail en python en utilisant IDLE et GitLab.

## Détails et approfondissement du projet d'automatisation.

Le projet a été réalisé en 3 grosses étapes :

- La compréhension de la documentation de l'API, un travail qui a mis un peu de temps car je ne connaissais pas du tout le monde des API. De plus la documentation n'était pas très claire et ne détaillait pas les différentes fonctions ainsi que leurs utilités.
- La création du premier script visant à ajouter des clients le temps mis à réaliser la tâche a été le plus important dans le projet.
- La réalisation du second script.

La temporisation du projet s'est déroulée sur 6 jours et a été découpé comme ceci :



- 1 jour et demi pour la compréhension de l'api du choix du meilleur support et du meilleur langage pour faire le script
- 3 jours pour réaliser le premier script
- 1 jour pour créer le second script
- Une demi-journée pour finaliser, rendre propre le code et le mettre sur le serveur portant l'applicatif.

La répartition du travail entre Bryan, l'alternant, et moi a été naturelle, j'ai réalisé le gros du script et Bryan m'a assisté pour tous les points bloquants cités ci-dessous et a complété le script en testant directement sur le serveur.

Lors de ce projet, plusieurs points m'ont posé des difficultés, notamment par manque de connaissances et d'exemples sur le web. Parmi les principales difficultés on peut citer :

- Comprendre l'API et de pouvoir la tester afin de voir si mes hypothèses de fonctionnement ou tests fonctionnaient. En effet je n'avais pas les droits d'accès suffisant sur mon compte pour réaliser l'ajout ou la suppression sur l'application. Ce souci à été résolu par l'arrivée de l'alternant la dernière semaine qui lui avait les permissions.
- Réaliser la connexion entre le serveur et l'applicatif, d'une part pour une authentification sécurisée et d'autre part pour l'envoi des requêtes. Néanmoins j'ai pu trouver des fragments de solutions qui mis bout à bout m'ont permis de surmonter le problème.
- Comment récupérer un code d'erreur lisible et compréhensible lorsque quelque chose n'allait pas. J'ai trouvé la solution grâce à la fonctionnalité python : `.status_code` et `.text`
- Comment demander un mot de passe caché en python. Résolution grâce à l'outil « `getpass` ».
- Et enfin comment mettre des variables dans des requêtes qui sont des chaînes de caractères.

Tout le code de ce projet est donné en annexe.



# Conclusion

Ce stage m'a permis de mettre en pratique les connaissances acquises pendant cette première année de BTS comme Packet Tracer, python ou encore HTML CSS. Ce stage m'a également permis de devenir plus autonome car j'ai pu rechercher outils et renseignements par moi-même lors de la réalisation d'un projet sur un intervalle de temps très compact.

Conjuguant réseau et développement, ce stage m'a beaucoup plu il m'a aussi permis de me situer sur ces deux technologies.

J'ai pu confirmer mon choix d'orientation car j'ai plus apprécié la partie développement, surtout une fois que j'ai pu constater la puissance des API.

# Annexe 1 : code du script d'ajout d'hôtes et de clients sur centreon

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import requests
import json
import os
import getpass

CENTREON_IP = "sup.ambitioncs.com"

#fonction de connexion
def auth_api(username, password):
    auth_data = {'username': username, 'password': password}
    r = requests.post(f'http://{CENTREON_IP}/centreon/api/index.php?action=authenticate', data=auth_data)
    a)
    if r.status_code == 200:
        print('Authentication successful.')
        print('-----')
    else:
        print('Authentication denied. %s' % r.status_code)
        exit()
    token = r.json()['authToken']

    return token

#fonction pour créer les requêtes
def send_request(data, headers):
    jsono = json.dumps(data, indent=2)
    r = requests.post(f'http://{CENTREON_IP}/centreon/api/index.php?action=action&object=centreon_clapi', data=jsono, headers=headers)
    if r.status_code != 200:
        print('Request error %s : %s' %(r.status_code, r.text))

#fonction pour ajouter un client et un hôte
def create_request(HOSTNAME, HOSTGROUP, HOST_ALIAS, IP_ADDRESS, SELECTED_TEMPLATE):
    add_host = {
        "action": "add",
        "object": "host",
        "values": f"{HOSTNAME};{HOST_ALIAS};{IP_ADDRESS};{SELECTED_TEMPLATE};central;"
    }

    apply_template = {
        "action": "applytpl",
```



```

        "object": "host",
        "values": f"{SELECTED_TEMPLATE}"
    }

    create_hostgroup = {
        "action": "add",
        "object": "hg",
        "values": f"{HOSTGROUP};{HOSTGROUP}"
    }

    set_hostgroup = {
        "action": "addhostgroup",
        "object": "host",
        "values": f"{HOSTNAME};{HOSTGROUP}"
    }

    send_request(add_host, headers)
    send_request(apply_template, headers)
    send_request(create_hostgroup, headers)
    send_request(set_hostgroup, headers)

#fonction pour ajouter un hôte
def recreate_request(HOSTNAME, HOSTGROUP, HOST_ALIAS, IP_ADDRESS, SELECTED_TEMPLATE):
    add_host = {
        "action": "add",
        "object": "host",
        "values": f"{HOSTNAME};{HOST_ALIAS};{IP_ADDRESS};{SELECTED_TEMPLATE};central;"
    }

    apply_template = {
        "action": "applytpl",
        "object": "host",
        "values": f"{SELECTED_TEMPLATE}"
    }

    set_hostgroup = {
        "action": "addhostgroup",
        "object": "host",
        "values": f"{HOSTNAME};{HOSTGROUP}"
    }

    send_request(add_host, headers)
    send_request(apply_template, headers)
    send_request(set_hostgroup, headers)

#fonction pour mettre un hôte dans des groupes d'hôtes
def hostgroups_changes():
    set_hostgroup_client = {
        "action": "addhostgroup",

```

```

        "object": "host",
        "values": f"{HOSTNAME};{CLIENT_TYPE}"
    }
    send_request(set_hostgroup_client, headers)

    if operator_inp == 'y':
        set_hostgroup_operator = {
            "action": "addhostgroup",
            "object": "host",
            "values": f"{HOSTNAME};OPERATOR_LINKS"
        }
        set_macro = {
            "action": "setmacro",
            "object": "host",
            "values": f"{HOSTNAME};PUBLICIP;{PUBLIC_IP}"
        }
        send_request(set_hostgroup_operator, headers)
        send_request(set_macro, headers)

#fonction pour redemarrer et mettre a jour le serveur
def reload_centreon():

    apply_config = {
        "action" : "APPLYCFG",
        "values" : "Central"
    }
    send_request(apply_config, headers)
    os.system('systemctl restart cbd centengine gorgoned')
    print('Reload successful. Client added.')

#-----

USERNAME = input("Entrez votre nom d'utilisateur : ")
PASSWORD = getpass.getpass(prompt = "Entrez votre mot de passe : ")

token = auth_api(USERNAME, PASSWORD)

#initialisation du header pour le token
headers = dict()
headers['content-type'] = "application/json"
headers['centreon-auth-token'] = token

#initialisation des variables
HOSTGROUP = input("Entrez le nom du client : ")
print('-----')
HOSTNAME = input("Entrez le nom de la machine : ")
HOST_ALIAS = input("Entrez le nom complet de la machine : ")
IP_ADDRESS = input("Entrez l'ip de la machine : ")

```

```

operator_inp = 'null'
while operator_inp != 'y' and operator_inp != 'n':
    operator_inp = input("Le routeur est-il un lien opérateur ? [Y/N] ").lower()
    if operator_inp == 'y':
        PUBLIC_IP = input("Entrez l'IP publique du routeur : ")

CLIENT_TYPE = "CLIENTS_HUAWEI"

print('-----')

template_nbr = 0

while template_nbr not in range(1,6):
    template_nbr = int(input("'Veuillez rentrer le nombre correspondant au modèle de l'hôte parmi les  
choix suivants :\n
    1: Generic-NewHost-Cisco
    2: Generic-NewHost-Fortigate
    3: Generic-NewHost-Huawei
    4: Generic-NewHost-Linux
    5: Generic-NewHost-Windows
    [1-5] : '"))
    if template_nbr == 1:
        SELECTED_TEMPLATE = "Generic-NewHost-Cisco"
        CLIENT_TYPE = "CLIENTS_CISCO"
    elif template_nbr == 2:
        SELECTED_TEMPLATE = "Generic-NewHost-Fortigate"
        CLIENT_TYPE = ""
    elif template_nbr == 3:
        SELECTED_TEMPLATE = "Generic-NewHost-Huawei"
        CLIENT_TYPE = "CLIENTS_HUAWEI"
    elif template_nbr == 4:
        SELECTED_TEMPLATE = "Generic-NewHost-Linux"
        CLIENT_TYPE = "ACS_VSL"
    else:
        SELECTED_TEMPLATE = "Generic-NewHost-Windows"
        CLIENT_TYPE = "ACS_VSM"

# envoi des requêtes pour enregistrer un client et un hôte
create_request(HOSTNAME, HOSTGROUP, HOST_ALIAS, IP_ADDRESS, SELECTED_TEMPLATE)
hostgroups_changes()

# récupération des informations afin de rajouter un nouvel hôte chez le même client
MULTI_EQUIP = 'null'
while MULTI_EQUIP != 'y' and MULTI_EQUIP != 'n':
    MULTI_EQUIP = input("Voulez vous ajouter un autre équipement chez le même client ? [Y/N] ").lower()
    while MULTI_EQUIP == 'y':
        HOSTNAME = input("Entrez le nom de la machine : ")
        HOST_ALIAS = input("Entrez le nom complet de la machine : ")
        IP_ADDRESS = input("Entrez l'ip de la machine : ")

```

```

operator_inp = 'null'
while operator_inp != 'y' and operator_inp != 'n':
    operator_inp = input("Le routeur est-il un lien opérateur ? [Y/N] ").lower()
    if operator_inp == 'y':
        PUBLIC_IP = input("Entrez l'IP publique du routeur : ")
print('-----')

template_nbr = 0

while template_nbr not in range(1,5):
    template_nbr = int(input('Veuillez rentrer le nombre correspondant au modèle de l'hôte pa
rmi les choix suivants :\n
1: Generic-NewHost-Cisco
2: Generic-NewHost-Fortigate
3: Generic-NewHost-Huawei
4: Generic-NewHost-Linux
5: Generic-NewHost-Windows
[1-5] : '))
    if template_nbr == 1:
        SELECTED_TEMPLATE = "Generic-NewHost-Cisco"
        CLIENT_TYPE = "CLIENTS_CISCO"
    elif template_nbr == 2:
        SELECTED_TEMPLATE = "Generic-NewHost-Fortigate"
    elif template_nbr == 3:
        SELECTED_TEMPLATE = "Generic-NewHost-Huawei"
    elif template_nbr == 4:
        SELECTED_TEMPLATE = "Generic-NewHost-Linux"
    else:
        SELECTED_TEMPLATE = "Generic-NewHost-Windows"
MULTI_EQUIP = 'null'
while MULTI_EQUIP != 'y' and MULTI_EQUIP != 'n':
    MULTI_EQUIP = input("Voulez vous encore ajouter un autre équipement chez le même client ? [
Y/N] ").lower()
# envoi des requêtes
recreate_request(HOSTNAME, HOSTGROUP, HOST_ALIAS, IP_ADDRESS, SELECTED_TEMPLATE)
hostgroups_changes()

# redémarrage du serveur
reload_centreon()

```

# Annexe 2 : suppression de clients ou d'hôtes sur centreon

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import requests
import json
import os
import getpass

CENTREON_IP = "sup.ambitioncs.com"

#fonction de connexion
def auth_api(username, password):
    auth_data = {'username': username, 'password': password}
    r = requests.post(f'http://{CENTREON_IP}/centreon/api/index.php?action=authenticate', data=auth_data)
    a)
    if r.status_code == 200:
        print('Authentication successful.')
        print('-----')
    else:
        print('Authentication denied. %s', r.status_code)
        exit()
    token = r.json()['authToken']

    return token

#fonction pour créer les requêtes
def send_request(data, headers):
    jsono = json.dumps(data, indent=2)
    r = requests.post(f'http://{CENTREON_IP}/centreon/api/index.php?action=action&object=centreon_clapi', data=jsono, headers=headers)
    if r.status_code != 200:
        print('Request error %s : %s' %(r.status_code, r.text))
    return r.json()

#fonction pour montrer tous les hotes
def create_request(CLIENT_NAME):
    show_host_in_hostgroup = {
        "action": "getmember",
        "object": "hg",
        "values": f"{CLIENT_NAME}"
    }
    result = send_request(show_host_in_hostgroup, headers)
    return result
```

```

#fonction pour supprimer un équipement
def create_request2(HOSTNAME):
    del_host = {
        "action": "del",
        "object": "host",
        "values": f"{HOSTNAME}"
    }
    send_request(del_host, headers)

#fonction pour supprimer un groupe d'hôte
def create_request3(CLIENT_NAME):
    del_hostgroup = {
        "action": "del",
        "object": "hg",
        "values": f"{CLIENT_NAME}"
    }
    send_request(del_hostgroup, headers)

#fonction pour redemarrer et mettre à jour le serveur
def reload_centreon():

    apply_config = {
        "action": "APPLYCFG",
        "values": "Central"
    }
    send_request(apply_config, headers)
    os.system('systemctl restart cbd centengine gorgoned')
    print('Reload successful. Client and Host deleted.')

#-----

USERNAME = input("Entrez votre nom d'utilisateur : ")
PASSWORD = getpass.getpass(prompt = "Entrez votre mot de passe : ")

#connexion
token = auth_api(USERNAME, PASSWORD)

#initialisation du header pour le token
headers = dict()
headers['content-type'] = "application/json"
headers['centreon-auth-token'] = token

#choix de l'option de suppression
choix = 0
while choix != 1 and choix != 2:
    choix = int(input('Veuillez rentrer le nombre correspondant à ce que vous voulez supprimer :\n
    1: Un Client et tous ses hôtes
    2: Un hôte

```

```

[1-2] : '')))

#suppression d'un client
if choix == 1:
    CLIENT_NAME = input("Entrez le nom du client dont vous voulez supprimer les machines : ")
    retour = create_request(CLIENT_NAME)
    validation = 'null'
    while validation != 'y' and validation != 'n':
        print ('Nom du client : '+CLIENT_NAME+"\n", 'Liste des hôtes : ', [name['name'] for name in retour.get("result")])
        validation = input("Etes vous sur de vouloir supprimer ce client ? [Y/N] ").lower()
    if validation == 'y':
        retour1 = retour.get("result")
        for i in range (len (retour1)):
            retour2 = retour1[i]
            HOSTNAME = retour2.get("name")
            create_request2(HOSTNAME)

        create_request3(CLIENT_NAME)
        reload_centreon()

#suppression d'un hôte
else :
    HOSTNAME = input("Entrez le nom de la machine que vous voulez supprimer : ")
    validation = 'null'
    while validation != 'y' and validation != 'n':
        print (HOSTNAME)
        validation = input("Etes vous sur de vouloir supprimer cet hôte ? [Y/N] ").lower()
    if validation == 'y':
        create_request2(HOSTNAME)
        # redémarrage du serveur
        reload_centreon()

```