

## TP1 : Programme qui liste tous les fichiers d'un répertoire donné

```
liste_fichiers_repertoire.py X
liste_fichiers_repertoire.py > ...
1 import os
2 from pathlib import Path
3
4 def list_files(directory):
5     try:
6         # Utiliser os pour lister les fichiers
7         files_os = [f for f in os.listdir(directory) if os.path.isfile(os.path.join(directory, f))]
8         print(f"Fichiers dans le répertoire '{directory}' (utilisant os):")
9         for file in files_os:
10             print(file)
11
12         print('-----')
13
14         # Utiliser pathlib pour lister les fichiers
15         p = Path(directory)
16         files_pathlib = [f for f in p.iterdir() if f.is_file()]
17         print(f"Fichiers dans le répertoire '{directory}' (utilisant pathlib):")
18         for file in files_pathlib:
19             print(file.name)
20     except Exception as e:
21         print(f"Erreur : {e}")
22
23 # Exemple d'utilisation avec un chemin réel
24 directory = 'C:\\Users\\Frank Landry\\Documents'
25 list_files(directory)
26
```

## Résultat

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
IMG-20240912-WA0006.jpg
IMG-20240912-WA0007.jpg
Melchi_Kembo_-_Epana_Yo__Clip_Officiel__(128k).m4a
protocole réseau tp(EDONGO AMENGNE).pdf
protocole réseau tp.rtf
public void Table-1-2-1-1.docx
public void Table-1.docx
~$otocole réseau tp.rtf
~$RL0001.tmp
~$RL1357.tmp
-----
Fichiers dans le répertoire 'C:\Users\Frank Landry\Documents' (utilisant pathlib):
1728776435643.jpg
1728776601696.jpg
1728777001329.jpg
1a5b2578-1b49-4127-becc-3c2be980c20f.webp
chapeau-50-ans-anniversaire-noir-or-3609810084926.webp
Commande pour creer les VLAN.txt
desktop.ini
IMG-20240912-WA0006.jpg
IMG-20240912-WA0007.jpg
Melchi_Kembo_-_Epana_Yo__Clip_Officiel__(128k).m4a
protocole réseau tp(EDONGO AMENGNE).pdf
protocole réseau tp.rtf
public void Table-1-2-1-1.docx
public void Table-1.docx
~$otocole réseau tp.rtf
~$RL0001.tmp
~$RL1357.tmp
PS C:\Users\Frank Landry\Desktop\Controle continu>
```

## TP 2 : Script qui exécute une commande système et récupère sa sortie

```
exécution_commande_système.py X
exécution_commande_système.py > ...
1 import subprocess
2
3 def list_processes():
4     # Utiliser tasklist pour lister les processus actifs sur Windows
5     result = subprocess.run(['tasklist'], capture_output=True, text=True)
6     print(result.stdout)
7
8 list_processes()
9
```

### Résultat

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
svchost.exe      1012 Services      0      8y520 Ko
taskhostw.exe    2600 Console        1     17y888 Ko
MicTray64.exe    6964 Console        1      5y352 Ko
svchost.exe      7024 Services      0      8y020 Ko
ctfmon.exe       6160 Console        1     21y276 Ko
PS C:\Users\Frank Landry\Desktop\Controle continu>
```

## TP 3 : Interception des signaux

```
gestion_des_processus.py  détection_des_changement_dans_un_repertoire.py  Interception_de_signal.py X  liste_fichiers_repertoire.py > v
Interception_de_signal.py > ...
1 import signal
2 import time
3
4 def handle_sigint(signum, frame):
5     print("Signal d'interruption (SIGINT) reçu. Terminaison propre du programme.")
6     exit(0)
7
8 signal.signal(signal.SIGINT, handle_sigint)
9
10 print("Appuyez sur Ctrl+C pour interrompre le programme.")
11 while True:
12     time.sleep(1)
13
```

## Résultat

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
..\debugpy\launcher '61140' '--' 'C:\Users\Frank Landry\Desktop\Controle continu\détection_des_changement_dans_un_repertoire.py'
PS C:\Users\Frank Landry\Desktop\Controle continu> ^C
PS C:\Users\Frank Landry\Desktop\Controle continu>
PS C:\Users\Frank Landry\Desktop\Controle continu> c:: cd 'c:\Users\Frank Landry\Desktop\Controle continu'; & 'c:\Python\Python
313\python.exe' 'c:\Users\Frank Landry\.vscode\extensions\ms-python.debugpy-2024.12.0-win32-x64\bundle\libs\debugpy\adapter\../
..\debugpy\launcher' '61145' '--' 'C:\Users\Frank Landry\Desktop\Controle continu\Interception_de_signal.py'
Appuyez sur Ctrl+C pour interrompre le programme.
Signal d'interruption (SIGINT) reçu. Terminaison propre du programme.
PS C:\Users\Frank Landry\Desktop\Controle continu> 
```

## TP 4 : Gestion des processus

```
gestion_des_processus.py X
gestion_des_processus.py > ...
1 import multiprocessing
2 import psutil
3 import time
4
5 def worker():
6     print('Processus en arrière-plan en cours...')
7     time.sleep(30) # Augmenter le temps pour observer
8
9 if __name__ == '__main__':
10     # Lancer le processus en arrière-plan
11     p = multiprocessing.Process(target=worker)
12     p.start()
13
14     # Attendre un instant pour s'assurer que le processus est en cours d'exécution
15     time.sleep(1)
16
17     # Lister les processus enfants
18     parent = psutil.Process(p.pid)
19     children = parent.children(recursive=True)
20     print(f'Processus enfant : {[child.pid for child in children]}')
21
22     # Afficher tous les processus pour vérifier
23     print("Tous les processus en cours :")
24     for proc in psutil.process_iter(['pid', 'name']):
25         print(proc.info)
26
27     # Arrêter le processus enfant
28     p.terminate()
29
```

```
PS C:\Users\Frank Landry\Desktop\Controle continu> c:: cd 'c:\Users\Frank Landry\Desktop\Controle continu'; & 'c:\Python\Python
313\python.exe' 'c:\Users\Frank Landry\.vscode\extensions\ms-python.debugpy-2024.12.0-win32-x64\bundle\libs\debugpy\adapter\../
..\debugpy\launcher' '64077' '--' 'C:\Users\Frank Landry\Desktop\Controle continu\gestion_des_processus.py'
Processus en arrière-plan en cours...
Processus enfant : []
Tous les processus en cours :
{'pid': 0, 'name': 'System Idle Process'}
{'pid': 4, 'name': 'System'}
{'pid': 100, 'name': 'Registry'}
{'pid': 456, 'name': 'smss.exe'}
{'pid': 556, 'name': 'svchost.exe'}
{'pid': 564, 'name': 'fontdrvhost.exe'}
{'pid': 616, 'name': 'csrss.exe'}
{'pid': 700, 'name': 'wininit.exe'}
{'pid': 720, 'name': 'csrss.exe'}
{'pid': 724, 'name': 'svchost.exe'}
{'pid': 772, 'name': 'services.exe'}
{'pid': 780, 'name': 'lsass.exe'}
{'pid': 820, 'name': 'igfxEM.exe'}
{'pid': 860, 'name': 'winlogon.exe'}
{'pid': 908, 'name': 'svchost.exe'}
{'pid': 972, 'name': 'svchost.exe'}
{'pid': 992, 'name': 'fontdrvhost.exe'}
{'pid': 1012, 'name': 'svchost.exe'}
{'pid': 1140, 'name': 'svchost.exe'}
{'pid': 1184, 'name': 'IntelCpHDCPSvc.exe'}
{'pid': 1204, 'name': 'svchost.exe'}
{'pid': 1208, 'name': 'dwm.exe'}
{'pid': 1220, 'name': 'svchost.exe'}
{'pid': 1228, 'name': 'svchost.exe'}
{'pid': 1236, 'name': 'OneDrive.exe'}
{'pid': 1252, 'name': 'svchost.exe'}
{'pid': 1360, 'name': 'SecurityHealthSystray.exe'}
```

## TP5 : Détection des changements dans un répertoire

```
détection_des_changement_dans_un_repertoire.py x ping_script_auto.py Serveur_TCP_simple.py
détection_des_changement_dans_un_repertoire.py > ...
1 import time
2 from watchdog.observers import Observer
3 from watchdog.events import FileSystemEventHandler
4
5 class MyHandler(FileSystemEventHandler):
6     def on_modified(self, event):
7         print(f'Fichier modifié: {event.src_path}')
8
9     def on_created(self, event):
10        print(f'Fichier créé: {event.src_path}')
11
12    def on_deleted(self, event):
13        print(f'Fichier supprimé: {event.src_path}')
14
15 if __name__ == "__main__":
16     path = "C:\\Users\\Frank Landry\\Desktop\\teste" # Remplacez par le chemin de votre répertoire
17     event_handler = MyHandler()
18     observer = Observer()
19
20     observer.schedule(event_handler, path=path, recursive=False)
21     observer.start()
22
23     try:
24         while True:
25             time.sleep(1) # Maintient le programme en cours d'exécution
26     except KeyboardInterrupt:
27         observer.stop()
28     observer.join()
29
```

### Résultat

```
C:\Windows\system32\cmd.exe - Python détection_des_changement_dans_un_repertoire.py
Microsoft Windows [version 10.0.19045.5131]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>CD C:\Users\Frank Landry\Desktop\Controle continu>
La syntaxe de la commande n'est pas correcte.

C:\Users\Frank Landry>cd C:\Users\Frank Landry\Desktop\Controle continu
C:\Users\Frank Landry\Desktop\Controle continu>Python détection_des_changement_dans_un_repertoire.py
Fichier créé: C:\Users\Frank Landry\Desktop\teste\Nouveau document texte.txt
Fichier supprimé: C:\Users\Frank Landry\Desktop\teste\hj.txt
Fichier créé: C:\Users\Frank Landry\Desktop\teste\Nouveau document texte.txt
Fichier supprimé: C:\Users\Frank Landry\Desktop\teste\Doc Msr DIPOKO.txt
```

## TP6 : Création d'un serveur TCP simple

```
Serveur_TCP_simple.py • Client_TCP_qui_se_connecte_au_serveur_TCP.py
Serveur_TCP_simple.py > tcp_server
1 import socket
2
3 def tcp_server():
4     server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5     server_socket.bind(('localhost', 12346))
6     server_socket.listen(5)
7     print("Serveur TCP en écoute sur le port 12346")
8
9     while True:
10        client_socket, addr = server_socket.accept()
11        print(f'Connexion de {addr}')
12        message = client_socket.recv(1024).decode()
13        print(f'Reçu: {message}')
14        client_socket.send("Message reçu".encode())
15        client_socket.close() # Fermer la connexion après avoir envoyé la réponse
16
17 tcp_server()
18
19
20
21
```

## Résultat

```
C:\Windows\system32\cmd.exe - python Serveur_TCP_simple.py
Microsoft Windows [version 10.0.19045.5011]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>cd "C:\Users\Frank Landry\Desktop\Controle continu"
C:\Users\Frank Landry\Desktop\Controle continu>c:
C:\Users\Frank Landry\Desktop\Controle continu>cd "C:\Users\Frank Landry\Desktop\Controle continu"
C:\Users\Frank Landry\Desktop\Controle continu>python Serveur_TCP_simple.py
Serveur TCP en écoute sur le port 12346
```

## TP7 : Client TCP se connectant au serveur TCP

```
Client_TCP_qui_se_connecte_au_serveur_TCP.py > tcp_client
1 import socket
2
3 def tcp_client():
4     client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5     client_socket.connect(('localhost', 12346))
6     client_socket.send("Bonjour, serveur!".encode())
7     response = client_socket.recv(1024).decode()
8     print(f'Reçu du serveur: {response}')
9     client_socket.close()
10
11 tcp_client()
12
```

## Résultats

```
C:\Windows\system32\cmd.exe - python Serveur_TCP_simple.py
Microsoft Windows [version 10.0.19045.5011]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>cd "C:\Users\Frank Landry\Desktop\Controle continu"
C:\Users\Frank Landry\Desktop\Controle continu>c:
C:\Users\Frank Landry\Desktop\Controle continu>cd "C:\Users\Frank Landry\Desktop\Controle continu"
C:\Users\Frank Landry\Desktop\Controle continu>python Serveur_TCP_simple.py
Serveur TCP en écoute sur le port 12346
Connexion de ('127.0.0.1', 57390)
Reçu: Bonjour, serveur!
```



## Résultats

C:\Windows\system32\cmd.exe - Python Implémentation\_serveur\_UDP.py

Microsoft Windows [version 10.0.19045.5011]  
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>cd "C:\Users\Frank Landry\Desktop\Controle continu"

C:\Users\Frank Landry\Desktop\Controle continu>Python Implémentation\_serveur\_UDP.py  
Serveur UDP en écoute sur le port 12345

C:\Windows\system32\cmd.exe - Python Implémentation\_serveur\_UDP.py

Microsoft Windows [version 10.0.19045.5011]  
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>cd "C:\Users\Frank Landry\Desktop\Controle continu"

C:\Users\Frank Landry\Desktop\Controle continu>Implémentation\_serveur\_UDP.py

C:\Users\Frank Landry\Desktop\Controle continu>  
Implémentation\_serveur\_UDP.py

C:\Users\Frank Landry\Desktop\Controle continu>

C:\Users\Frank Landry\Desktop\Controle continu>Python Implémentation\_serveur\_UDP.py  
Serveur UDP en écoute sur le port 12345  
Reçu de ('127.0.0.1', 63534): Bonjour, serveur!

C:\Windows\system32\cmd.exe

Microsoft Windows [version 10.0.19045.5011]  
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>cd "C:\Users\Frank Landry\Desktop\Controle continu"

C:\Users\Frank Landry\Desktop\Controle continu>Python testons\_le\_serveur\_UDP.py  
Reçu du serveur: Message reçu

C:\Users\Frank Landry\Desktop\Controle continu>

## TP 9 : Téléchargement des données depuis une URL

telechargement\_des\_données\_depuis\_une\_URL.py X

telechargement\_des\_données\_depuis\_une\_URL.py > ...

```
1 import requests
2
3 def download_content(url):
4     response = requests.get(url)
5     print(response.text)
6
7 download_content('https://facebook.com')
8
```

## Résultat





## Création du fichier du client

```
file_upload_client.py x ping_script_auto.py Serveur_TCP_simple.py
uploads > file_upload_client.py > upload_file
1 import socket
2 import os
3
4 def upload_file(filename):
5     host = 'localhost' # Remplacez par l'adresse IP du serveur si nécessaire
6     port = 5000
7     buffer_size = 1024
8
9     # Création du socket du client
10    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
11    client_socket.connect((host, port))
12
13    # Envoyer le nom du fichier
14    client_socket.send(os.path.basename(filename).encode('utf-8'))
15
16    # Envoyer le fichier
17    with open(filename, 'rb') as f:
18        print(f"Envoi du fichier {filename}...")
19        while True:
20            data = f.read(buffer_size)
21            if not data:
22                break
23            client_socket.send(data)
24
25    print(f"Fichier {filename} envoyé.")
26    client_socket.close()
27
28 if __name__ == "__main__":
29     file_to_upload = input("Entrez le chemin du fichier à uploader: ")
30     upload_file(file_to_upload)
31
```

## Résultats

```
C:\Users\Frank Landry\Desktop\Controle continu>cd uploads
C:\Users\Frank Landry\Desktop\Controle continu\uploads>Python file_upload_server.py
Serveur démarré sur 0.0.0.0:5000. En attente de connexions...
Connexion établie avec ('127.0.0.1', 49717)
Réception du fichier COURS.TXT...
Fichier COURS.TXT reçu et enregistré dans uploads.
```

← → ▾ ▴ 📁 > Controle continu > uploads > uploads				
^				
Nom		Modifié le	Type	Taille
📄 COURS		27/11/2024 02:58	Document texte	0 Ko
📄 mon_fichier		26/11/2024 23:47	Document texte	0 Ko
📄 mon_fichier.txt		26/11/2024 23:52	Fichier TXT	0 Ko

## TP 11 : Application de chat simple

Nous allons créer deux fichiers, un pour le chat server et l'autre pour le chat client.

```
chat_server.py x ping_script_auto.py Serveur_TCP_simple.py
chat_server.py > handle_client
1 import socket
2 import threading
3
4 def handle_client(client_socket, clients):
5     while True:
6         msg = client_socket.recv(1024)
7         if not msg:
8             break
9         for client in clients:
10            if client != client_socket:
11                client.send(msg)
12            clients.remove(client_socket)
13            client_socket.close()
14
15 def start_server():
16     server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
17     server.bind(('0.0.0.0', 7777))
18     server.listen(5)
19     clients = []
20
21     print("Serveur de chat démarré, en attente de connexions...")
22
23     while True:
24         client_socket, addr = server.accept()
25         clients.append(client_socket)
26         print(f"Client connecté: {addr}")
27         threading.Thread(target=handle_client, args=(client_socket, clients)).start()
28
29 if __name__ == "__main__":
30     start_server()
31
```

```
chat_client.py x ping_script_auto.py Serveur_TCP_simple.py
chat_client.py > receive_messages
2 import threading
3
4 def receive_messages(client_socket):
5     while True:
6         try:
7             message = client_socket.recv(1024).decode('utf-8')
8             if message:
9                 print(message)
10            else:
11                break
12        except:
13            print("Erreur de connexion au serveur.")
14            break
15
16 def start_client():
17     client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
18     client.connect(('localhost', 7777)) # Remplacez 'localhost' par l'adresse IP du serveur si nécessaire
19
20     threading.Thread(target=receive_messages, args=(client,), daemon=True).start()
21
22     while True:
23         message = input()
24         if message.lower() == 'exit':
25             break
26         client.send(message.encode('utf-8'))
27
28     client.close()
29
30 if __name__ == "__main__":
31     start_client()
32
```

## Résultat

C:\Windows\system32\cmd.exe - python chat\_server.py

```
Microsoft Windows [version 10.0.19045.5131]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>cd C:\Users\Frank Landry\Desktop\Controle continu

C:\Users\Frank Landry\Desktop\Controle continu>python chat_server.py
Serveur de chat démarré, en attente de connexions...
Client connecté: ('127.0.0.1', 49722)
Client connecté: ('127.0.0.1', 49723)
Client connecté: ('127.0.0.1', 49724)
```

```
ca. Sélection C:\Windows\system32\cmd.exe - python chat_client.py
Microsoft Windows [version 10.0.19045.5131]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>cd C:\Users\Frank Landry\Desktop\Controle continu

C:\Users\Frank Landry\Desktop\Controle continu>python chat_client.py
Bonsoir mon ami EDONGO
Oui bonsoir comment tu vas
bien et toi ?
Bien bien
```

## TP 12 : Script de ping automatique

```
chat_server.py ping_script_auto.py X Serveur_TCP_simple.py
ping_script_auto.py > ...
1 import os
2 import time
3
4 def ping(host):
5     # Exécute la commande ping
6     response = os.popen(f"ping -c 1 {host}").read()
7     return response
8
9 def log_results(host, log_file):
10     while True:
11         result = ping(host)
12         with open(log_file, 'a') as f:
13             f.write(f"{time.strftime('%Y-%m-%d %H:%M:%S')} - {result}\n")
14         print(f"Ping effectué pour {host}, résultat enregistré.")
15         time.sleep(60) # Attendre 60 secondes avant le prochain ping
16
17 if __name__ == "__main__":
18     target_host = input("Entrez l'adresse IP ou le nom de domaine à pinguer : ")
19     log_file_path = "ping_results.txt"
20     log_results(target_host, log_file_path)
21
```

## Résultats

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Users\Frank Landry\Desktop\Controle continu\ping_script_auto.py'
Entrez l'adresse IP ou le nom de domaine à pinguer : 8.8.8.8
Ping effectué pour 8.8.8.8, résultat enregistré.
Ping effectué pour 8.8.8.8, résultat enregistré.
Ping effectué pour 8.8.8.8, résultat enregistré.

```

ping_results - Bloc-notes						
Fichier	Edition	Format	Affichage	Aide		
Tue Nov 26	22:32:18	2024:	facebook.com	is	down	
Tue Nov 26	22:33:18	2024:	facebook.com	is	down	
Tue Nov 26	22:34:18	2024:	facebook.com	is	down	
Tue Nov 26	22:35:18	2024:	facebook.com	is	down	
Tue Nov 26	22:36:18	2024:	facebook.com	is	down	
Tue Nov 26	22:37:18	2024:	facebook.com	is	down	
Tue Nov 26	22:38:18	2024:	facebook.com	is	down	
Tue Nov 26	22:39:18	2024:	facebook.com	is	down	
Tue Nov 26	22:40:18	2024:	facebook.com	is	down	
Tue Nov 26	22:41:18	2024:	facebook.com	is	down	
Tue Nov 26	22:42:18	2024:	facebook.com	is	down	
Tue Nov 26	22:43:18	2024:	facebook.com	is	down	
Tue Nov 26	22:44:18	2024:	facebook.com	is	down	
Tue Nov 26	22:45:18	2024:	facebook.com	is	down	
Tue Nov 26	22:46:18	2024:	facebook.com	is	down	
Tue Nov 26	22:47:18	2024:	facebook.com	is	down	
Tue Nov 26	22:48:19	2024:	facebook.com	is	down	
Tue Nov 26	22:49:19	2024:	facebook.com	is	down	
Tue Nov 26	22:50:19	2024:	facebook.com	is	down	

## **TP 13 : Journalisation des activités réseau**

```

chat_server.py ping_script_auto.py X Serveur_TCP_simple.py
ping_script_auto.py > ...
1 import os
2 import time
3
4 def ping(host):
5     # Exécute la commande ping
6     response = os.popen(f"ping -c 1 {host}").read()
7     return response
8
9 def log_results(host, log_file):
10    while True:
11        result = ping(host)
12        with open(log_file, 'a') as f:
13            f.write(f"{time.strftime('%Y-%m-%d %H:%M:%S')} - {result}\n")
14        print(f"Ping effectué pour {host}, résultat enregistré.")
15        time.sleep(60) # Attendre 60 secondes avant le prochain ping
16
17 if __name__ == "__main__":
18     target_host = input("Entrez l'adresse IP ou le nom de domaine à pinguer : ")
19     log_file_path = "ping_results.txt"
20     log_results(target_host, log_file_path)
21

```

## Résultat

[illegible]

## ETUDE DE CAS

## TP14 : Création d'un système de gestion de processus distribué

Pour cela nous allons créer un server pour qui vas enregistrer les taches et les distribués à plusieurs clients pour le traitement.

```
server.py x client.py
création_d'un_système_de_gestion_de_processus_distribué > server.py > handle_client
1 import socket
2 import threading
3
4 # Liste pour stocker les résultats des tâches
5 results = []
6
7 def handle_client(client_socket):
8     while True:
9         # Recevoir la tâche du client
10        task = client_socket.recv(1024).decode('utf-8')
11        if not task:
12            break
13
14        print(f"Tâche reçue: {task}")
15
16        # Simuler le traitement de la tâche (ici, juste une réponse)
17        result = f"Résultat de '{task}'"
18        results.append(result)
19
20        # Envoyer le résultat au client
21        client_socket.send(result.encode('utf-8'))
22
23    client_socket.close()
24
25 def start_server(host='localhost', port=12345):
26     server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
27     server_socket.bind((host, port))
28     server_socket.listen(5)
29     print(f'Serveur en écoute sur {host}:{port}')
30
31     while True:
32         client_socket, addr = server_socket.accept()
33         print(f'Connexion de {addr}')
34         client_handler = threading.Thread(target=handle_client, args=(client_socket,))
35         client_handler.start()
36
37 if __name__ == "__main__":
38     start_server()
39
```

```
server.py Communication_entre_un_client_et_un_serveur_sur_réseau_local.py client.py x
création_d'un_système_de_gestion_de_processus_distribué > client.py > send_task
1 import socket
2
3 def send_task(task, host='localhost', port=12345):
4     client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5     client_socket.connect((host, port))
6
7     # Envoyer la tâche au serveur
8     client_socket.send(task.encode('utf-8'))
9
10    # Recevoir le résultat du serveur
11    result = client_socket.recv(1024).decode('utf-8')
12    print(f'Résultat reçu: {result}')
13
14    client_socket.close()
15
16 if __name__ == "__main__":
17     # Exemple de tâches à envoyer
18     tasks = ["Tâche 1", "Tâche 2", "Tâche 3"]
19
20     for task in tasks:
21         send_task(task)
22
```

## Résultats

```
C:\Windows\system32\cmd.exe - python server.py
```

```
Microsoft Windows [version 10.0.19045.5131]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>cd C:\Users\Frank Landry\Desktop\Controle continu

C:\Users\Frank Landry\Desktop\Controle continu>cd C:\Users\Frank Landry\Desktop\Controle continu\création_d'un_système_de_gestion_de_processus_distribué

C:\Users\Frank Landry\Desktop\Controle continu\création_d'un_système_de_gestion_de_processus_distribué>python server.py
Serveur en écoute sur localhost:12345
Connexion de ('127.0.0.1', 49683)
Tâche reçue: Tâche 1
Connexion de ('127.0.0.1', 49684)
Tâche reçue: Tâche 2
Connexion de ('127.0.0.1', 49685)
Tâche reçue: Tâche 3
```

```
C:\Windows\system32\cmd.exe
```

```
Microsoft Windows [version 10.0.19045.5131]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Frank Landry>cd C:\Users\Frank Landry\Desktop\Controle continu\création_d'un_système_de_gestion_de_processus_distribué

C:\Users\Frank Landry\Desktop\Controle continu\création_d'un_système_de_gestion_de_processus_distribué>python client.py
Resultat reçu: Résultat de 'Tâche 1'
Resultat reçu: Résultat de 'Tâche 2'
Resultat reçu: Résultat de 'Tâche 3'

C:\Users\Frank Landry\Desktop\Controle continu\création_d'un_système_de_gestion_de_processus_distribué>
```

## TP : INSTALLATION ET DEPLOIEMENT D'ACTIVE DIRECTORY

### Installation de Windows server 2022



Crée une machine virtuelle

## Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Nom : windows Server

Folder : C:\Users\landr\VirtualBox VMs

ISO Image : C:\Users\landr\OneDrive\Images\Documents\SERVER\_EVAL\_x64FRE\_fr-fr.iso

Edition : Windows Server 2022 Standard Evaluation (10.0.20348.587 / x64 / fr-FR)

Type : Microsoft Windows

Subtype :

Version : Windows 2022 (64-bit)

☐ Skip Unattended Installation

Detected OS type: Windows 2022 (64-bit). This OS type can be installed unattendedly. The install will start after this wizard is closed.

Aide Précédent Suivant Annuler

Crée une machine virtuelle

## Hardware

You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Mémoire vive : 2048 MB

Processors : 1

CPU 1 CPUs 4

☐ Enable EFI (special OSes only)

Aide Précédent Suivant Annuler

Crée une machine virtuelle

## Virtual Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select an existing one. Alternatively you can create a virtual machine without a virtual hard disk.

☒ Create a Virtual Hard Disk Now

Disk Size : 50,00 Gio

☐ Pre-allocate Full Size

☐ Use an Existing Virtual Hard Disk File

Vide

☐ Do Not Add a Virtual Hard Disk

Aide Précédent Suivant Annuler

Crée une machine virtuelle

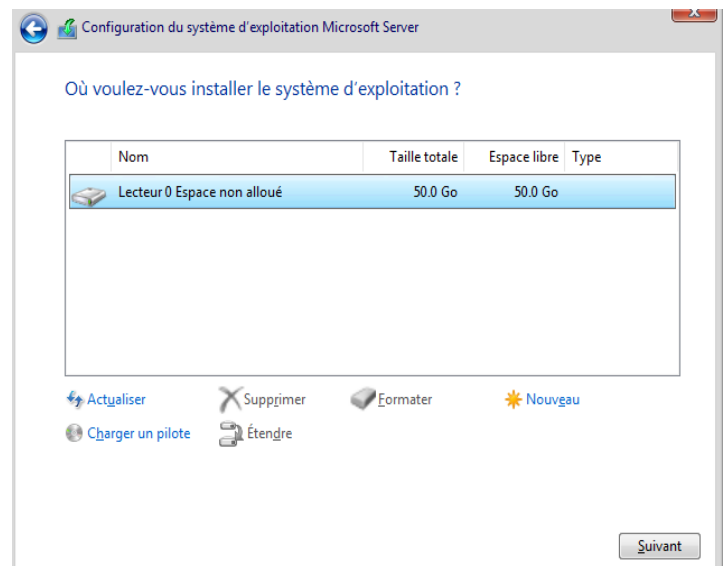
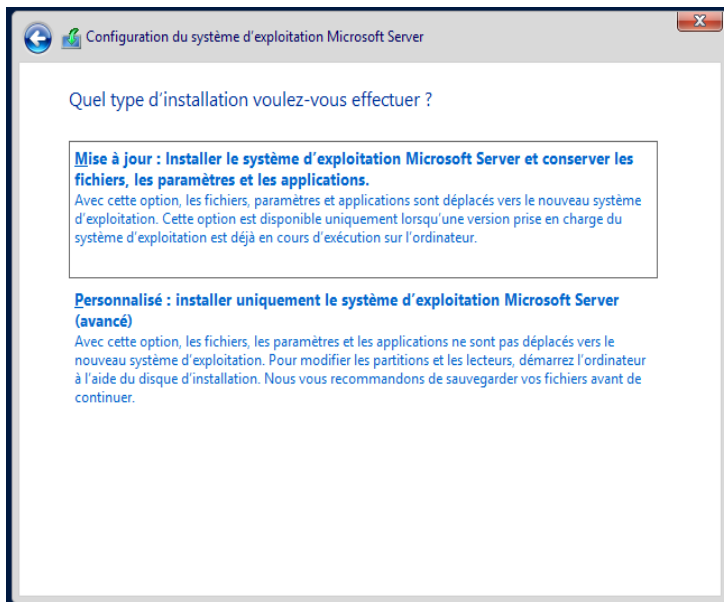
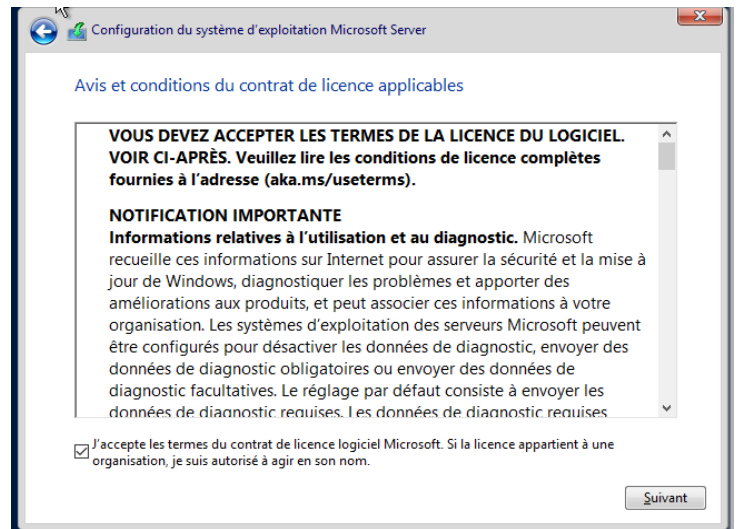
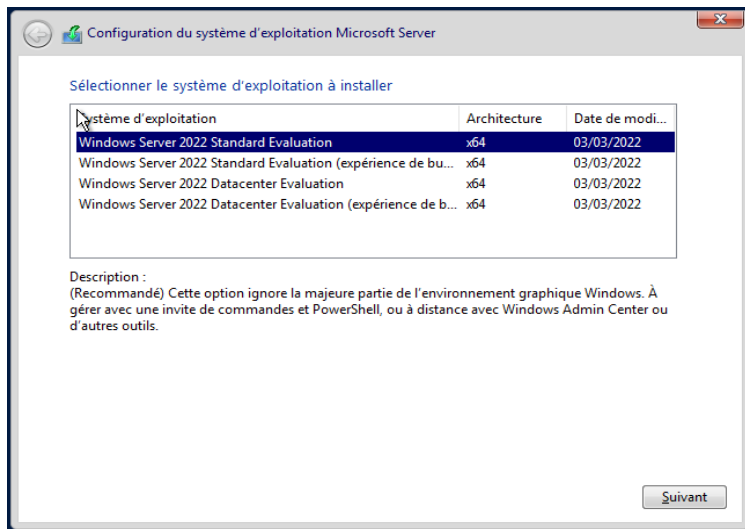
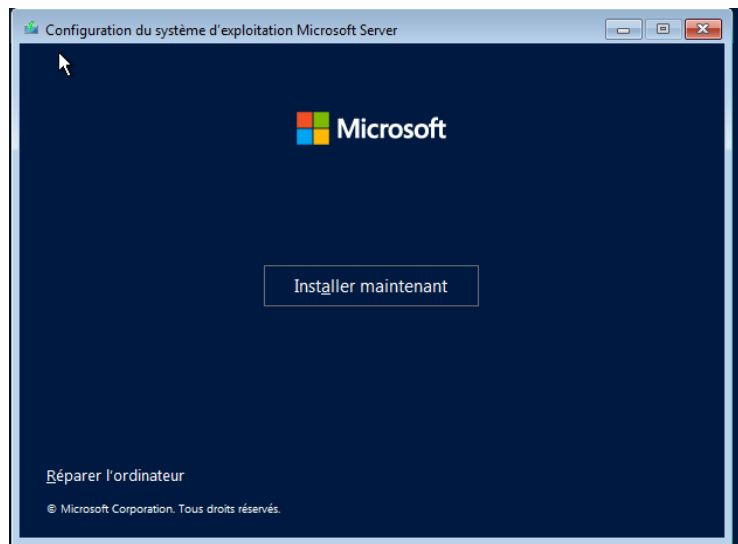
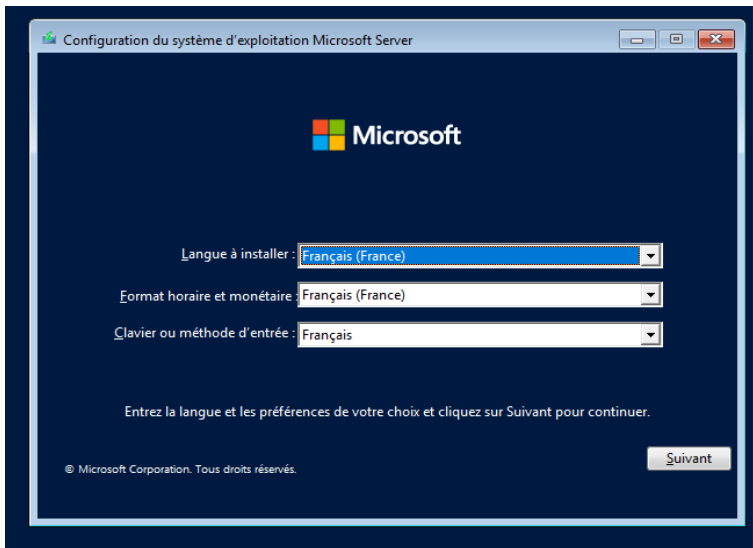
## Récapitulatif

The following table summarizes the configuration you have chosen for the new virtual machine. When you are happy with the configuration press Finish to create the virtual machine. Alternatively you can go back and modify the configuration.

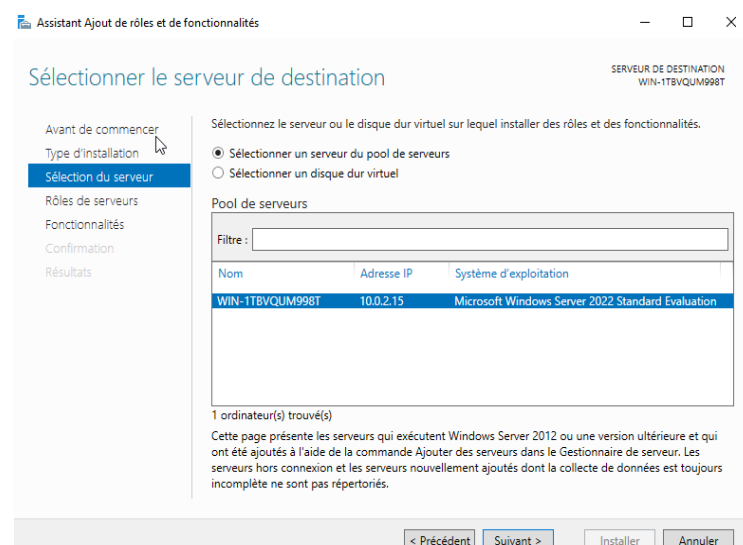
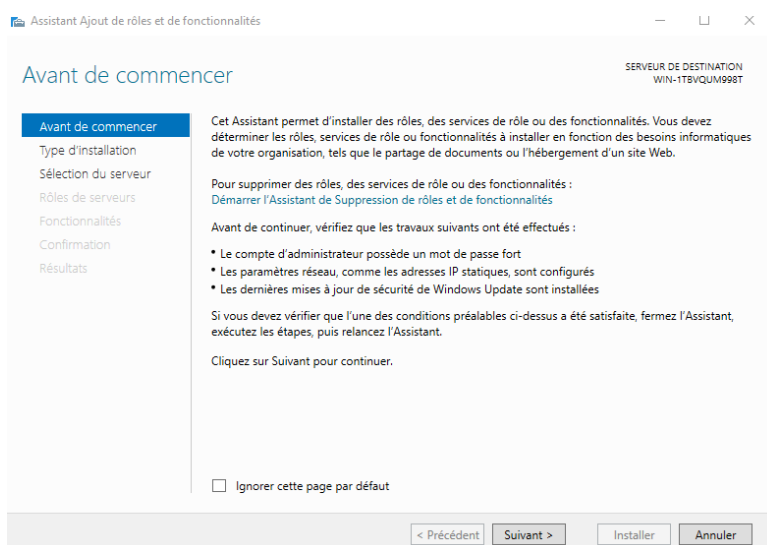
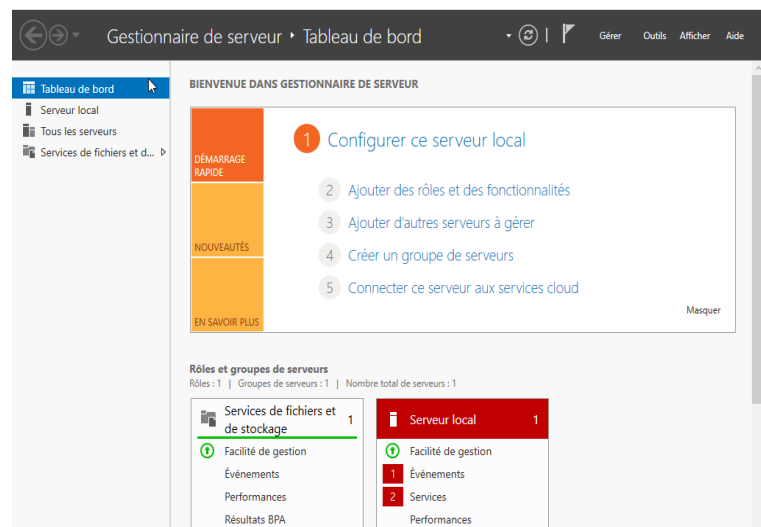
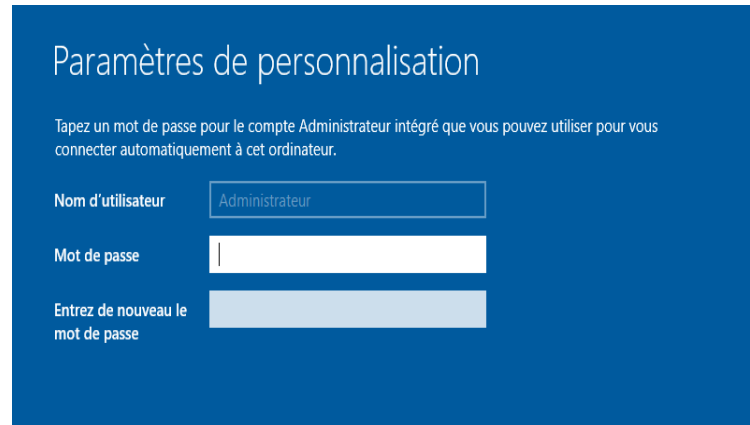
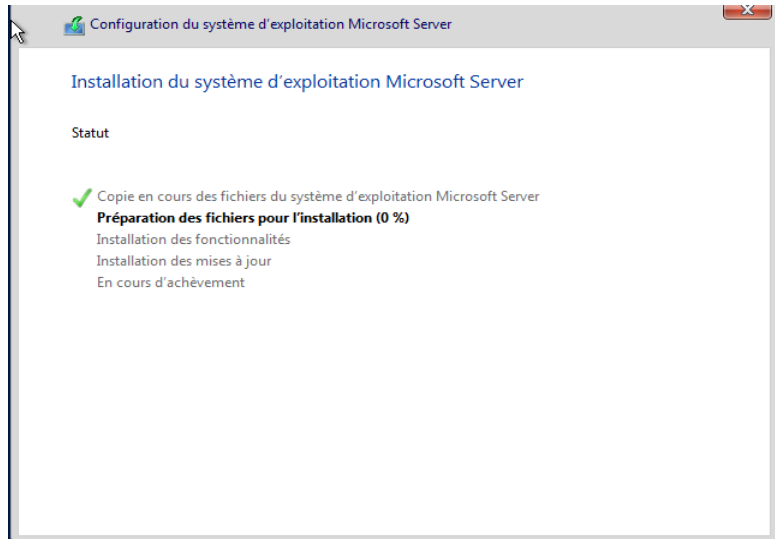
	<b>Machine Name and OS Type</b>
Machine Name	windows Server
Machine Folder	C:\Users\landr\VirtualBox VMs\windows Server
ISO Image	C:\Users\landr\OneDrive\Images\Documents\SERVER_EVAL_x64FRE_fr-fr.iso
Guest OS Type	Windows 2022 (64-bit)
Skip Unattended Install	true
	<b>Hardware</b>
Mémoire vive	2048
Processor(s)	1
EFI Enable	false
	<b>Disk</b>
Disk Size	50,00 Gio
Pre-allocate Full Size	false

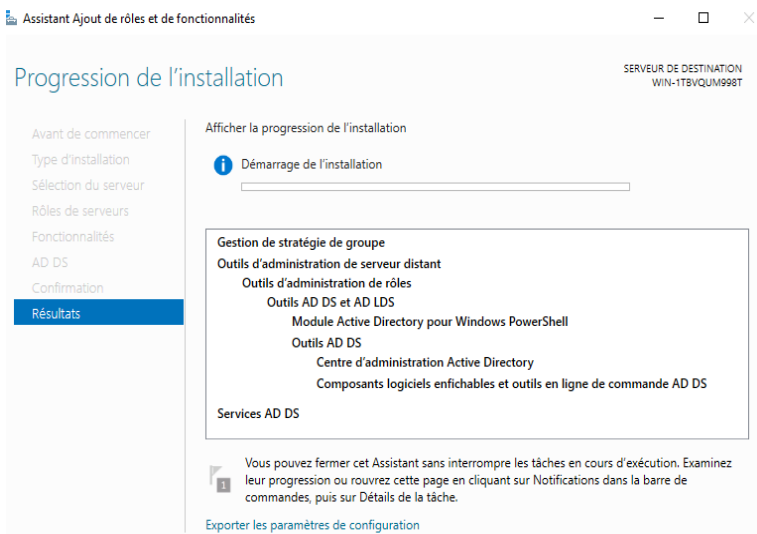
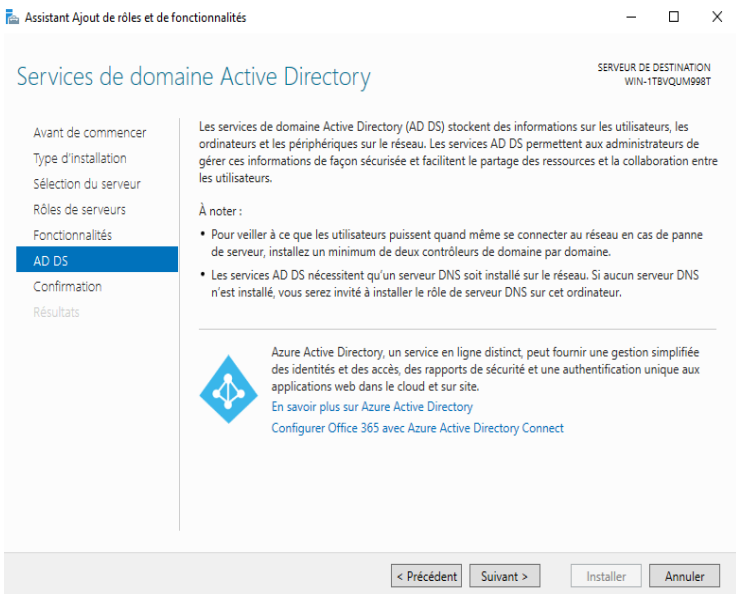
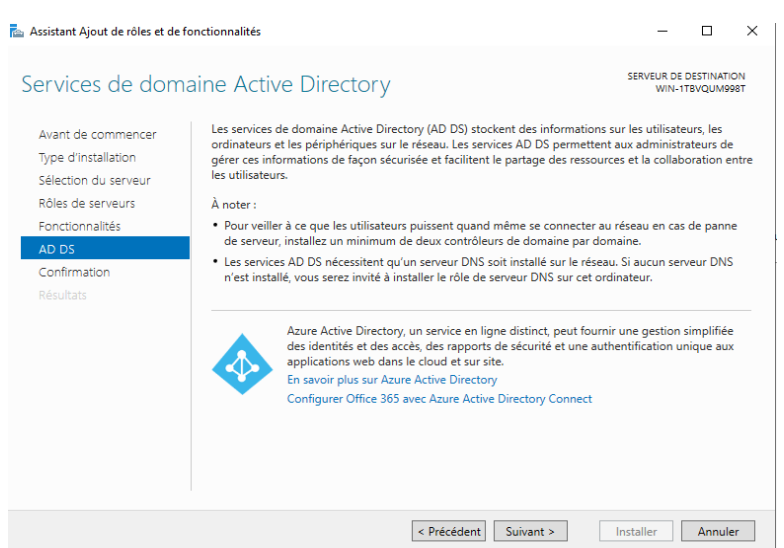
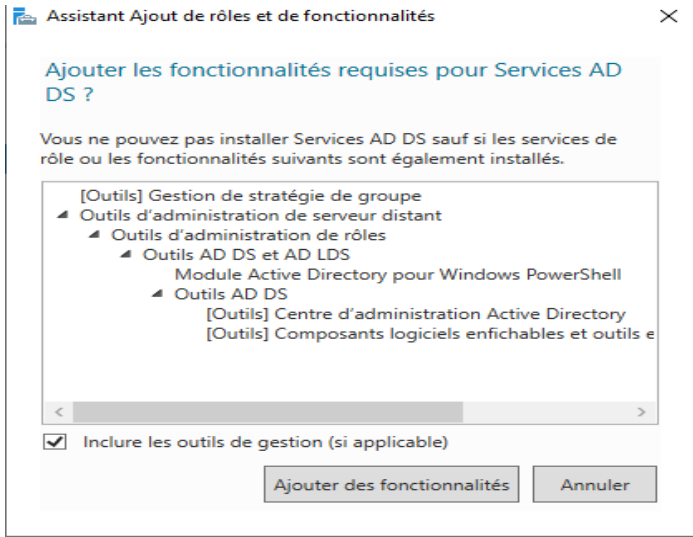
Aide Précédent Finish Annuler











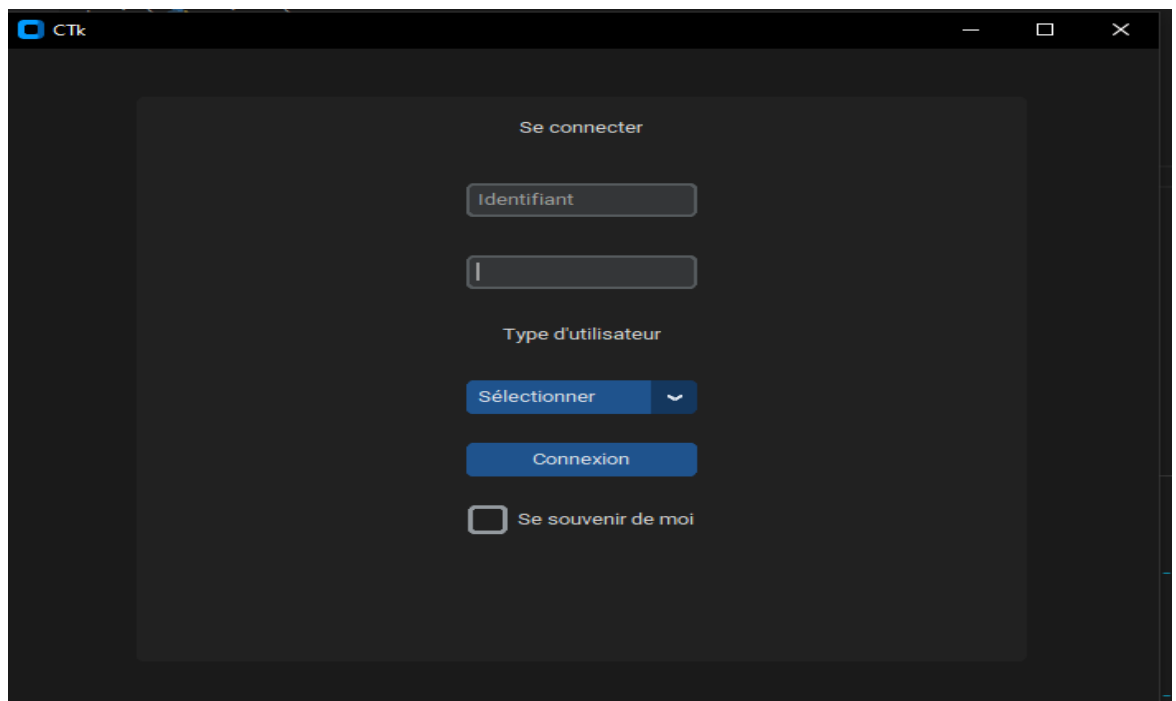
# TP : sur la conception d'une application de gestion des incidents

## Capture de quelques bouts de code

### Login

```
Login > main.py > ...
1  import subprocess
2  import customtkinter as ctk
3
4  ctk.set_appearance_mode("dark")
5  ctk.set_default_color_theme("dark-blue")
6
7  login = ctk.CTk()
8  login.geometry("700x550")
9
10 def connexion():
11     print("Bienvenue")
12
13 def Menu_Principal():
14     subprocess.run(["python", "Menu_Principal.py"])
15
16 frame = ctk.CTkFrame(master=login)
17 frame.pack(pady=40, padx=80, fill="both", expand=True)
18
19 label = ctk.CTkLabel(master=frame, text="Se connecter")
20 label.pack(pady=12, padx=10)
21
22 champ1 = ctk.CTkEntry(master=frame, placeholder_text="Identifiant")
23 champ1.pack(pady=20)
24
25 champ2 = ctk.CTkEntry(master=frame, placeholder_text="Mot de passe", show="*")
26 champ2.pack(pady=12)
27
28 # Ajout du champ de sélection pour le type d'utilisateur
29 label_user_type = ctk.CTkLabel(master=frame, text="Type d'utilisateur")
30 label_user_type.pack(pady=12, padx=10)
31
32 user_type = ctk.StringVar(value="Sélectionner")
33 user_type_menu = ctk.CTkOptionMenu(master=frame, variable=user_type, values=["Admin", "Standard", "Technicien"])
34 user_type_menu.pack(pady=12, padx=10)
35
36 button = ctk.CTkButton(master=frame, text="Connexion", command=Menu_Principal)
37 button.pack(pady=12, padx=10)
38
39 checkbox = ctk.CTkCheckBox(master=frame, text="Se souvenir de moi")
40 checkbox.pack(pady=12, padx=10)
41
42 login.mainloop()
43
```

### Résultat



## Menu principal

```
Menu_Principal.py > ...
1  from tkinter import *
2  import subprocess
3
4  def Incident():
5      subprocess.run(["python", "Incident.py"])
6
7  def Technicien():
8      subprocess.run(["python", "Technicien.py"])
9
10 def Commentaire():
11     subprocess.run(["Commentaire", ".py"])
12
13 def Cathegorie():
14     subprocess.run(["python", "Categorie.py"])
15
16
17 #titre general
18 root = Tk()
19 root.title("Gestion des incidents ")
20 root.geometry("600x600")
21
22 #Ajouter le titre
23 lbltitre = Label(root,bd = 20, relief = RIDGE, text = "GESTION DES INCIDENTS", font = ("Arial", 30), bg = "#9B59B6")
24 lbltitre.place(x = 0, y = 0, width = 600, height = 500)
25
26 Button(root,text="Incident",font=("Cambria",18),command=Incident, width=30, height=0).pack(pady=(100,20))
27 Button(root,text="Technicien",font=("Cambria",18),command=Technicien, width=30, height=0).pack(pady=(0,20))
28 Button(root,text="Commntaire",font=("Cambria",18),command=Commentaire, width=30, height=0).pack(pady=(0,20))
29 Button(root,text="Categorie",font=("Cambria",18),command=Cathegorie, width=30, height=0).pack(pady=(0,20))
30
31
32 root.mainloop()
```

## Résultat



## Signaler un incident bout de code

```
Signaler_Incident.py > lancer_impression > impression > draw_wrapped_text
1  from tkinter import ttk
2  from tkinter import *
3  from tkinter import messagebox
4  import mysql.connector
5  from datetime import datetime
6  from tkcalendar import DateEntry
7  from reportlab.lib.pagesizes import letter
8  from reportlab.pdfgen import canvas
9  from reportlab.lib import colors
10 import threading
11
12
13
14 import smtplib
15 from email.mime.multipart import MIMEMultipart
16 from email.mime.text import MIMEText
17 from tkinter import messagebox
18 import mysql.connector
19
20 def envoyer_email_incident(admin_email, titre, description, utilisateur, date_signalement):
21     # Configuration SMTP
22     smtp_server = "smtp.gmail.com"
23     smtp_port = 587
24     sender_email = "landryamengne@gmail.com" # Remplacez par votre adresse email
25     sender_password = "sasouker" # Remplacez par votre mot de passe ou mot de passe d'application
26
27     # Création du message email
28     message = MIMEMultipart()
29     message['From'] = sender_email
30     message['To'] = admin_email
31     message['Subject'] = "Nouveau Signalement d'Incident"
32
33     # Contenu du message
34     body = f"""
35     Bonjour Administrateur,
36
37     Un nouvel incident a été signalé par l'utilisateur : {utilisateur}
38
39     **Titre** : {titre}
40     **Description** : {description}
41     **Date de Signalement** : {date_signalement}
42
43     Merci de prendre en charge cet incident.
44
45     Cordialement,
46     Votre application de gestion des incidents.
47     """
48     message.attach(MIMEText(body, 'plain'))
49
50     try:
51         # Connexion au serveur SMTP
52         server = smtplib.SMTP(smtp_server, smtp_port)
53         server.starttls() # Active la sécurité TLS
54         server.login(sender_email, sender_password) # Authentification
55         # Envoi du message
56         server.send_message(message)
57         print("Email envoyé avec succès à l'administrateur.")
58         server.quit()
59     except Exception as e:
60         print(f"Erreur lors de l'envoi de l'email : {e}")
61
```

```

62 def ajouter():
63     try:
64         # Récupération des données depuis les champs
65         titre = entrertitre.get()
66         description = entrerdescription.get()
67         etat = combostatut.get()
68         priorite = entrerpriorite.get()
69         categorie = entrercategorie.get()
70         impact = entrerimpact.get()
71         service = entrerservice.get()
72         utilisateur = entrerutilisateur.get()
73         signalement = entrerde_date_signalement.get_date().strftime('%Y-%m-%d')
74
75         # Connexion à la base de données
76         con = mysql.connector.connect(
77             host="localhost",
78             user="root",
79             password="",
80             database="gestion_incidents"
81         )
82         cuser = con.cursor()
83         cuser.execute("INSERT INTO incident (titre, description, etat, priorite, categorie, impact, service, utilisateur, signalement)
84             VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)"
85             (titre, description, etat, priorite, categorie, impact, service, utilisateur, signalement))
86         con.commit()
87         con.close()
88
89         # Envoi d'un email à l'administrateur
90         admin_email = "landryamengne@gmail.com" # Remplacez par l'email de l'administrateur
91         envoyer_email_incident(admin_email, titre, description, utilisateur, signalement)
92
93         # Message de confirmation
94         messagebox.showinfo("Information", "Incident ajouté avec succès et email envoyé.")
95         afficher() # Mettre à jour le tableau
96     except ValueError:
97         messagebox.showerror("Erreur", "Le format de la date doit être JJ/MM/AAAA")
98     except Exception as e:
99         messagebox.showerror("Erreur", f"Une erreur est survenue : {e}")
100
101 # Assure-toi que les autres fonctions et la configuration de l'interface utilisateur sont définies ici
102
103 def modifier():
104     try:
105         codeSelectionner = table.item(table.selection())['values'][0]
106         titre = entrertitre.get()
107         description = entrerdescription.get()
108         etat = combostatut.get()
109         priorite = entrerpriorite.get()
110         categorie = entrercategorie.get()
111         impact = entrerimpact.get()
112         service = entrerservice.get()
113         utilisateur = entrerutilisateur.get()
114         signalement = entrerde_date_signalement.get_date().strftime('%Y-%m-%d') # Conversion de la date
115

```

## Résultat

Gestion des incidents

INCIDENT

Titre incident

Description Incident

Enregistrer

Statut

En attente

Priorité Incident

Modifier

Catégorie Incident

Impact Incident

Supprimer

Service Affecté

Utilisateur Affecté

Imprimer

Date de Signalement

06/02/2025

Afficher Historique

LISTES DES INCIDENTS

CODE	Titre	Description	Statut	Priorité	Catégorie	Impact	Service	Utilisateur
------	-------	-------------	--------	----------	-----------	--------	---------	-------------

Gestion des incidents

INCIDENT

Titre incident

problème de logiciel

Description Incident

ne fonctionne pas correctement

Enregistrer

Statut

En attente

Priorité Incident

Basse

Modifier

Catégorie Incident

Problème logiciel

Impact Incident

ecte pas le travail de l'utilisateur

Supprimer

Service Affecté

Utilisation du logiciel

Utilisateur Affecté

IKO CAMDEU PAUL EVA

Imprimer

Date de Signalement

26/12/2024

Afficher Historique

Information

Incident ajouté avec succès et email envoyé.

OK

CODE	Titre	Description	Statut	Priorité	Catégorie	Impact	Service	Utilisateur	Date
2	problème de connexion au réseau	l'utilisateur ne peut pas se connecter au réseau internet depuis s	En attente	élevée	Problème technique	Bloque le travail de l'utilisateur	Accès au réseau internet	Ms Dipoko Jonathan	2025-02-1
3	problème de virus	l'utilisateur ne parvient plus à accéder à certains logiciels	En attente	élevée	Problème technique	Bloque le travail de l'utilisateur	Accès à certains logiciels	EDONGO AMENGNE	2025-01-
4	problème d'impression	l'utilisateur ne peut pas se connecter au serveur	En attente	élevée	Problème technique	Bloque le travail de l'utilisateur	Imprimante	OMBIOMI LYDIE RAISSA	2025-01-1



Gestion des incidents

INCIDENT

Titre incident

problème de logiciel

Description Incident

ne fonctionne pas correctement

Statut

En attente

Priorité Incident

Basse

Catégorie Incident

Problème logiciel

Impact Incident

ecte pas le travail de l'utilisateur

Service Affecté

Utilisation du logiciel

Date de Signalement

26/12/2024

Enregistrer

Modifier

Supprimer

Imprimer

Afficher Historique

Confirmation

Voulez-vous imprimer le tableau ?

OuiNon

CODE	Titre	Description	Statut	Priorité	Catégorie	Impact	Service	Utilisateur	Date
2	problème de connexion au réseau	l'utilisateur ne peut pas se connecter au réseau internet depuis s	En attente	élevée	Problème technique	Bloque le travail de l'utilisateur	Accès au réseau internet	Ms Dipoko Jonathan	2025-02-1
3	problème de virus	l'utilisateur ne parvient plus a accéder à certains logiciel	En attente	élevée	Problème technique	Bloque le travail de l'utilisateur	Accès a certains logiciel	EDONGO AMENGNE	2025-01-1
4	problème d'impression	l'utilisateur ne peut pas se connecter au serveur	En attente	élevée	Problème technique	Bloque le travail de l'utilisateur	Imprimante	OMBIOMI LYDIE RAISSA	2025-01-1
5	problème de logiciel	le logiciel ne fonctionne pas correctement	En attente	Basse	Problème logiciel	n'affecte pas le travail de l'utilis	Utilisation du logiciel	KENKO CAMDEU PAUL EVA	2024-12-1

Aperçu avant impression

CODE	Titre	Description	Statut	Priorité	Catégorie	Impact	Service	Utilisateur	Date
2	problème de connexion au	l'utilisateur ne peut pas se connecter au réseau internet depi	En attente	élevée	Problème technique	Bloque le travail de l'utilisateur	Accès au réseau internet	Ms Dipoko Jonathan	2025-02-06
3	problème de virus	l'utilisateur ne parvient plus a accéder à certains logiciel	En attente	élevée	Problème technique	Bloque le travail de l'utilisateur	Accès a certains logiciel	EDONGO AMENGNE	2025-01-16
4	problème d'impression	l'utilisateur ne peut pas se connecter au serveur	En attente	élevée	Problème technique	Bloque le travail de l'utilisateur	Imprimante	OMBIOMI LYDIE RAISSA	2025-01-06
5	problème de logiciel	le logiciel ne fonctionne pas correctement	En attente	Basse	Problème logiciel	n'affecte pas le travail de l'utilisateur	Utilisation du logiciel	KENKO CAMDEU PAUL EVA	2024-12-26

Imprimer



# TP : MARCHINE LEARNING

## 1. Définition de fichier logs

C'est l'historicité des évènements dans un ordinateur

## 2. Caractéristiques

Format, taille, Encodage, structure, chronologie

## 3. Avantages et inconvénients

### a) Avantages

- Traquer les utilisateurs
- Surveillance réseaux
- Sécurité du système

### b) Inconvénients

- Le temps de traitement peut être élevé
- Le fichier peut occuper rapidement beaucoup d'espace
- Trop de détails inutiles peut réduire l'efficacité

## 4. Syntaxe du Fichier logs dans PowerShell

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\landr> Get-Process | Export-Csv -Path "C:\Users\landr\Documents\logs.csv" -NoTypeInformation
```

	A	B	C	D	E	F	G	H	I	J	K	L
1	Name	SI	Handles	VM	WS	PM	NPM	Path	Company	CPU	FileVersion	ProductVersion
2	AdobeIPCBroker	1	218	76984320	10567680	2772992	13368	Adobe\Adobe Desktop	Adobe Inc	0.09375	6.7.0.44	6.7.0.44
3	AggregatorHost	0	136	2,20E+12	6864896	2453504	7128					
4	ai	1	144	2,20E+12	13197312	11296768	10936	\ProgramFilesComrosoft Corpora		0.140625	0.12.3.0	0.12.3.0
5	CCXProcess	1	50	4343926784	2863104	634880	3936	obe Creative Cloud Systems Incor		0.015625	4.3.2.1	4.3.2.1
6	cmd	1	86	2,20E+12	4190208	3313664	5704	indows\SYSTEM32\crossoft Corpora		0.0625	22 (WinBuild 10.0.19041.4522	
7	cmd	1	80	2,20E+12	3829760	2342912	5024	indows\system32\cnosoft Corpora		0.015625	22 (WinBuild 10.0.19041.4522	
8	Code	1	299	2,24E+12	44974080	16814080	17064	Code\Microsoft VS osoft Corpora		0.234375	1.94.2	1.94.2
9	Code	1	1013	3,42E+12	116174848	96083968	49496	Code\Microsoft VS osoft Corpora		19.171875	1.94.2	1.94.2
10	Code	1	189	3,42E+12	92041216	30171136	16256	Code\Microsoft VS osoft Corpora		0.328125	1.94.2	1.94.2
11	Code	1	343	3,42E+12	215912448	195170304	28040	Code\Microsoft VS osoft Corpora		9.734375	1.94.2	1.94.2
12	Code	1	282	3,42E+12	116539392	100093952	23496	Code\Microsoft VS osoft Corpora		6.96875	1.94.2	1.94.2
13	Code	1	518	2,24E+12	111808512	79048704	21592	Code\Microsoft VS osoft Corpora		110.359375	1.94.2	1.94.2
14	Code	1	241	2,24E+12	32473088	14876672	11872	Code\Microsoft VS osoft Corpora		0.03125	1.94.2	1.94.2
15	Code	1	212	3,42E+12	490532864	479133696	22552	Code\Microsoft VS osoft Corpora		33.125	1.94.2	1.94.2
16	Code	1	522	3,43E+12	231067648	217030656	36408	Code\Microsoft VS osoft Corpora		214.71875	1.94.2	1.94.2
17	Code	1	282	3,43E+12	113954816	101818368	28896	Code\Microsoft VS osoft Corpora		2.015625	1.94.2	1.94.2
18	Code	1	187	3,42E+12	101478400	37650432	16256	Code\Microsoft VS osoft Corpora		0.625	1.94.2	1.94.2
19	Code	1	241	3,42E+12	92553216	78651392	22840	Code\Microsoft VS osoft Corpora		2.546875	1.94.2	1.94.2
20	conhost	1	130	2,20E+12	7704576	6885376	9776	ows\system32\conlosoft Corpora		0.09375	22 (WinBuild 10.0.19041.4522	
21	conhost	1	274	2,20E+12	17158144	4612096	14400	ows\svstem32\conlosoft Corpora		0.296875	22 (WinBuild 10.0.19041.4522	

## 5. Analyse et Visualisation des graphiques

### a) Analyse

Nous sommes dans le cas d'un apprentissage non supervisé nous avons identifié des patterns anormaux sans données étiquetées, en se basant sur les caractéristiques du trafic. Il est particulièrement utile pour détecter de nouvelles attaques inconnues.

### **Nuage de points :**

- **Visualisation** : Le nuage de points montre la distribution des données selon deux caractéristiques principales (Feature1 et Feature2).

### **Matrice de confusion :**

- **Visualisation** : La matrice de confusion présente les vrais positifs, vrais négatifs, faux positifs, et faux négatifs.

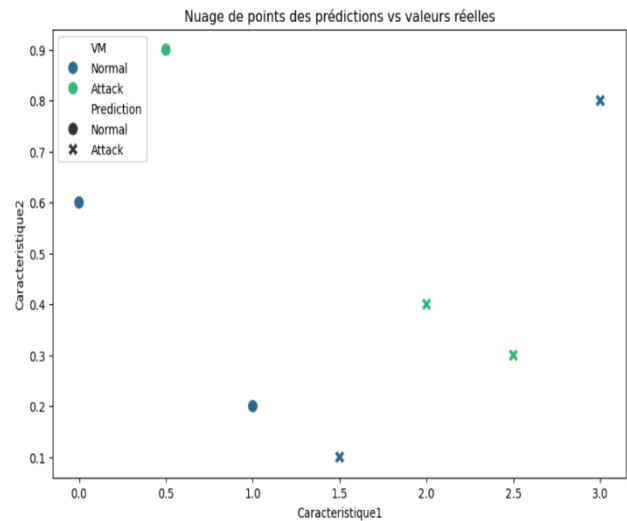
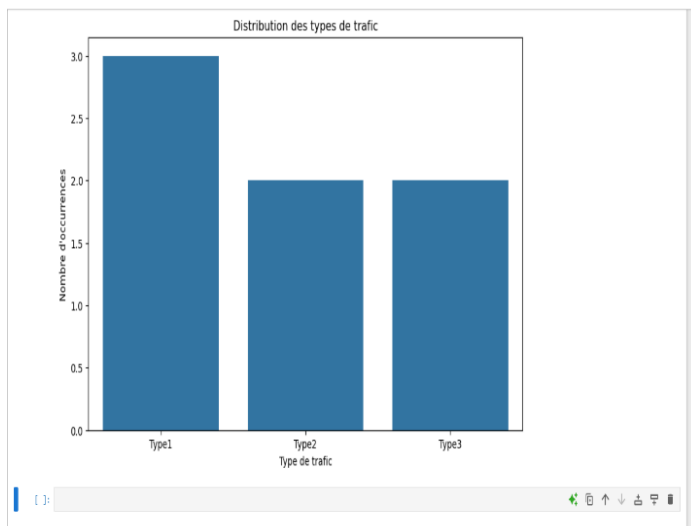
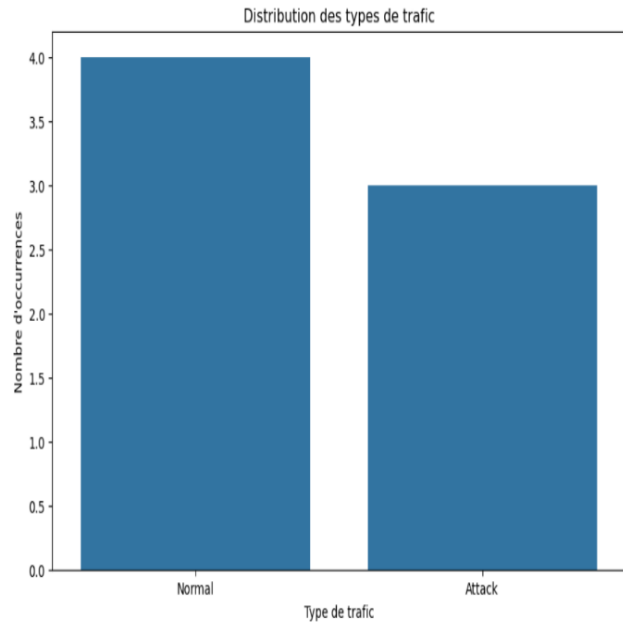
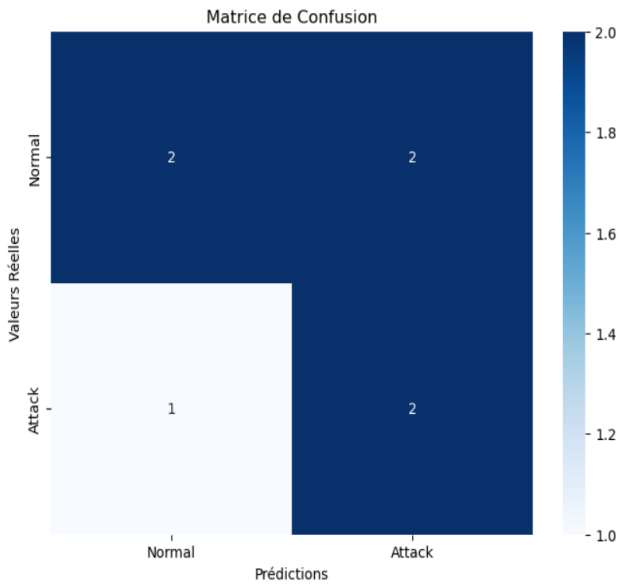
---

### **Métriques de classification :**

- **Précision** : Proportion des prédictions correctes.
  - Une haute précision globale indique un modèle fiable, mais elle doit être complétée par des métriques spécifiques par classe.
- **Rappel** : Capacité du modèle à identifier tous les exemples d'une classe.
  - Faible rappel pour Attack signifie que des attaques passent inaperçues.
- **F1-Score** : Moyenne harmonique entre précision et rappel.
  - Idéal pour évaluer les performances globales, en particulier si les classes sont déséquilibrées.

### **Visualisation**

Rapport de Classification :				
	precision	recall	f1-score	support
Normal	0.50	0.67	0.57	3
Attack	0.67	0.50	0.57	4
accuracy			0.57	7
macro avg	0.58	0.58	0.57	7
weighted avg	0.60	0.57	0.57	7



## 6. Interprétation

### Nuage de points :

#### ➤ Interprétation :

- Les couleurs distinguent les classes réelles (Normal et Attack).
- Les styles (formes des points) représentent les prédictions faites par le modèle.
- Si les points de différentes classes se chevauchent, cela peut indiquer une difficulté du modèle à bien séparer les classes, nécessitant peut-être une ingénierie des caractéristiques ou un modèle plus complexe.

➤ **Matrice de confusion :**

- **Vrais Positifs (TP)** : Nombre de cas Normal correctement prédits comme Normal.
- **Vrais Négatifs (TN)** : Nombre de cas Attack correctement prédits comme Attack.
- **Faux Positifs (FP)** : Cas Normal mal prédits comme Attack (erreurs coûteuses si "Normal" est important).
- **Faux Négatifs (FN)** : Cas Attack mal prédits comme Normal (erreurs critiques si "Attack" représente une menace).

Une diagonale dominante (valeurs élevées dans les TP et TN) indique de bonnes performances. Si les FP ou FN sont élevés, une révision du modèle ou des données pourrait être nécessaire.

## **Conclusion**

**1. Performance globale :**

- Si la précision est élevée et les FP/FN faibles, le modèle est performant.
- Si les FN sont élevés pour Attack, il y a un risque opérationnel élevé (non-détection des menaces).

**2. Recommandations :**

- Si les métriques sont faibles pour une classe (ex. Attack), il est essentiel d'explorer :
  - Des transformations ou sélections des caractéristiques.
  - Un échantillonnage pour équilibrer les classes.
  - Un modèle plus sophistiqué ou des hyperparamètres ajustés.

**3. Interprétation globale :**

- Les visualisations (nuage de points, matrice) mettent en évidence les zones où le modèle réussit ou échoue, offrant des indications claires pour affiner le pipeline d'analyse.