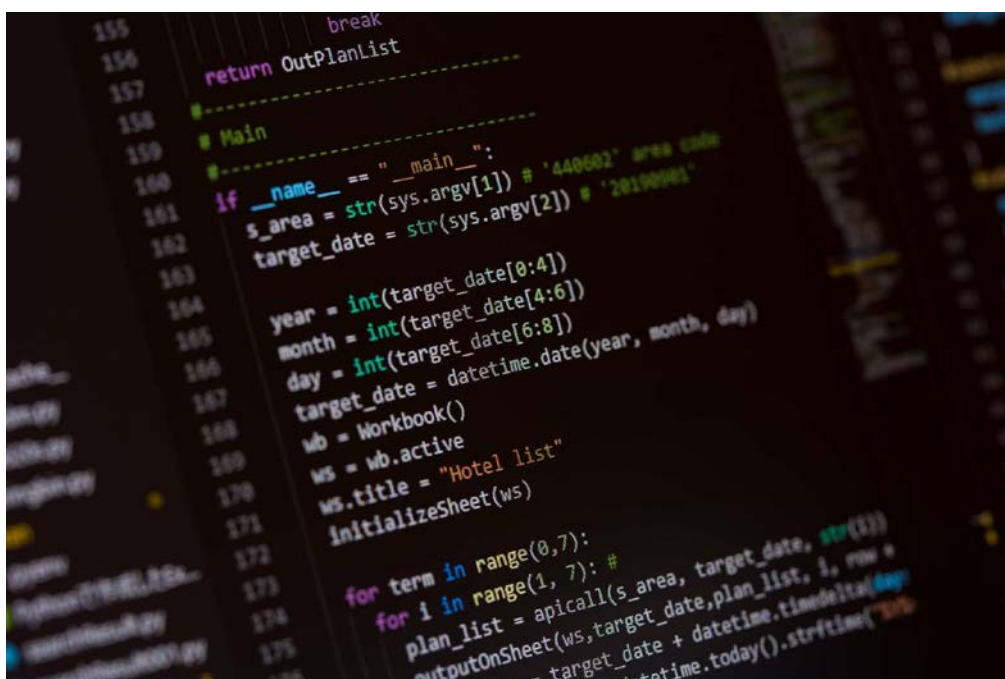


Rapport de projet



Participer à la vie de la communauté Open Source



Source : Google image

Sommaire

Présentation du projet	p.3
Mise en situation	p.3
Objectifs	p.3
Algorithme	p.4
Thème/Sujet du script/choix	p.4
Algorithme 1	p.5
Algorithme 2	p.6
Algorithme 3	p.7
PowerShell	p.8
Python	p.10
Téléchargement et installation de Python	p.10
Téléchargement et installation de PyCharm	p.11
Scripts	p.12
Tests du script	p.15
GitHub	p.16
Création du répertoire GitHub	p.16
Répertoire GitHub	p.18
Conclusion	p.19
Annexe	p.20
Lexique	p.21

Présentation du projet

Mise en situation :

Le web repose en grande partie sur un gigantesque réseau d'entraide où chacun peut à la fois recevoir mais aussi contribuer à sa mesure.

Aujourd'hui, une des plateformes les plus populaires pour contribuer à la communauté Open Source se nomme GitHub.

GitHub propose toute l'infrastructure permettant de facilement mettre à disposition du code, de la documentation, de rechercher des projets et de pouvoir y contribuer.

Vous allez maintenant créer un script sous Python permettant d'automatiser des tâches d'administration et de partager votre code avec la communauté sur votre répertoire personnel GitHub.

Objectifs :

- Identifier une tâche complexe ou un ensemble de tâches d'administration que vous aimeriez pouvoir automatiser.
- Une fois votre projet défini, créez un répertoire GitHub qui hébergera votre projet
- Il est maintenant temps de coder en Python votre outil d'automatisation, soit sous forme d'un script, soit sous forme d'un module Ansible
- Mettez votre code à disposition de la communauté
- Documentez à la fois le fonctionnement de votre code et la manière d'utiliser votre code sans entrer trop dans les détails de sa conception technique

Algorithme

Thème :

Script d'automatisation pour configurer un Windows Serveur.

Sujet du script :

Configuration IP, Nom et installation des services DHCP et AD.

Choix :

J'ai choisi de partir dans cette direction car dans mon entreprise nous travaillons sur des serveurs Windows et je me suis dit que ça pouvait être utile de travailler sur des scripts d'automatisation pour configurer des Serveurs Windows.

Nous allons avoir un algorithme principal qui va regrouper ces trois fonctions suivantes :

- 1- ConfigurerIPNom()
- 2- InstallationDHCP()
- 3- InstallationAD()

Le programme est basé sur du Windows 2019 avec un paramétrage basique.

Algorithmes 1 : ConfigurerIPNom()

Rôle : Configurer une IP fixe (Masque, DNS, Passerelle) ainsi que de nommer le serveur.

Début

Allez dans 'Centre réseau partage'

Allez dans 'Modifier paramètre de la carte'

Allez dans 'Propriété de la carte'

Allez dans 'Protocol IPV4'

Cliquer 'Utiliser l'adresse IP suivante'

Entrer : Adresse IP

Entrer : Masque de sous-réseau

Entrer : Passerelle par défaut

Entrer : Serveur DNS

Cliquer 'Ok'

Fermer fenêtre

Fermer fenêtre

Allez dans 'Paramètre'

Allez dans 'Système'

Allez dans 'A propos de'

Cliquer 'Renommer ce PC'

Entrer : Nom du serveur

Cliquer 'Suivant'

Faire Redémarrer

Fin

Algorithmes 2 : InstallationDHCP()

Rôle : Installation du service DHCP sans configuration.

(Une fois sur le Gestionnaire de serveur sur Windows Serveur 2019)

Début

Allez dans 'Gérer'

Allez dans 'Ajouter Rôle et fonctionnalité'

Cliquer 'Suivant'

Cliquer 'Suivant'

Cliquer 'Suivant'

Sélectionner 'Serveur DHCP'

Cliquer 'Ajouter'

Cliquer 'Suivant'

Cliquer 'Suivant'

Cliquer 'Suivant'

Cliquer 'Installer'

Cliquer 'Fermer'

Fin

Algorithmes 3 : InstallationAD()

Rôle : Installation du service Active Directory avec une configuration simple.

(Une fois sur le Gestionnaire de serveur sur Windows Serveur 2019)

Début

Allez dans 'Gérer'

Allez dans 'Ajouter Rôle et fonctionnalité'

Cliquer 'Suivant'

Cliquer 'Suivant'

Cliquer 'Suivant'

Sélectionner 'Serveur DNS'

Sélectionner 'Service AD DS'

Cliquer 'Ajouter les fonctionnalités'

Cliquer 'Suivant'

Cliquer 'Suivant'

Cliquer 'Suivant'

Cliquer 'Installer'

Cliquer 'Fermer'

Fin

PowerShell

Aide 1 : <https://www.it-connect.fr/windows-server-2012-r2-installer-dhcp-via-powershell/>

Aide 2 : <https://www.it-connect.fr/active-directory-deployer-adds-avec-powershell/>

Nous allons créer des script PowerShell pour pouvoir configurer notre serveur Windows.
Nous allons ensuite appeler ces scripts sur notre programme python.

Script IP et Nom :

```
# Configuration IP

#Debut du script :

echo `n "Bienvenue dans la configuration de l'adressage IP et du Nom du Serveur Windows."
echo `n "Voici les informations de votre carte réseau : "
Get-NetIPConfiguration

$ipAddress = Read-Host -Prompt "Entrez l'adresse IP (IPv4) : "
$ethernet = Read-Host -Prompt "Entrez l'index de l'interface"
$gateway = Read-Host -Prompt "Entrez le Gateway : "
$masqueSousReseau = Read-Host -Prompt "Entrez le masque de sous-réseau en notation CIDR (24) : "
$serverAddress = Read-Host -Prompt "Entrez l'adresse DNS : "

# Vérifier le numéro d'index avant

New-NetIPAddress -IPAddress $ipAddress -InterfaceIndex $ethernet -DefaultGateway $gateway -AddressFamily IPv4 -PrefixLength $masqueSousReseau
Set-DnsClientServerAddress -InterfaceIndex $ethernet -ServerAddresses $serverAddress

# Configuration du Nom du Serveur

#Debut du script :

$nom = Read-Host -Prompt "Entrez le nom du serveur Windows : "
Rename-Computer $nom
#Restart-Computer
```

Script DHCP :

```
# Installation du service DHCP
# Source du code : IT-CONNECT.FR

echo "Installation du service DHCP"
Install-WindowsFeature -Name DHCP -IncludeManagementTools

# Création d'un groupe de sécurité DHCP

Add-DhcpServerSecurityGroup
Restart-Service dhcpserver
```


Script AD :

```
# Installation du service AD
# Source du code : IT-CONNECT.FR

# Installation des rôles : AD, DNS, RSAT-AD-Tools

$FeatureList = @("RSAT-AD-Tools","AD-Domain-Services","DNS")

Foreach($Feature in $FeatureList)
{
    Install-WindowsFeature $Feature -IncludeManagementTools -IncludeAllSubFeature
}

# Création du domaine Active Directory

$DomainNameDNS = Read-Host -Prompt "Entrez le nom de domaine : "
$DomainNameNetbios = Read-Host -Prompt "Entrez le nom Netbios du domaine : "

$ForestConfiguration = @{
    '-DatabasePath' = 'C:\Windows\NTDS';
    '-DomainMode' = 'Default';
    '-DomainName' = $DomainNameDNS;
    '-DomainNetbiosName' = $DomainNameNetbios;
    '-ForestMode' = 'Default';
    '-InstallDns' = $true;
    '-LogPath' = 'C:\Windows\NTDS';
    '-NoRebootOnCompletion' = $false;
    '-SysvolPath' = 'C:\Windows\SYSVOL';
    '-Force' = $true;
    '-CreateDnsDelegation' = $false }

Import-Module ADDSDeployment
Install-ADDSForest @ForestConfiguration
```

Toutes les informations détaillées se trouvent sur mon répertoire GitHub dans un fichier texte qui accompagne les Scripts.

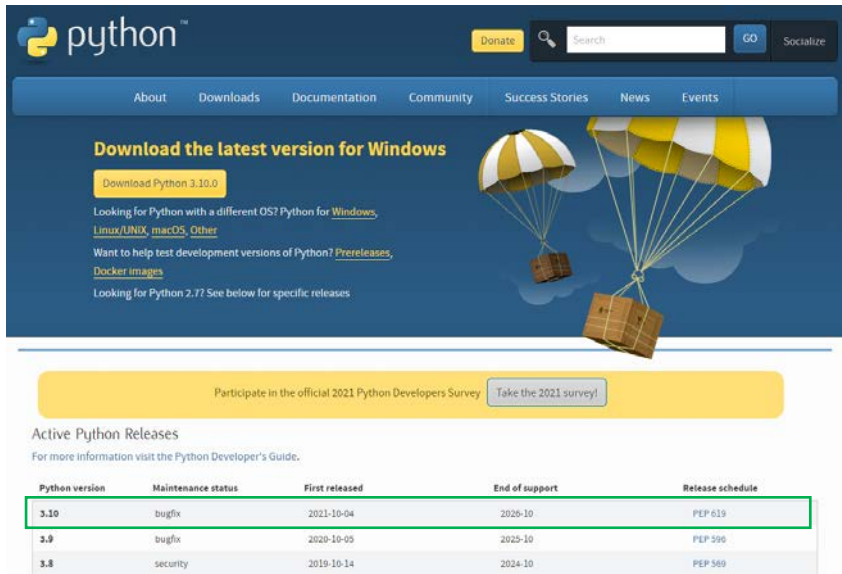
Python

Aide 1 : <https://stackoverflow.com/questions/21944895/running-powershell-script-within-python-script-how-to-make-python-print-the-pow>

Nous allons télécharger et installer python et de quoi développer sur python.

Téléchargement et installation de python :

Il faut télécharger la dernière version de python et suivre son installation.



The screenshot shows the Python.org homepage. The main heading is "Download the latest version for Windows" with a button "Download Python 3.10.0". Below this, there are links for "Linux/UNIX, macOS, Other", "Want to help test development versions of Python? Prereleases, Docker images", and "Looking for Python 2.7? See below for specific releases". There is also a banner for the "2021 Python Developers Survey" with a "Take the 2021 survey!" button. At the bottom, there is a table titled "Active Python Releases" with columns: Python version, Maintenance status, First released, End of support, and Release schedule.

Python version	Maintenance status	First released	End of support	Release schedule
3.10	bugfix	2021-10-04	2026-10	PEP 619
3.9	bugfix	2020-10-05	2025-10	PEP 596
3.8	security	2019-10-14	2024-10	PEP 569

Maintenant pour vérifier si Python est bien installé sur l'ordinateur on peut taper la commande ci-dessous.

```
C:\> Invite de commandes
Microsoft Windows [version 10.0.19043.1288]
(c) Microsoft Corporation. Tous droits réservés.

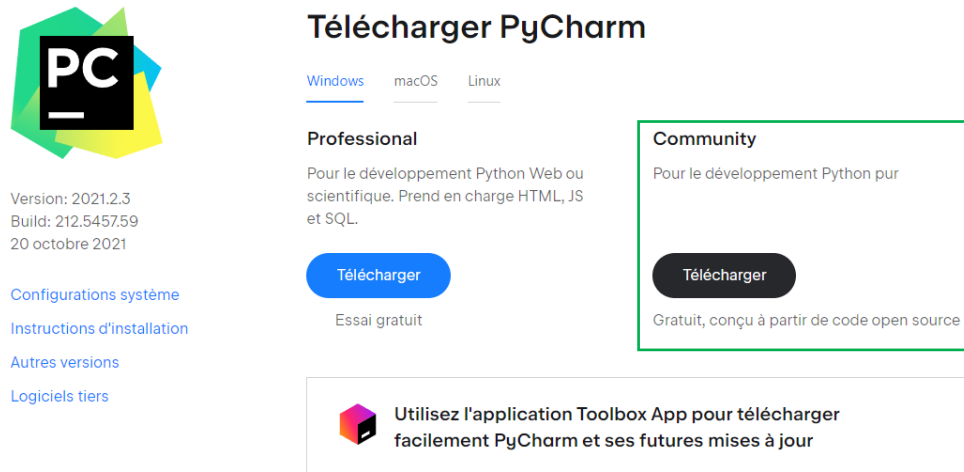
C:\Users\Dotrakii>python --version
Python 3.10.0

C:\Users\Dotrakii>
```

Téléchargement et installation de PyCharm :

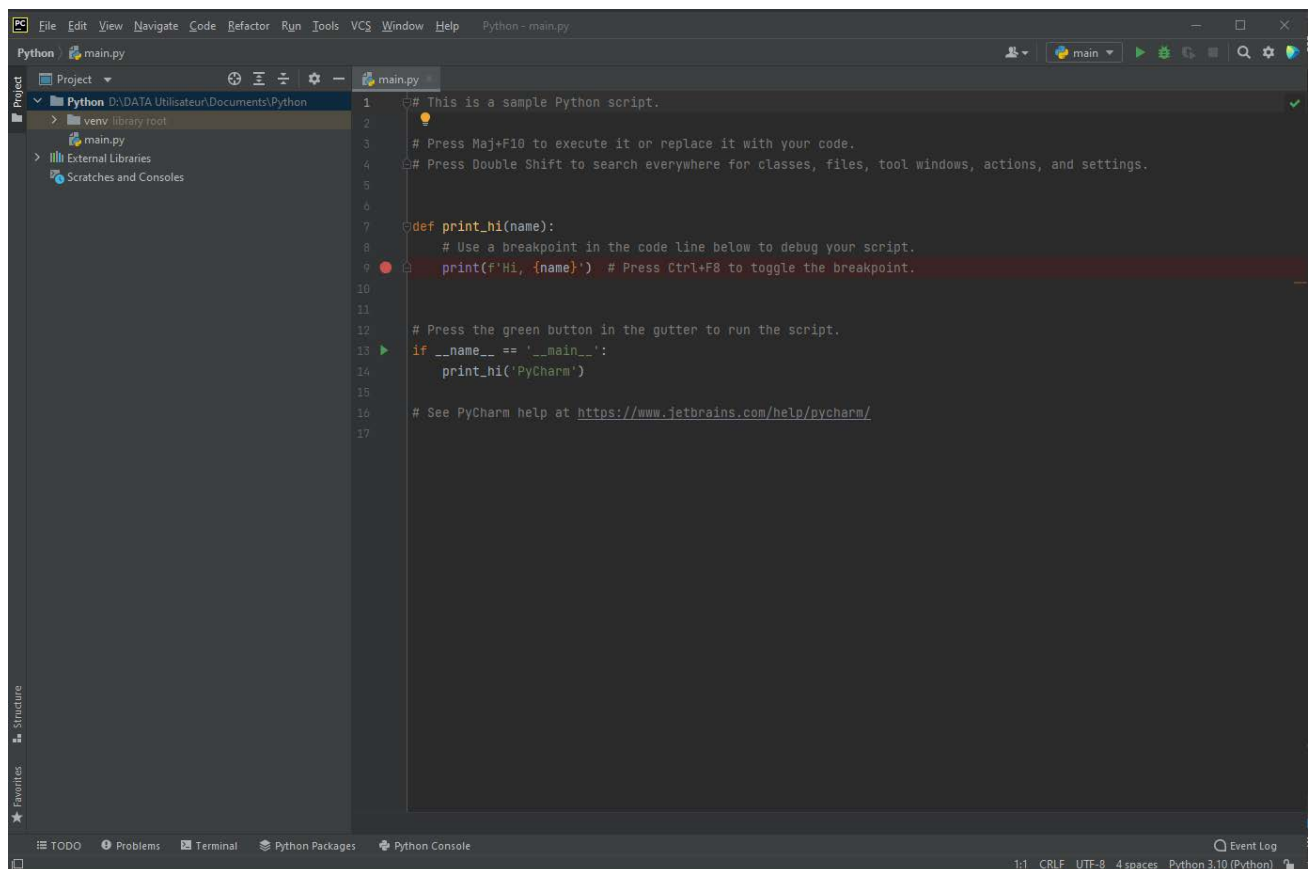
PyCharm est un logiciel qui permet de développer en langage python. J'ai choisi celui-là suite à une vidéo qui proposait deux choix et cette interface m'a plu.

Nous allons procéder comme l'installation de python, télécharger la dernière version et suivre l'installation de PyCharm.



The image shows the PyCharm download page. On the left is the PyCharm logo (a green and yellow hexagon with 'PC' in a black square) and version information: Version: 2021.2.3, Build: 212.5457.59, 20 octobre 2021. Below this are links for 'Configurations système', 'Instructions d'installation', 'Autres versions', and 'Logiciels tiers'. The main heading is 'Télécharger PyCharm' with tabs for 'Windows', 'macOS', and 'Linux'. Under 'Professional', it says 'Pour le développement Python Web ou scientifique. Prend en charge HTML, JS et SQL.' and has a blue 'Télécharger' button and 'Essai gratuit' text. Under 'Community', it says 'Pour le développement Python pur' and has a dark blue 'Télécharger' button and 'Gratuit, conçu à partir de code open source' text. At the bottom, a box says 'Utilisez l'application Toolbox App pour télécharger facilement PyCharm et ses futures mises à jour' with a Toolbox App icon.

Et voici le résultat quand on ouvre PyCharm :



Scripts :

Voici la capture d'écran du programme avec les trois fonctions ainsi que le programme principal.

```
# -----#  
#      Formation Administrateur Infrastructure et Cloud OpenClassRooms      #  
#      Projet 6 : Participer à la vie de la communauté Open Source      #  
#      CECCHINATO Quentin      #  
# -----#  
  
# -----#  
#      Programme d'installation et de configuration de Windows Serveur 2019      #  
  
# Rôle : Ce programme sera composé de 3 fonctions qui vont installer et      #  
#      configurer, avec un paramétrage basique, un Windows serveur2019.      #  
# -----#  
  
import ...  
  
#      Les Fonctions      #  
  
# -----#  
#      Fonction 1 : ConfigurerIPNom()      #  
  
# Rôle : Avec cette fonction on va pouvoir configurer l'adresse IP, le masque      #  
#      de sous-réseau, le DNS, la passerelle par défaut et le nom du Windows      #  
#      Serveur 2019.      #  
  
#      ConfigurerIPNom()      #  
  
def configureripnom():  
    print("Bienvenue dans la configuration IP et Nom du serveur Windows.")  
    # appel du script powershell IPNom  
  
    p = subprocess.Popen(["powershell.exe",  
                           "D:\\PowerShell\\IPNom.ps1"],  
                           stdout=sys.stdout)  
    p.communicate()  
    print("Vous avez terminé la configuration IP et Nom du serveur Windows.")  
    pass  
  
#      Fin      #  
# -----#
```

```

# ----- #
#                               Fonction 2 : InstallationDHCP()                               #
#
# Rôle : Cette fonction va installer le service DHCP sans le configurer. #
#
#                               InstallationDHCP()                               #
def installationdhcp():
    print("Bienvenue dans l'installation du service DHCP.")
    # appel du script powershell DHCP

    p = subprocess.Popen(["powershell.exe",
                           "D:\\PowerShell\\DHCP.ps1"],
                           stdout=sys.stdout)

    p.communicate()
    print("Vous avez terminé l'installation du service DHCP.")
    pass

#                               Fin                               #
# ----- #

```

```

# ----- #
#                               Fonction 3 : InstallationAD()                               #
#
# Rôle : Cette fonction va installer le service AD et le configurer (simplement) #
#
#                               InstallationAD()                               #
def installationad():
    print("Bienvenue dans l'installation du service AD.")
    # appel du script powershell AD

    p = subprocess.Popen(["powershell.exe",
                           "D:\\PowerShell\\AD.ps1"],
                           stdout=sys.stdout)

    p.communicate()
    print("Vous avez terminé l'installation du service AD.")
    pass

#                               Fin                               #
# ----- #

```

```

# ----- #
#   Programme d'installation et de configuration de Windows Serveur 2019   #
#
# Rôle : Ce programme sera composé de 4 fonctions qui vont installer et    #
#   configurer, avec un paramétrage basique, un Windows serveur2019.      #
#
#                               Programme : Installer&ConfigurerWinSV2019    #
#
Oui = "o"
Non = "n"

reponse = input("Bonjour, voulez vous configurer Windows Serveur 2019 ? (Oui : o Non : n) : ")
if reponse == Oui:
    print("Nous allons commencer l'installation.")

    print("Première étape : Configuration de l'adressage IP et du Nom du Serveur Windows")
    configureripnom()

    print("Deuxième étape : Installation du service DHCP")
    installationdhcp()

    print("troisième étape : Installation du service AD")
    installationad()

    print("Vous avez terminer la configuration de votre serveur Windows.")
    print("Le serveur va redemarrer.")

# Au début j'avais noté ça : elif reponse == Non :
else:
    print("La configuration a été annulé.")
pass

#                               Fin                               #
# ----- #

```

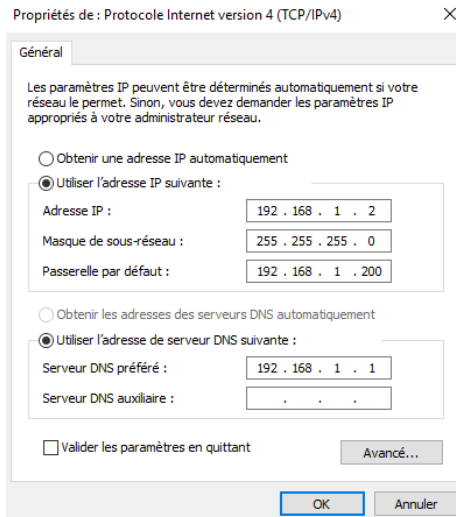
Notes importantes :

- faire attention aux chemins d'accès qu'il ne soit pas trop long
- faire attention à la réutilisation des variables
- faire attention à la structure du code (placement des fonctions, la présentation, les lignes de codes)
- faire attention aux chemins avec des espaces

Tests du script

Voici les résultats des tests après le lancement du programme.

IP et Nom :

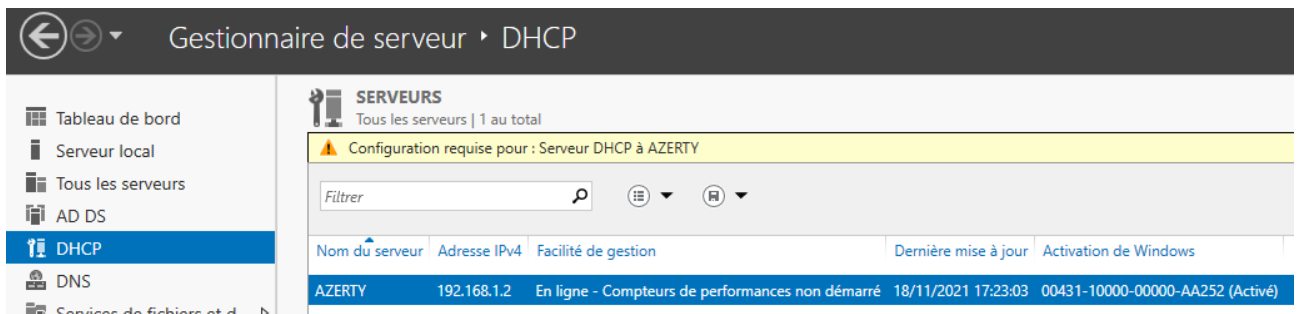


Spécifications de l'appareil

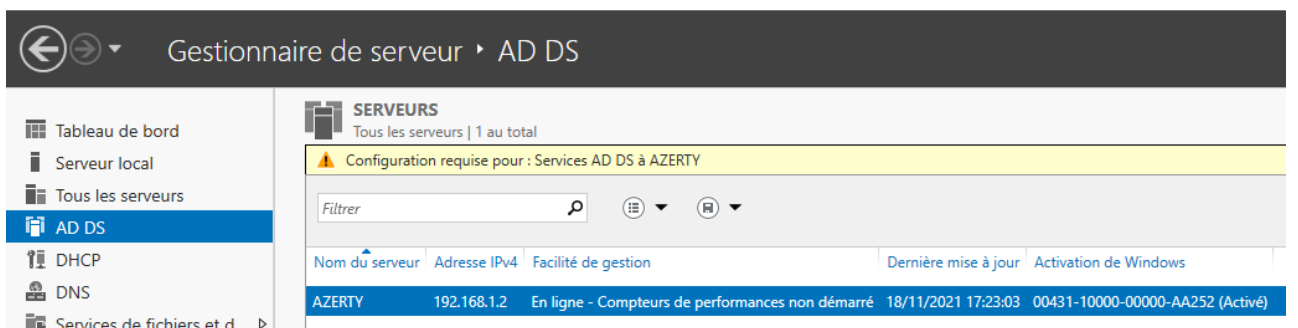
Nom de l'appareil

AZERTY

DHCP :

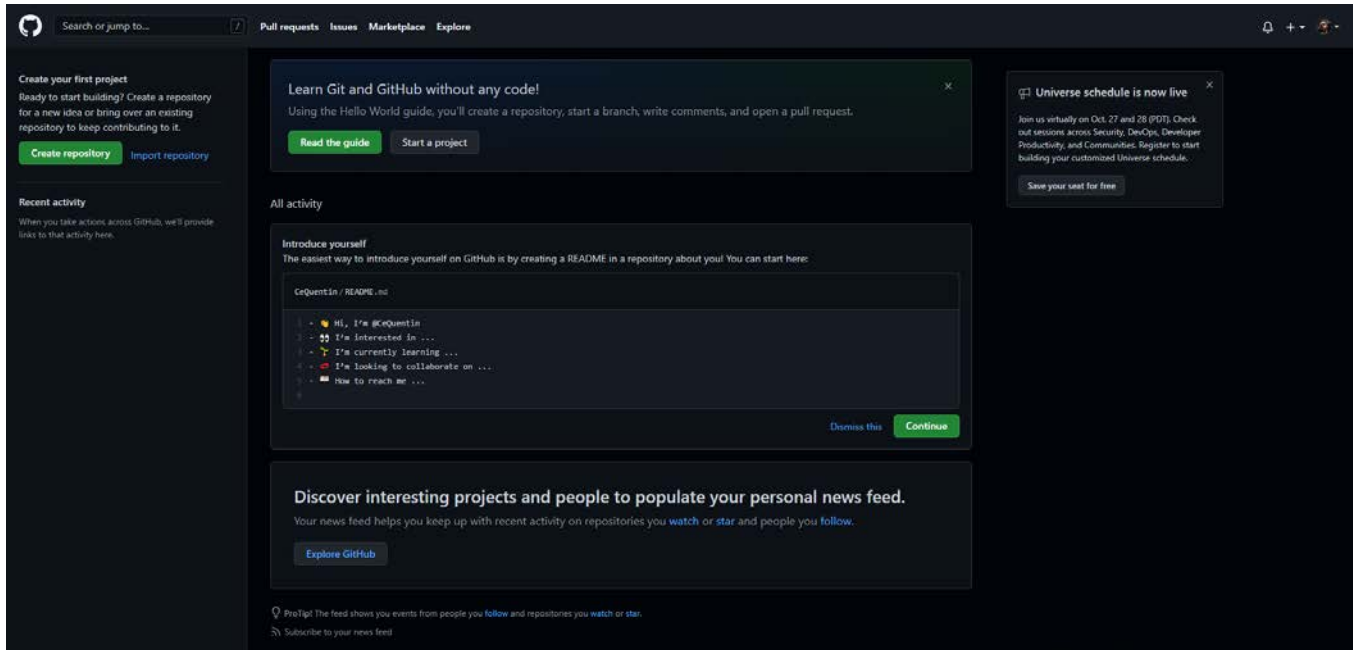


AD :



GitHub

Création du compte GitHub :

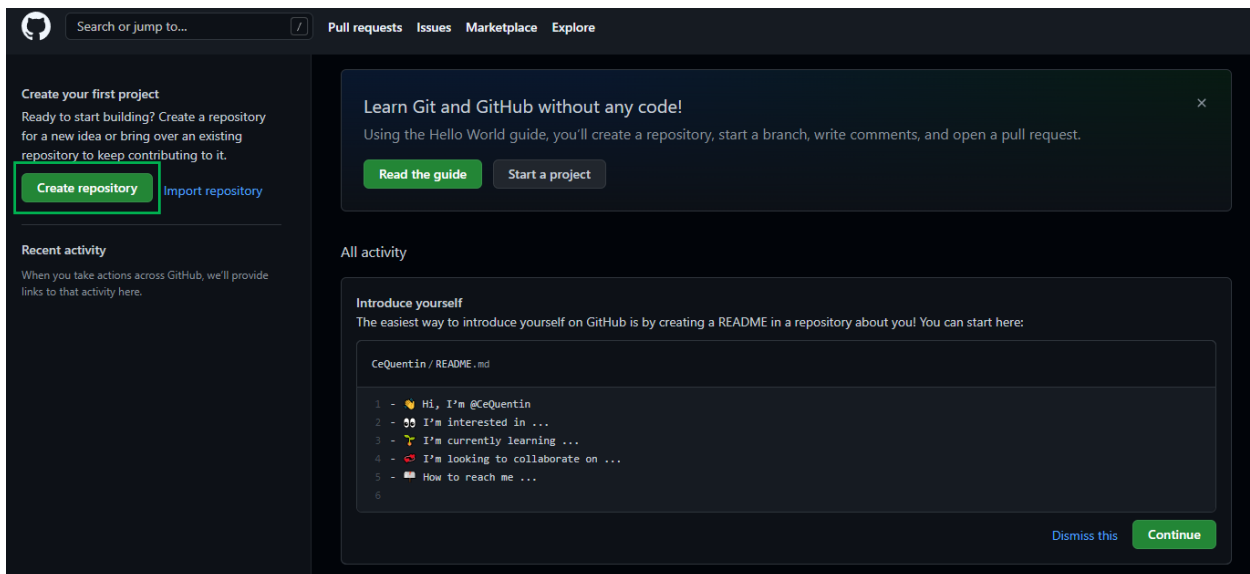


Voici mon compte GitHub avec le lien ci-dessous.

<https://github.com/CeQuentin/Projet-6-Configuration-Windows-Serveur-2016>

Création du répertoire GitHub :

Pour créer un nouveau répertoire il faut cliquer en haut à gauche sur « **Create repository** ».



Ensuite il faut choisir le nom du répertoire ainsi que les option de ce dernier.

The screenshot shows the 'Create a new repository' page on GitHub. The 'Owner' is 'CeQuentin'. The 'Repository name' is 'Projet 6 : Configuration Windows', which is highlighted with a green checkmark. A tooltip indicates that the repository will be created as 'Projet-6-Configuration-Windows-Serveur-2016'. The 'Description' field is empty. The 'Public' option is selected, indicating that anyone on the internet can see the repository. The 'Initialize this repository with' section has three options: 'Add a README file' (checked), 'Add .gitignore' (unchecked), and 'Choose a license' (unchecked). A green 'Create repository' button is at the bottom.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner **CeQuentin** / Repository name **Projet 6 : Configuration Windows** ✓

Great repository names are [Projet-6-Configuration-Windows-Serveur-2016](#). Your new repository will be created as **Projet-6-Configuration-Windows-Serveur-2016**.

Description (optional)

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

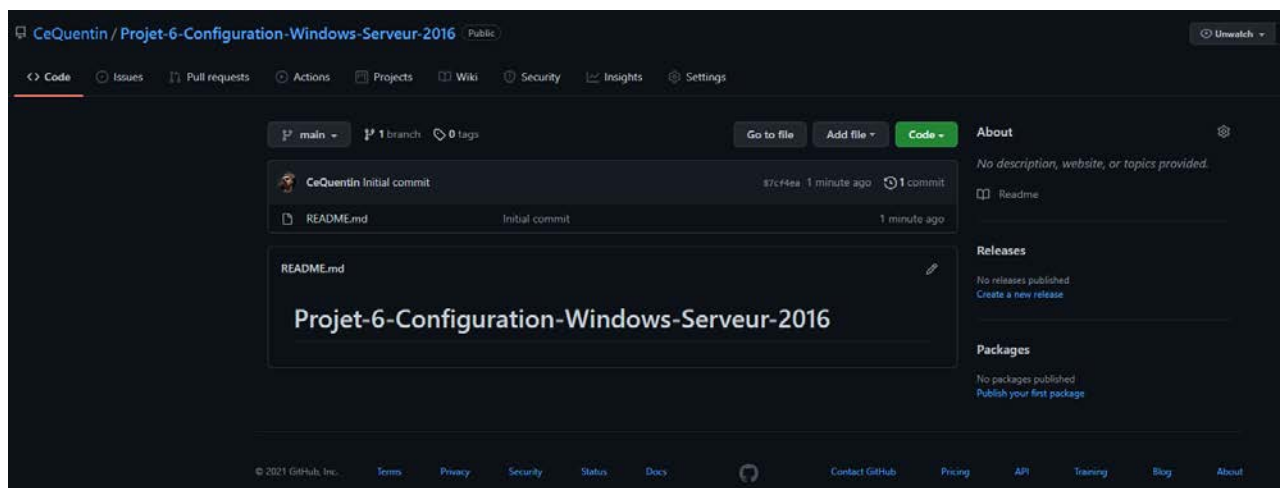
☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

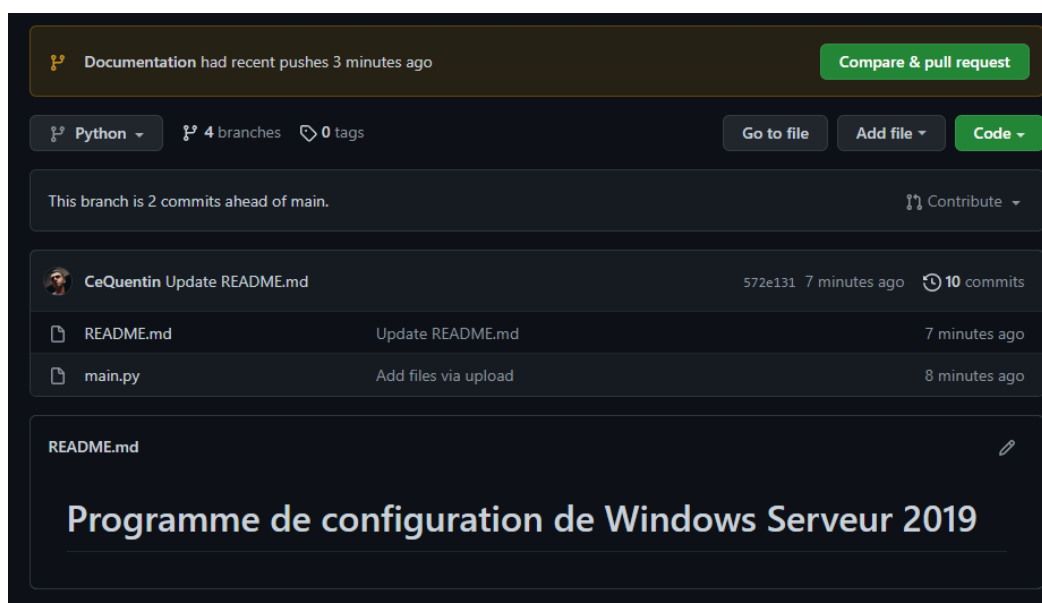
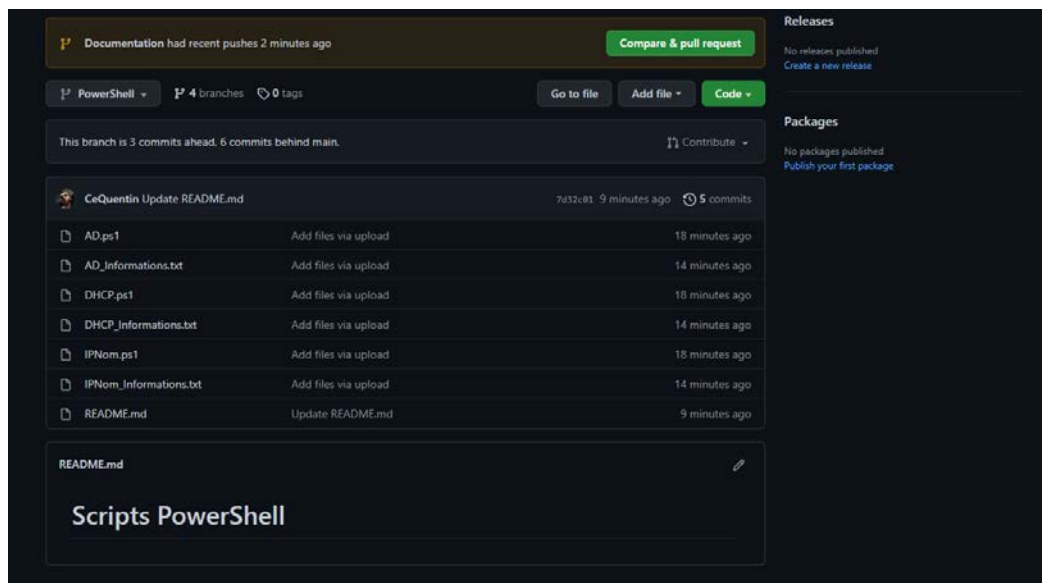
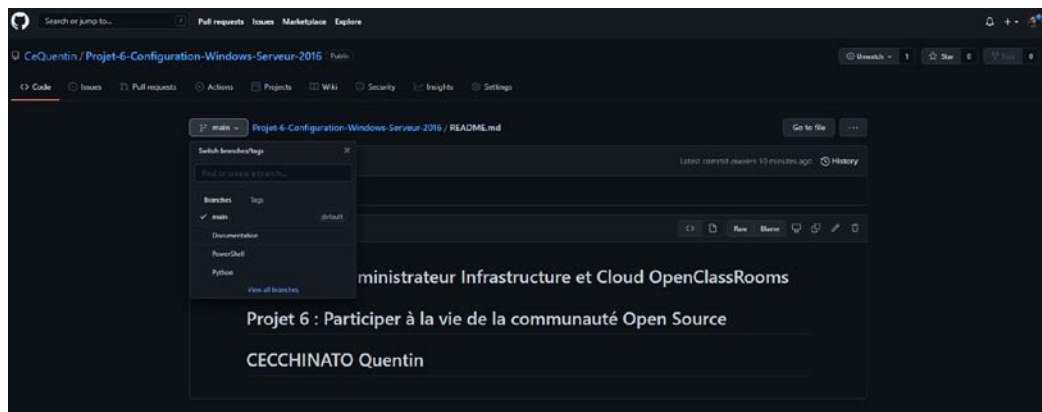
This will set `main` as the default branch. Change the default name in your [settings](#).

Create repository

Et voilà le résultat, le répertoire a été créé.



Répertoire GitHub :



Conclusion

Dans ce projet j'ai dû travailler sur la création d'un programme d'automatisation. Je suis parti sur un programme qui appelle des scripts PowerShell pour pouvoir configurer un serveur Windows. Enfin j'ai pu faire fonctionner ce programme en le testant sur un serveur Windows 2019.

Annexe

Au début du projet j'avais pensé à faire un quatrième algorithme pour pouvoir installer un serveur depuis la sélection de langue jusqu'au bureau, mais j'ai vu que ce n'était pas possible.

Algorithme : InstallerServeur()

Rôle : Installation et mise en place du serveur Windows depuis la sélection des langues jusqu'à l'arrivée sur le bureau.

(Une fois l'installation du Windows Serveur 2016 lancée)

Début

Cliquer 'suivant'

Cliquer 'Installer maintenant'

Choisir entre 'Standard(1)' ou 'Datacenter(3)'

Cliquer 'suivant'

Cliquer 'J'accepte les termes du contrat de licence'

Cliquer 'suivant'

Cliquer 'personnaliser'

Cliquer 'choisir disque dur'

Cliquer 'suivant'

##une fois redémarré##

Entrer : Mot de passe

Entrer : Confirmation du mot de passe

Cliquer 'Terminer'

Fin

Lexique

echo : Affiche une phrase à l'écran

\$Variables : C'est où l'on stocke des données à rentrer

Get-NetIPConfiguration : Permet d'obtenir la configuration réseau

New-NetIPAddress : Permet de créer et configurer une adressage IP

Set-DnsClientServerAddress : permet d'ajouter statiquement une adresse DNS

Read-Host : permet de rentrer les information demandé (dans des variables)

Rename-Computer : Permet de renommer une machine

Install-WindowsFeature : Permet d'installer des fonctionnalités Windows

Add-DhcpServerSecurityGroup : Permet de créer un groupe de sécurité DHCP

Restart-Service : Permet de redémarrer un service

Import-Module : Permet d'importer un module

Install-ADDSForest : Permet d'installer une forêt