

Lab 7 – Using PowerShell for Bulk Users Creation

■ Prep Step: Prepare the PowerShell environment
System: DC107

Command:

```
Set-ExecutionPolicy RemoteSigned -Scope Process
```

Explanation:

Before running any script, I ensured the environment was configured properly.

- `Set-ExecutionPolicy RemoteSigned` allows me to run `*.ps1` files **in** this session without modifying global settings

Screenshot:

```
PS C:\> Get-ExecutionPolicy
RemoteSigned
```

■ Task 1: Prepare users.csv file

System: DC107

Steps:

- I copied the provided file `users.csv` to the root of the C:\ drive.
- I opened the file **using** Notepad to verify formatting and content.

Validation:

- The first line contains exactly the headers:
FIRST NAME;LAST NAME;DEPARTEMENT
- Each subsequent line represents one user.
- All fields are separated by semicolons (;).

Sample content:

FIRST NAME;LAST NAME;DEPARTEMENT

Alice;Martin;IT

Bob;Johnson;HR

Carol;White;Finance

- This file will be read **using** `Import-CSV`
- Each row becomes a PowerShell object with accessible properties (`\$user.FIRSTNAME`, etc.)

The file is saved **in** the same folder as the script.

Screenshot :

The screenshot shows three windows side-by-side:

- File Explorer:** Shows the contents of Local Disk (C:). A red box highlights the 'addusers.ps1' file in the 'This PC' folder.
- Notepad:** Titled "users - Notepad", it displays a list of user names separated by semicolons, such as Adam;Deschamps;Sales, Amaury;Fleury;Sales, etc.
- PowerShell:** A terminal window showing the command PS C:\Users\Administrator> Get-Content C:\users.csv followed by a list of user names in CSV format.

Name	Date modified	Type	Size
\$WinREAgent	4/29/2025 5:06 PM	File folder	
PerfLogs	5/8/2021 4:20 AM	File folder	
Program Files	4/29/2025 3:04 PM	File folder	
Program Files (x86)	5/8/2021 5:42 AM	File folder	
ProgramData	5/1/2025 8:59 PM	File folder	
Users	4/29/2025 3:04 PM	File folder	
Windows	5/1/2025 8:55 PM	File folder	
addusers.ps1	5/12/2025 1:12 PM	Windows PowerShell	3 KB
cleanAD	5/12/2025 1:12 PM	Windows PowerShell	2 KB
users.csv	5/12/2025 1:12 PM	CSV File	5 KB

■ Task 2: Execute addusers.ps1 to automate user creation

System: DC107

Steps:

- I opened Windows PowerShell ISE.
- I pasted the contents of the provided script `addusers.ps1` into the editor.
- I modified the following lines **in** the script to reflect my actual environment:

1. Set the path to the CSV file:
`$csvPath = "C:\users.csv"`
2. Set the domain Distinguished Name:
`$domainDN = "DC=vlabs07,DC=com"`
3. Set the UPN (User Principal Name) domain:
`$userPrincipalName = "$userName@vlabs07.com"`

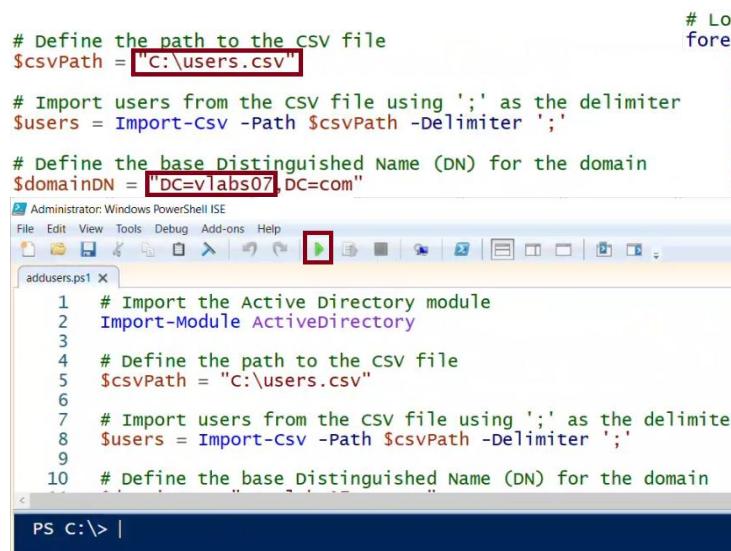
- I saved the modified script as:
`C:\addusers.ps1 (for backup)`
- I ran the script **directly from PowerShell ISE** by pressing **F5** (or clicking the green "Run Script" button)

Explanation:

The script automates user creation based on `users.csv` and includes:

- Reading user **data using** `Import-Csv -Delimiter ';'`
- Looping through each user with `foreach`
- Creating a username (**first initial + last name, lowercase**)
- Verifying/creating the users department OU
- Creating the user with:
 - GivenName, Surname, Department
 - SamAccountName and UserPrincipalName (**UPN**)
 - Default password: Passw0rd\$
 - Enabled = True
- Moving user to their department OU
- Creating and populating the correct security group **for** each department

Screenshot:



```
# Define the path to the csv file
$csvPath = "C:\users.csv"

# Import users from the csv file using ';' as the delimiter
$users = Import-Csv -Path $csvPath -Delimiter ';'

# Define the base Distinguished Name (DN) for the domain
$domainDN = "DC=vlabs07,DC=com"

# Loop through each user record in the csv file
foreach ($user in $users) {
    $firstName = $user.'FIRST NAME'
    $lastName = $user.'LAST NAME'
    $department = $user.DEPARTEMENT
    $userName = "$firstName.$lastName"
    $displayName = "$firstName $lastName"
    $ouPath = "OU=$department,$domainDN"
    $userPrincipalName = "$userName@vlabs07.com"
}

# Import the Active Directory module
Import-Module ActiveDirectory
```

■ Task 3: Validate OU, User, and Group Creation

System: DC107

Validation (PowerShell):

```
# Check that each department has a corresponding OU
Get-ADOrganizationalUnit -Filter * | Select Name, DistinguishedName
PS C:\Users\Administrator> Get-ADOrganizationalUnit -Filter * | Select Name, DistinguishedName

Name          DistinguishedName
----          -----
Domain Controllers  OU=Domain Controllers,DC=vlabs07,DC=com
HR            OU=HR,DC=vlabs07,DC=com
IT            OU=IT,DC=vlabs07,DC=com
Shared Resources  OU=Shared Resources,DC=vlabs07,DC=com
Workstations    OU=Workstations,DC=vlabs07,DC=com
Servers         OU=Servers,DC=vlabs07,DC=com
IT Services     OU=IT Services,DC=vlabs07,DC=com
Sales           OU=Sales,DC=vlabs07,DC=com
Call Center     OU=Call Center,DC=vlabs07,DC=com
Accounting      OU=Accounting,DC=vlabs07,DC=com
Finance         OU=Finance,DC=vlabs07,DC=com
Engineering     OU=Engineering,DC=vlabs07,DC=com
Purchases       OU=Purchases,DC=vlabs07,DC=com

# Confirm each user account was created
Get-ADUser -Filter * | Select SamAccountName, GivenName, Surname, Department,
Enabled, DistinguishedName
PS C:\Users\Administrator> Get-ADUser -Filter * | Select SamAccountName, GivenName, Surname, Department,
Enabled, DistinguishedName

SamAccountName : Administrator
GivenName       :
Surname         :
Department      :
Enabled         : True
DistinguishedName: CN=Administrator,CN=Users,DC=vlabs07,DC=com

SamAccountName : Guest
GivenName       :
Surname         :
Department      :
Enabled         : False
DistinguishedName: CN=Guest,CN=Users,DC=vlabs07,DC=com

# Confirm security groups were created
Get-ADGroup -Filter * | Select Name, GroupScope, DistinguishedName
PS C:\Users\Administrator> Get-ADGroup -Filter * | Select Name, GroupScope, DistinguishedName

GG_HR_Admins   Global CN=GG_HR_Admins,OU=HR,DC=vlabs07,DC=com
GG_IT_Admins   Global CN=GG_IT_Admins,OU=IT,DC=vlabs07,DC=com
DLG_HR_Share   DomainLocal CN=DLG_HR_Share,OU=Shared Resources,DC=vlabs07...
DLG_IT_Share   DomainLocal CN=DLG_IT_Share,OU=Shared Resources,DC=vlabs07...
UG_IT_Global   Global CN=UG_IT_Global,DC=vlabs07,DC=com
Sales          Global CN=Sales,OU=Sales,DC=vlabs07,DC=com
Call Center    Global CN=Call Center,OU=Call Center,DC=vlabs07,DC=com
Accounting     Global CN=Accounting,OU=Accounting,DC=vlabs07,DC=com
Finance        Global CN=Finance,OU=Finance,DC=vlabs07,DC=com
IT             Global CN=IT,OU=IT,DC=vlabs07,DC=com
Engineering    Global CN=Engineering,OU=Engineering,DC=vlabs07,DC=com
HR             Global CN=HR,OU=HR,DC=vlabs07,DC=com
Purchases      Global CN=Purchases,OU=Purchases,DC=vlabs07,DC=com
```

```
# Check group memberships (repeat per department)

Get-ADGroupMember -Identity "IT"
PS C:\Users\Administrator> Get-ADGroupMember -Identity "IT"

distinguishedName : CN=Aloyse Dupont,OU=IT,DC=vlabs07,DC=com
name              : Aloyse Dupont
objectClass       : user
objectGUID        : ea4d18f1-31a4-4133-9a30-e0226c5ad8e5
SamAccountName   : Aloyse.Dupont
SID               : S-1-5-21-2428485534-1961598418-1246186656-1224

distinguishedName : CN=Augustine Andre_,OU=IT,DC=vlabs07,DC=com
name              : Augustine Andre_
objectClass       : user
objectGUID        : 364baee4-e78d-4b62-a8f0-04cad4de68f3
SamAccountName   : Augustine.Andre_
SID               : S-1-5-21-2428485534-1961598418-1246186656-1226

Get-ADGroupMember -Identity "HR"
PS C:\Users\Administrator> Get-ADGroupMember -Identity "HR"

distinguishedName : CN=Adem Vasseur,OU=HR,DC=vlabs07,DC=com
name              : Adem Vasseur
objectClass       : user
objectGUID        : 87e87031-1d6e-4841-a37d-da8010519c60
SamAccountName   : Adem.Vasseur
SID               : S-1-5-21-2428485534-1961598418-1246186656-1279

distinguishedName : CN=Ambre Rousseau,OU=HR,DC=vlabs07,DC=com
name              : Ambre Rousseau
objectClass       : user
objectGUID        : d3c4161b-b440-4158-a8ea-d524651bdc99
SamAccountName   : Ambre.Rousseau
SID               : S-1-5-21-2428485534-1961598418-1246186656-1281

Get-ADGroupMember -Identity "Finance"
PS C:\Users\Administrator> Get-ADGroupMember -Identity "Finance"

distinguishedName : CN=Alienor Lambert,OU=Finance,DC=vlabs07,DC=com
name              : Alienor Lambert
objectClass       : user
objectGUID        : a8021a6d-5cbd-4f0c-9c27-45dc8c7218fb
SamAccountName   : Alienor.Lambert
SID               : S-1-5-21-2428485534-1961598418-1246186656-1196

distinguishedName : CN=Ava Mercier,OU=Finance,DC=vlabs07,DC=com
name              : Ava Mercier
objectClass       : user
objectGUID        : 8c576575-ea38-4118-8a39-be1e87c13461
SamAccountName   : Ava.Mercier
SID               : S-1-5-21-2428485534-1961598418-1246186656-1198
```

```
Get-ADGroupMember -Identity "Sales"
PS C:\Users\Administrator> Get-ADGroupMember -Identity "Sales"

distinguishedName : CN=Adam Deschamps,OU=Sales,DC=vlabs07,DC=com
name              : Adam Deschamps
objectClass       : user
objectGUID        : 94d8beff-4fa3-4c10-9341-57ae25b07e5b
SamAccountName   : Adam.Deschamps
SID               : S-1-5-21-2428485534-1961598418-1246186656-1121

distinguishedName : CN=Amaury Fleury,OU=Sales,DC=vlabs07,DC=com
name              : Amaury Fleury
objectClass       : user
objectGUID        : f570f71c-973f-4d0b-b61d-5e06d39771e9
SamAccountName   : Amaury.Fleury
SID               : S-1-5-21-2428485534-1961598418-1246186656-1123

Get-ADGroupMember -Identity "Call Center"
PS C:\Users\Administrator> Get-ADGroupMember -Identity "Call Center"

distinguishedName : CN=Aaron Louis,OU=Call Center,DC=vlabs07,DC=com
name              : Aaron Louis
objectClass       : user
objectGUID        : a4d9ad9b-5167-4e2a-afbb-efa7a35c58c1
SamAccountName   : Aaron.Louis
SID               : S-1-5-21-2428485534-1961598418-1246186656-1149

distinguishedName : CN=Aloès Boucher,OU=Call Center,DC=vlabs07,DC=com
name              : Aloès Boucher
objectClass       : user
objectGUID        : 064387ff-09f7-4733-bee1-00cdbbc1937f4
SamAccountName   : Aloès.Boucher
SID               : S-1-5-21-2428485534-1961598418-1246186656-1151

Get-ADGroupMember -Identity "Accounting"
PS C:\Users\Administrator> Get-ADGroupMember -Identity "Accounting"

distinguishedName : CN=Alix Fontaine,OU=Accounting,DC=vlabs07,DC=com
name              : Alix Fontaine
objectClass       : user
objectGUID        : 55dd30f3-8b85-42f1-8e00-4e2ab9d266b4
SamAccountName   : Alix.Fontaine
SID               : S-1-5-21-2428485534-1961598418-1246186656-1173

distinguishedName : CN=Axelle Blanc,OU=Accounting,DC=vlabs07,DC=com
name              : Axelle Blanc
objectClass       : user
objectGUID        : 7d74a299-3ce0-4d10-a0bd-bed055c9f3c2
SamAccountName   : Axelle.Blanc
SID               : S-1-5-21-2428485534-1961598418-1246186656-1175
```

```

Get-ADGroupMember -Identity "Engineering"
PS C:\Users\Administrator> Get-ADGroupMember -Identity "Engineering"

distinguishedName : CN=Agathe Bonnet,OU=Engineering,DC=vlabs07,DC=com
name              : Agathe Bonnet
objectClass       : user
objectGUID        : ae9fca28-b6a5-48c3-bacd-09a5d00f67e7
SamAccountName   : Agathe.Bonnet
SID               : S-1-5-21-2428485534-1961598418-1246186656-1253

distinguishedName : CN=Alexandre Perez,OU=Engineering,DC=vlabs07,DC=com
name              : Alexandre Perez
objectClass       : user
objectGUID        : e51a51e2-874d-4c60-b1d5-1b701508755c
SamAccountName   : Alexandre.Perez
SID               : S-1-5-21-2428485534-1961598418-1246186656-1255

Get-ADGroupMember -Identity "Purchases"
PS C:\Users\Administrator> Get-ADGroupMember -Identity "Purchases"

distinguishedName : CN=Adrien Huet,OU=Purchases,DC=vlabs07,DC=com
name              : Adrien Huet
objectClass       : user
objectGUID        : 2602f9b2-48a9-40e9-8a38-4323d2153b12
SamAccountName   : Adrien.Huet
SID               : S-1-5-21-2428485534-1961598418-1246186656-1304

distinguishedName : CN=Anael Royer,OU=Purchases,DC=vlabs07,DC=com
name              : Anael Royer
objectClass       : user
objectGUID        : b5e294e2-c015-45dc-9ae8-beab87c2243d
SamAccountName   : Anael.Royer
SID               : S-1-5-21-2428485534-1961598418-1246186656-1306

```

Confirm all accounts are enabled

```

Get-ADUser -Filter * | Select SamAccountName, Enabled

List only accounts that are NOT enabled (for verification)
Get-ADUser -Filter * | Where-Object { $_.Enabled -eq $false }

```

Explanation:

I used these commands to verify the "Enabled" status of all users.
This ensures that the accounts created by the script are active and usable.

Results:

- The majority of user accounts created from users.csv are shown as Enabled = True.
- Disabled accounts (like Guest, krbtgt, hr.template) are system/template accounts that are **not part of the lab script**.
- No user account created by the script appeared in the disabled list.

Screenshots:

```
PS C:\Users\Administrator> Get-ADUser -Filter * | Select SamAccountName, Enabled
SamAccountName      Enabled
-----
Administrator        True
Guest                False
krbtgt               False
krbtgt_443           False
LAB07$              True
s.lambert            True
e.morel              False
l.bernard             True
c.girard              True
hr.template          False
Adam.Deschamps       True
Amaury.Fleury        True
Andrea.Lefèvre       True
Arnaud.Adam           True
Cesar.Renault         True
PS C:\Users\Administrator> Get-ADUser -Filter * | Where-Object { $_.Enabled -eq $false }

DistinguishedName : CN=Guest,CN=Users,DC=vlabs07,DC=com
Enabled           : False
GivenName          :
Name               : Guest
ObjectClass        : user
ObjectGUID         : 1cdf9a94-994f-4d87-98a2-cbf518297b8b
SamAccountName     : Guest
SID                : S-1-5-21-242848534-1961598418-1246186656-501
Surname            :
UserPrincipalName  :
```

Explanation:

I used PowerShell and Active Directory Users and Computers (**ADUC**) to validate the following:

1. Each department listed **in** users.csv has a corresponding OU created **in** AD.
2. Each user account exists **in** Active Directory.
3. The username format follows the correct convention **(first letter of first name + full last name)**.
4. Each account is located inside the appropriate OU **for** their department.
5. Each account is enabled.
6. Each department has a matching security **group** (e.g., IT, HR).
7. Each user is a member of the corresponding **group**.
8. OU structure and DistinguishedName paths are correct.

Manual GUI Validation (ADUC):

Steps Performed:

- Opened Active Directory Users and Computers (**Start** → Run → dsa.msc).
- Enabled "Advanced Features" from the View menu.
- Navigated to the domain tree: vlabs07.com.
- Expanded and inspected each Organizational Unit (OU) created by the script.
- Opened **group** objects to check members.
- Opened user objects to inspect account settings and attributes.

Screenshots Captured:

1 - OU Structure

Name	Type	Description
Accounting	Organizational Unit	
Builtin	builtinDomain	
Call Center	Organizational Unit	Default container for up...
Computers	Container	Default container for do...
Domain Controll...	Organizational Unit	
Engineering	Organizational Unit	
Finance	Organizational Unit	
ForeignSecurityPr...	Container	Default container for sec...
HR	Organizational Unit	
IT	Organizational Unit	Handles IT operations an...
IT Services	Organizational Unit	Default container for key...
Keys	Container	Default container for opr...
LostAndFound	lostAndFound	Default container for ma...
Managed Service Accounts	Container	Default location for stora...
Program Data	Container	
Purchases	Organizational Unit	
Sales	Organizational Unit	
Servers	Organizational Unit	
Shared Resources	Organizational Unit	
System	Container	Builtin system settings
Users	Container	Default container for up...
Workstations	Organizational Unit	
NTDS Quotas	msDS-QuotaContainer	Quota specifications con...
TPM Devices	msTPM-InformationOb...	
Infrastructure	infrastructureUpdate	
UG_IT_Global	Security Group - Global	

2 - Users Inside OU

Name	Type
Accounting	Security Group - Global
Alix Fontaine	User
Axelle Blanc	User
Baptiste Aubry	User
Clara Thomas	User
Cylia François	User
Dorian Hervé	User
eleonore Nicolas	User
Enzo Simon	User
Etienne Bouvier	User
Garance Denis	User
Ilan Germain	User
Joy Brun	User
Kevin Levêque	User
Lana Joly	User
Lily Arnaud	User
Lucie Colin	User
Martin Monnier	User
Noah Fournier	User
Nolan Chauvin	User
Océane Bourgeois	User
Suzanne Jean	User
Tiago Lamy	User

3 – Group Membership

The screenshot shows the Active Directory interface. On the left, there's a navigation pane with 'IT' selected. The main area displays the 'IT Properties' dialog. Under the 'Members' tab, a list of users is shown, all belonging to the 'IT' group. A red box highlights the 'Members' section.

Name	Type
GG_IT_Admins	Security Group - Global
Gildas Leroux	User
Isaline Dufour	User
IT	Security Group - Global

IT Properties

Members:

Name	Type
Aloyse Dupont	vlabs07.com/IT
Augustine An...	vlabs07.com/IT
Axel Poirier	vlabs07.com/IT
Camille Martin	vlabs07.com/IT
Capucine Gar...	vlabs07.com/IT
Clement Men...	vlabs07.com/IT
Diane Clement	vlabs07.com/IT
Erwan Collet	vlabs07.com/IT
Faustine Marc...	vlabs07.com/IT
Gildas Leroux	vlabs07.com/IT
Isaline Dufour	vlabs07.com/IT
Josselin LeGoff	vlabs07.com/IT
June Blanchard	vlabs07.com/IT

4 – User Account Details

The screenshot shows the Active Directory interface. On the left, there's a navigation pane with 'Adrien Huet' selected. The main area displays the 'Adrien Huet Properties' dialog. Under the 'General' tab, the 'User logon name' field is highlighted. It shows 'Adrien.Huet' in the primary field and '@vlabs07.com' in the dropdown.

Name	Type	Description
Adrien Huet		
Anael Royer		
Apolline Faure		
Arthur Paris		
Charlie Charpentier		
Corentin Bertin		
Diva Robin		
Eva Masson		
Gaspard Daniel		
Iris Meyer		
Jordan Remy		
Lilian Marchal		
Louane Schmitt		

Adrien Huet Properties

User logon name:
Adrien.Huet @vlabs07.com

User logon name (pre-Windows 2000):
VLABS07\Adrien.Huet

Brief Validation Summary:

- All users were created based on `users.csv` and placed **in** the correct department OUs.
- Each user account includes proper `GivenName`, `Surname`, and `Department` attributes.
- All accounts are enabled and follow the expected username format.
- A departmental `group` exists **for** each OU, and all users are added to the correct `group`.
- The OU and `group` structures match the intended Active Directory layout.