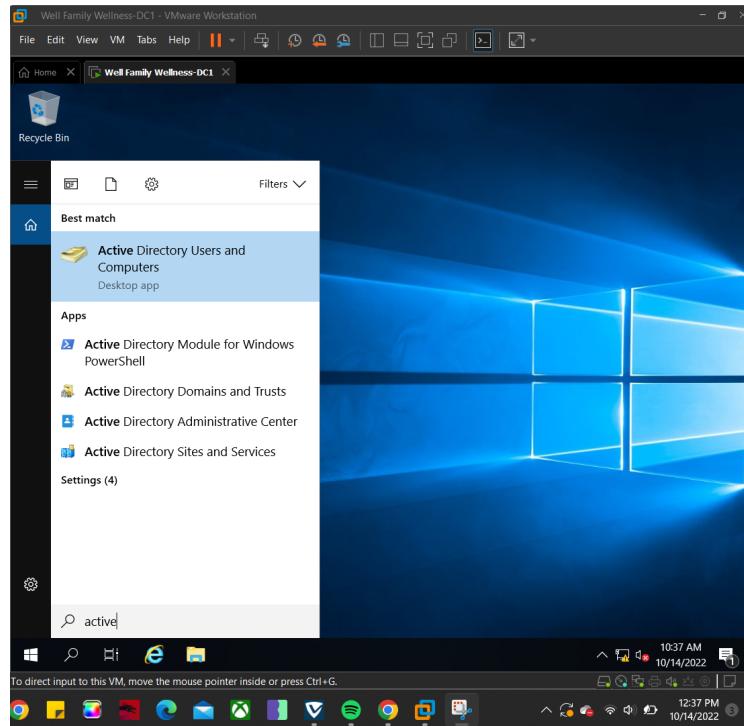


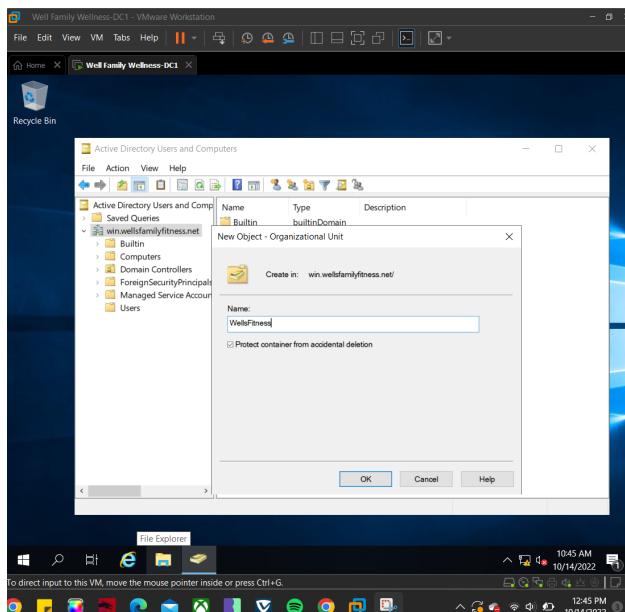
WINDOWS-

1. To add users and create groups, open the app, Active Directory Users and Computers.



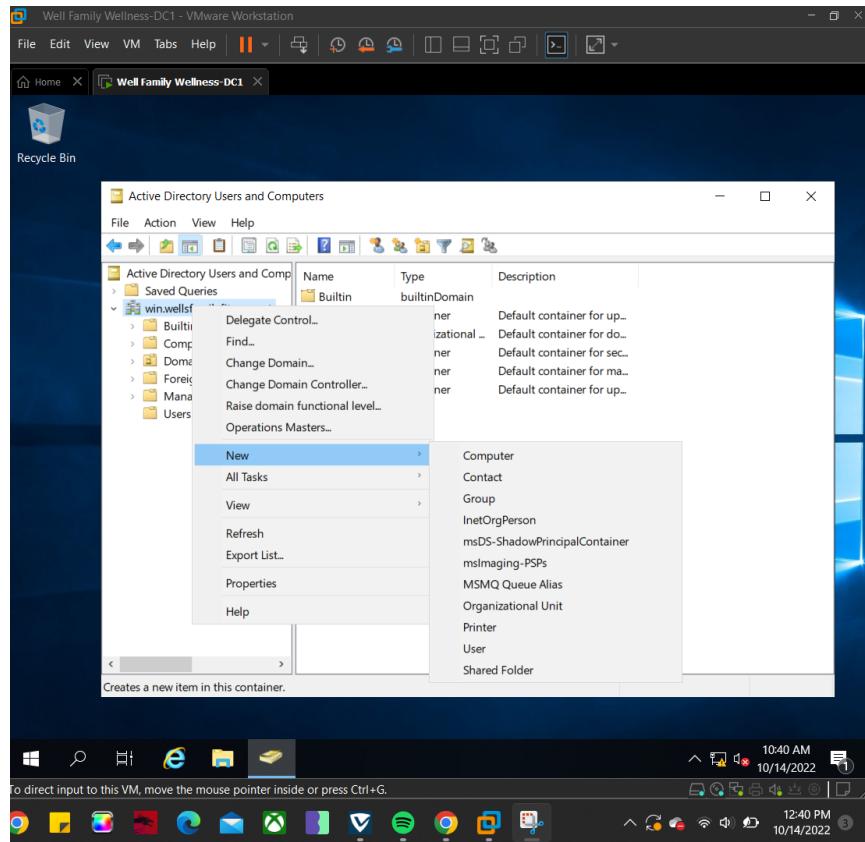
a.

2. Select your domain, right click and under New, Select *Organizational Unit*



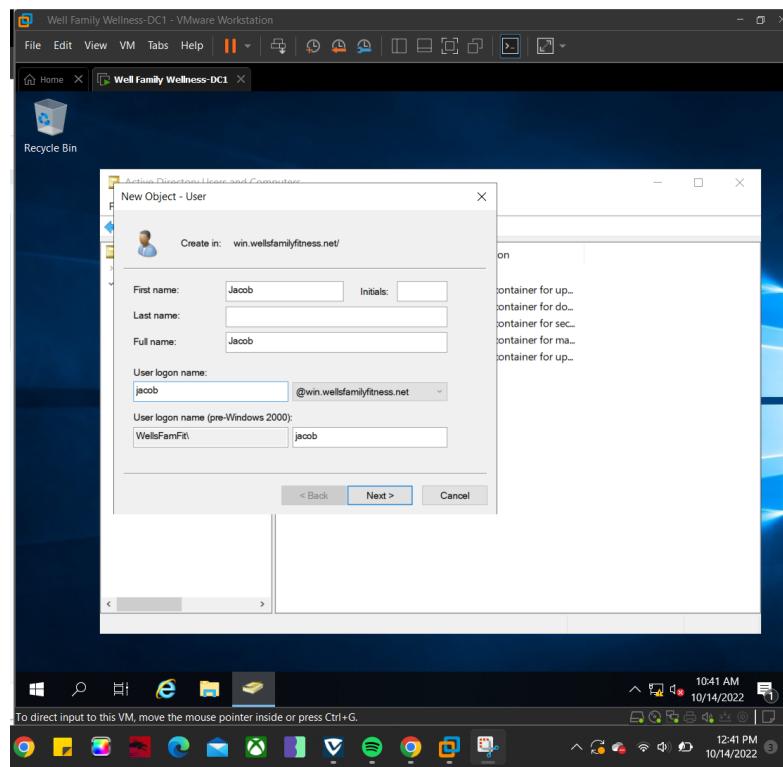
3.

4. Select your domain and right click, under the *new* option select *user*



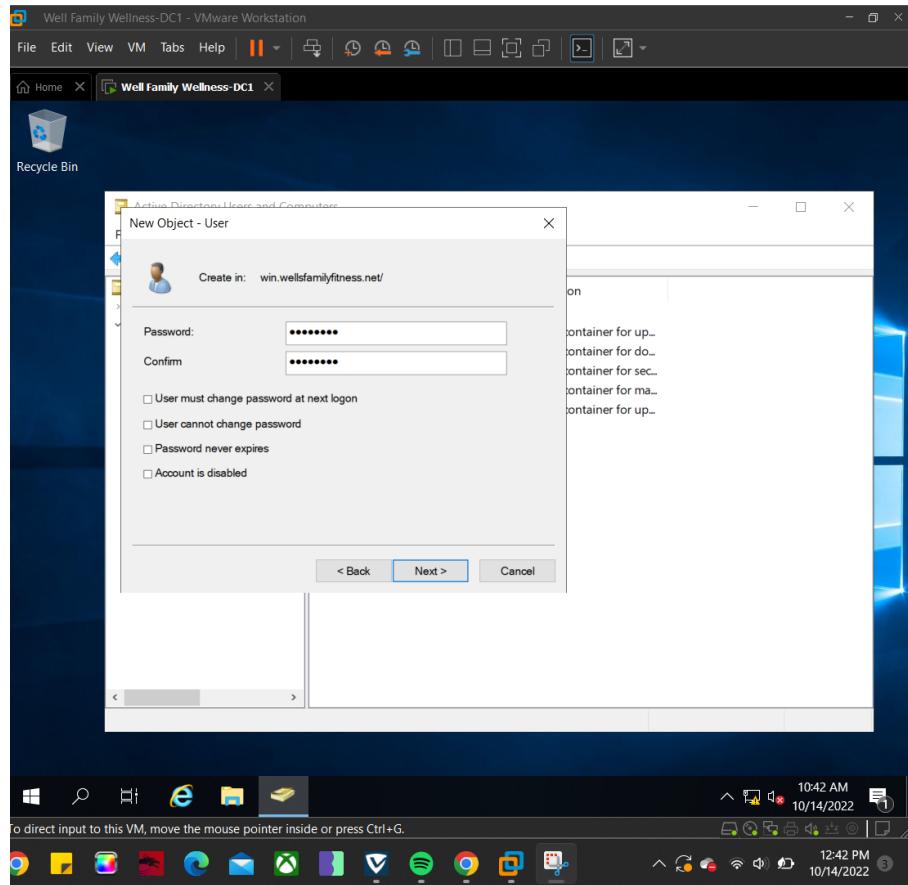
a.

5. Input the user's name in the appropriate boxes then select next



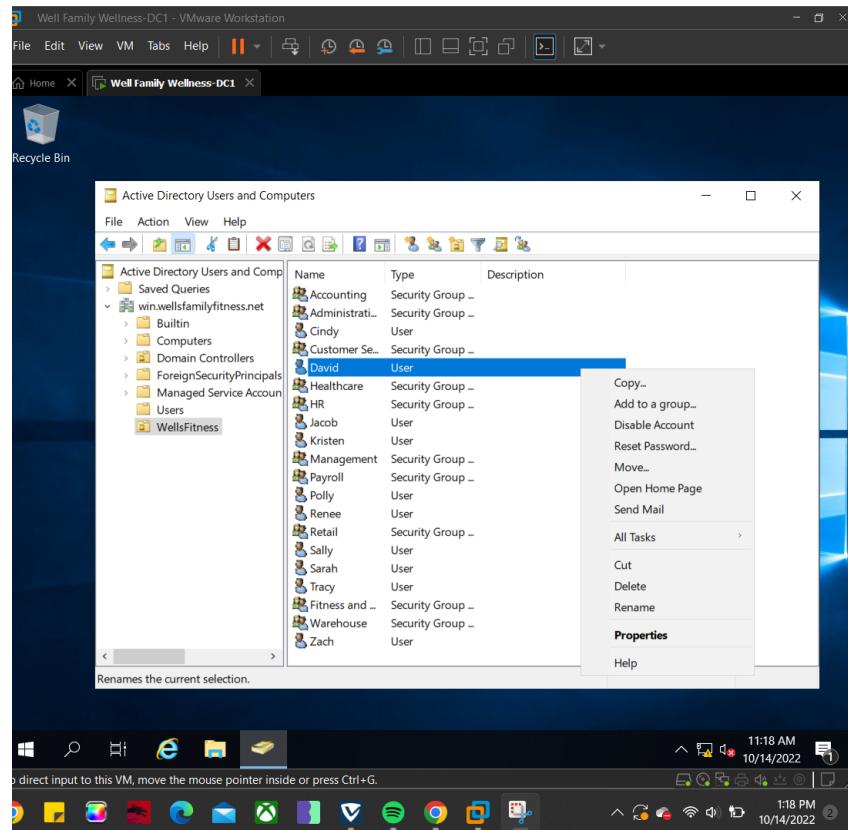
a.

6. Uncheck the box that requires the user to change their password and then type in password for that user

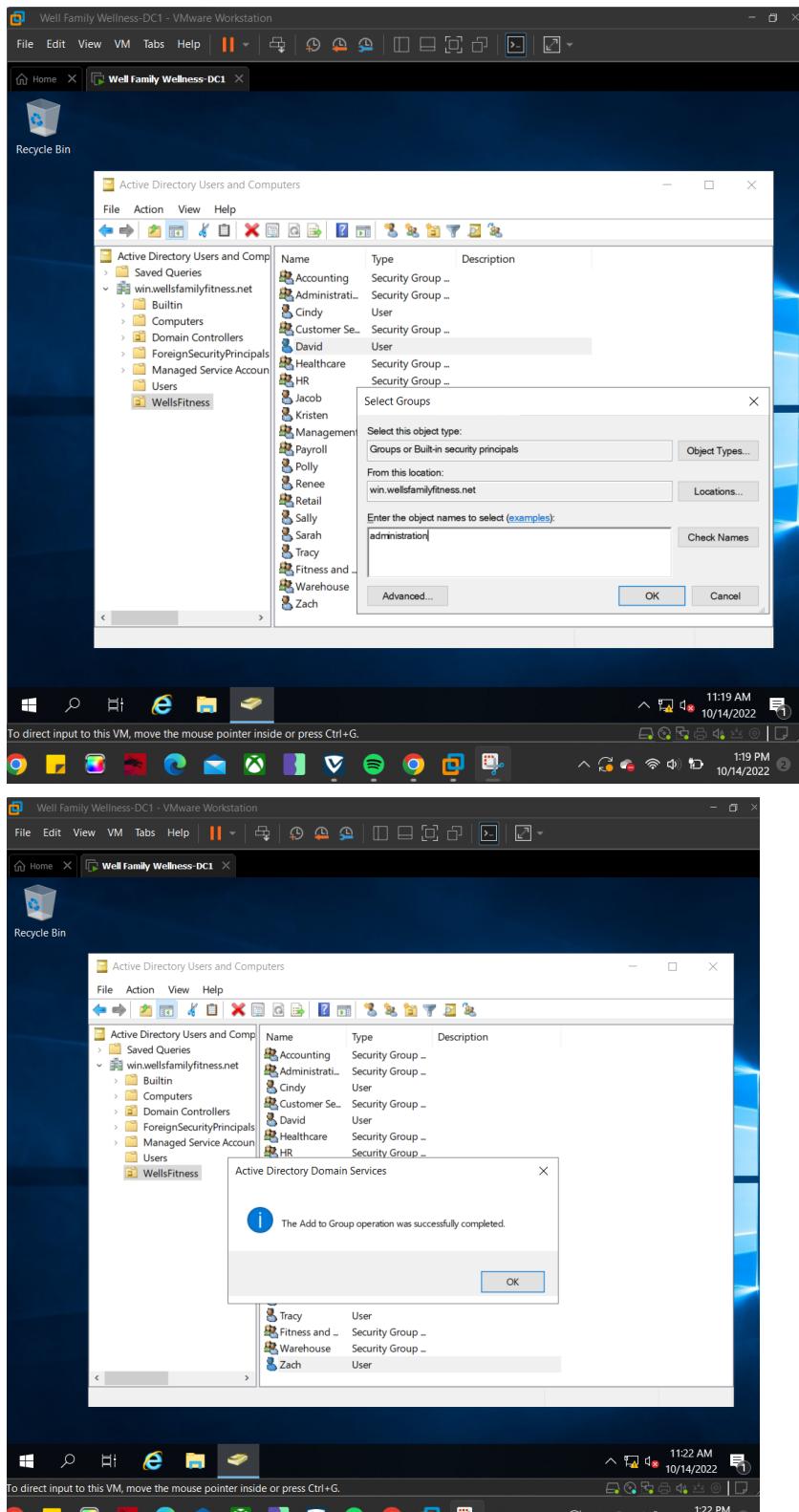


7. To add a user to a group right click on the user's name, select it *add to a group*

a.



8. Type in the name of the group you wish for the user to join and select ok (make sure that you spelled the group name correctly (it is not case sensitive))



- 9.

10. In powershell type in `get-aduser -filter {whencreated -ge $when} -properties whencreated | select name` This will show all the users recently created then copy it again but change `-aduser` to `-adgroup` which will show the different groups that are cr

Well Family Wellness-DC1 - VMware Workstation

File Edit View VM Tabs Help | || ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌃ ⌁ ⌂ ⌄ ⌅ ⌆ ⌇ ⌈ ⌉ ⌊ ⌋ ⌃ ⌁

Home Well Family Wellness-DC1

Administrator: Windows PowerShell

Tuesday, October 11, 2022 12:00:00 AM

```
PS C:\Users\Administrator> Get-aduser -filter {whencreated - ge $when} -properties whencreated | select name
Get-aduser : Error parsing query: 'whencreated - ge $when' Error Message: 'Operator Not supported: ' at position: '13'.
At line:1 char:1
+ Get-aduser -filter {whencreated - ge $when} -properties whencreated | ...
+-----+
+ CategoryInfo          : ParserError: () [Get-ADUser], ADFilterParsingException
+ FullyQualifiedErrorId : ActiveDirectoryCmdlet:Microsoft.ActiveDirectory.Management.ADFilterParsingException,Microsoft.ActiveDir
ect
ory.Management.Commands.GetADUser

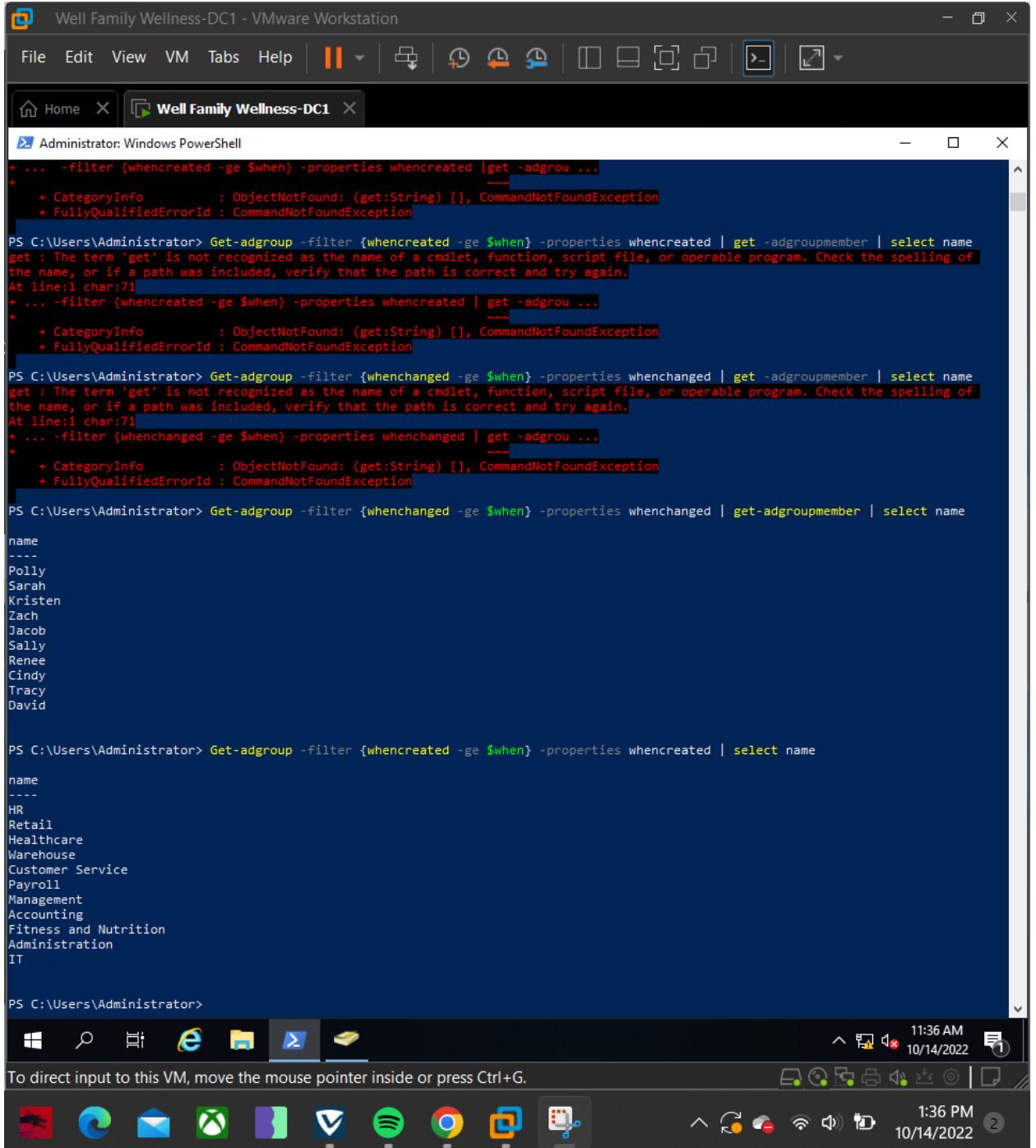
PS C:\Users\Administrator> Get-aduser -filter {whencreated -ge $when} -properties whencreated | select name
name
-----
Jacob
Sarah
David
Cindy
Zach
Sally
Kristen
Polly
Tracy
Renee

PS C:\Users\Administrator> Get-adgroup -filter {whencreated -ge $when} -properties whencreated | select name
name
-----
HR
Retail
Healthcare
Warehouse
Customer Service
Payroll
Management
Accounting
Fitness and Nutrition
Administration

PS C:\Users\Administrator> Get-adgroup -filter {whencreated -ge $when} -properties whencreated | get -adgroupmember | select name
get : The term 'get' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of
the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:70
+ ... -filter {whencreated -ge $when} -properties whencreated |get -adgrou ...
+-----+
+ CategoryInfo          : ObjectNotFound: (get:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Administrator> Get-adgroup -filter {whencreated -ge $when} -properties whencreated | get -adgroupmember | select name
get : The term 'get' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of
the name, or if a path was included, verify that the path is correct and try again.
```

11.



```
+ ... -filter {whencreated -ge $when} -properties whencreated | get -adgroupmember | select name
+ + CategoryInfo          : ObjectNotFound: (get:String) [], CommandNotFoundException
+ + FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Administrator> Get-adgroup -filter {whencreated -ge $when} -properties whencreated | get -adgroupmember | select name
get : The term 'get' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of
the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:71
+ ... -filter {whencreated -ge $when} -properties whencreated | get -adgroup ...
+ + CategoryInfo          : ObjectNotFound: (get:String) [], CommandNotFoundException
+ + FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Administrator> Get-adgroup -filter {whenchanged -ge $when} -properties whenchanged | get -adgroupmember | select name
get : The term 'get' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of
the name, or if a path was included, verify that the path is correct and try again.
At line:1 char:71
+ ... -filter {whenchanged -ge $when} -properties whenchanged | get -adgroup ...
+ + CategoryInfo          : ObjectNotFound: (get:String) [], CommandNotFoundException
+ + FullyQualifiedErrorId : CommandNotFoundException

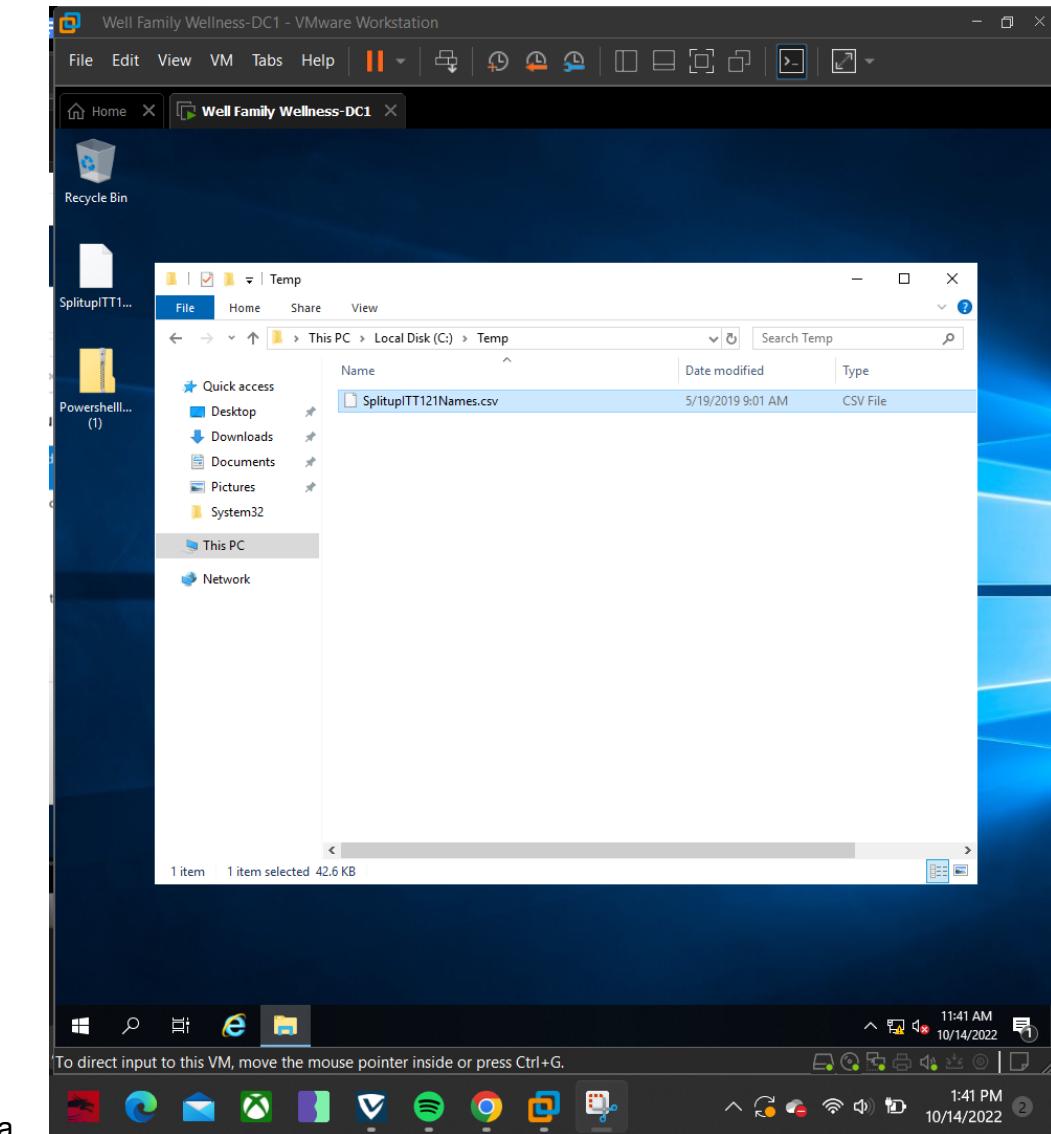
PS C:\Users\Administrator> Get-adgroup -filter {whenchanged -ge $when} -properties whenchanged | get-adgroupmember | select name
name
-----
Polly
Sarah
Kristen
Zach
Jacob
Sally
Renée
Cindy
Tracy
David

PS C:\Users\Administrator> Get-adgroup -filter {whencreated -ge $when} -properties whencreated | select name
name
-----
HR
Retail
Healthcare
Warehouse
Customer Service
Payroll
Management
Accounting
Fitness and Nutrition
Administration
IT

PS C:\Users\Administrator>
```

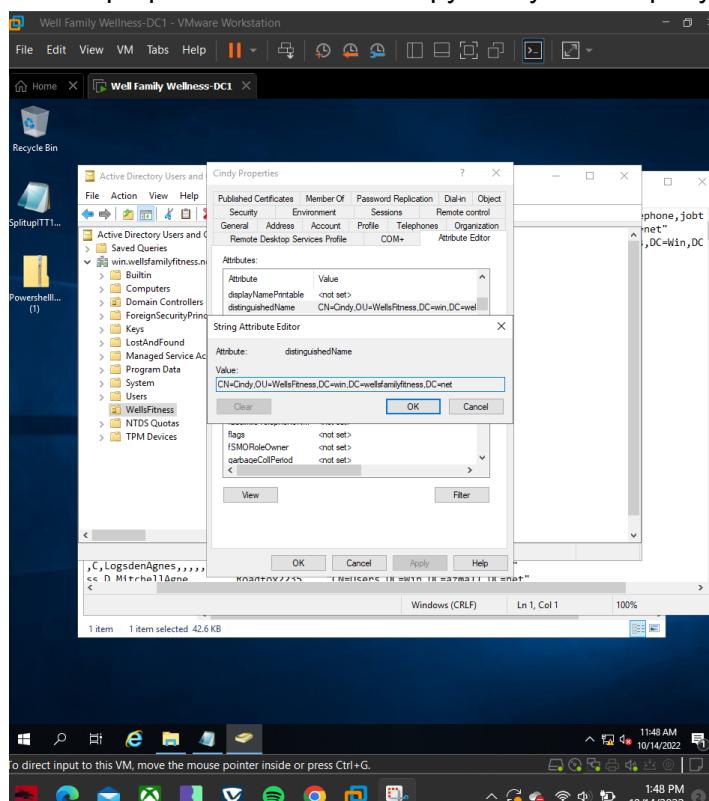
12.

13. To import the users needed, copy the text file into VMware then move this txt file into a folder in local disk C:\ temp

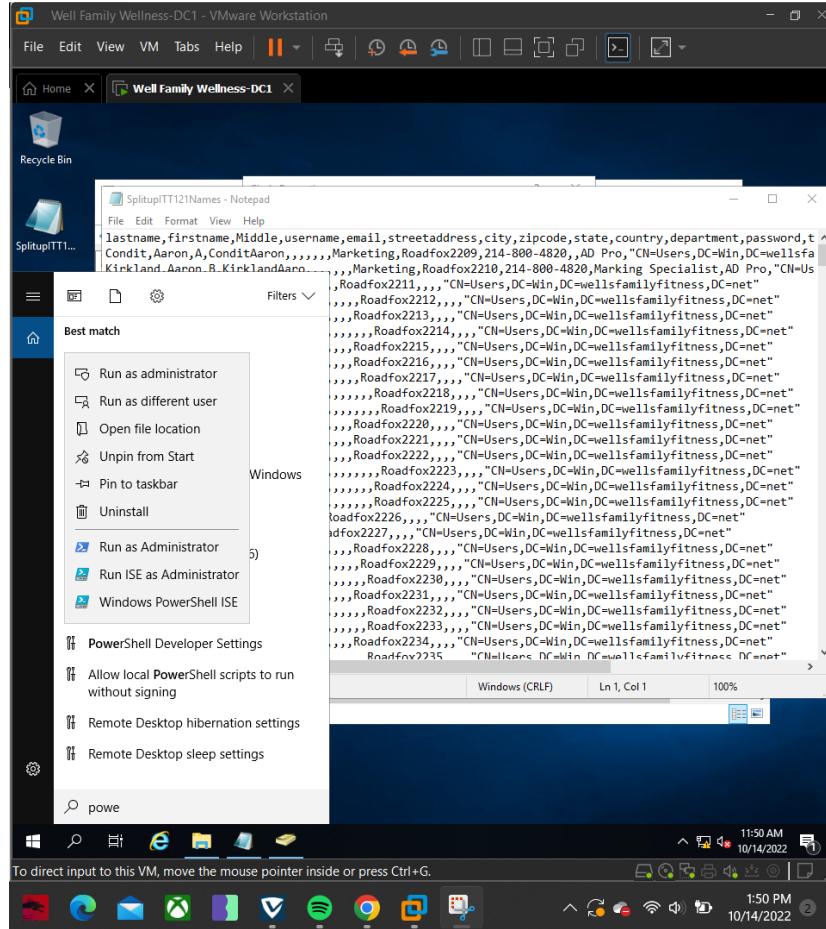


a.

14. Search properties of a user to copy DC=your company's domain name

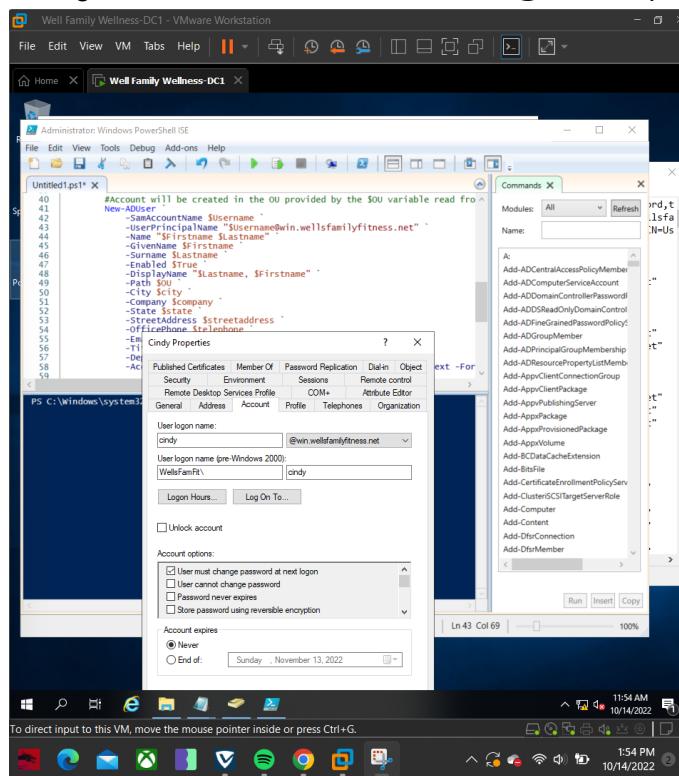


15. Replace DC=azmall with your company's domain name

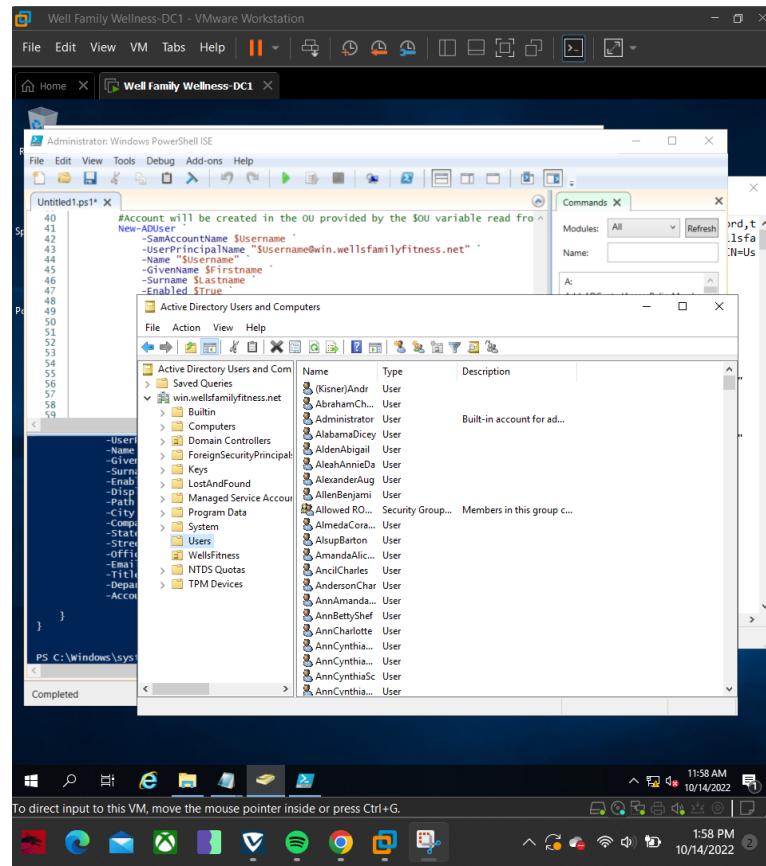


16. after replacing the domain names open Powershell ISE as administrator then copy/paste the PowerShell import file's code

17. Change first name to username, insert @win.companyname.net



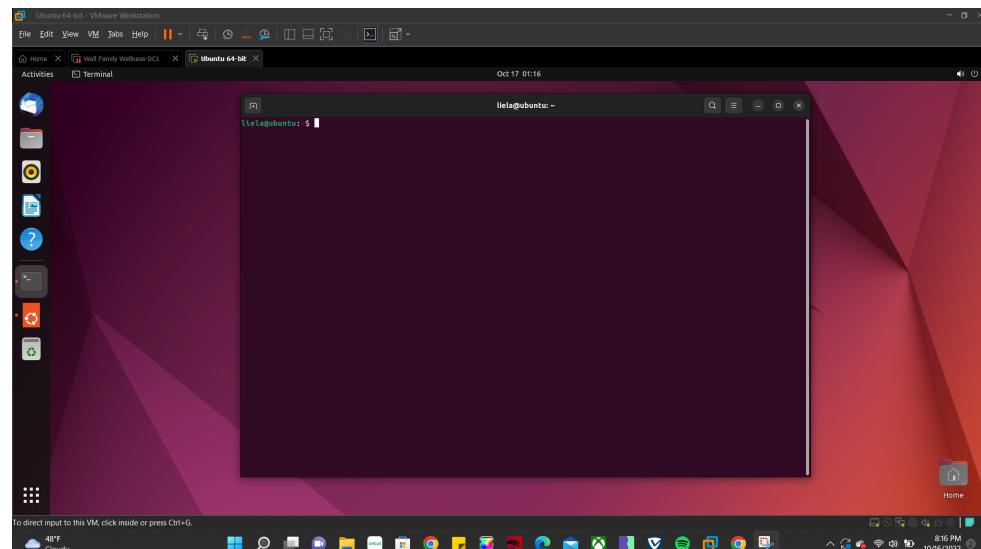
18. After making those changes, run the program and the users will import into the Active Directory



a.

UBUNTU

1. Open your desktop on linux and go to the command line



a.

2. To add a user type `sudo add user Julia`, it will prompt for your password to verify that you are the authorized user to make this change and then ask to input an appropriate password for the new user along with additional information about them.

```

lileabuntu:~ $ sudo adduser zach
[sudo] password for lilea:
Adding user `zach' ...
Adding new group `zach' (1081) with group `zach' ...
Adding user `zach' to group `zach'
Creating home directory `/home/zach' ...
Copying files from `/etc/skel' ...
New password:
Re-type new password:
The password fails the dictionary check - it is based on a dictionary word
password: password updated successfully!
Changing password for user zach
Enter the new value, or press ENTER for the default
[ ]:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [y/n] y
lileabuntu:~

```

a.

3. To add a group type `sudo addgroup groupname`

```

Done.
lileabuntu:~ $ sudo addgroup payroll
Adding group `payroll' (GID 101) ...
Done.
lileabuntu:~ $ sudo addgroup management
Adding group `management' (GID 1014) ...
Done.
lileabuntu:~ $ sudo addgroup lt
Adding group `lt' (GID 1015) ...
Done.
lileabuntu:~ $ sudo addgroup hr
Adding group `hr' (GID 1016) ...
Done.
lileabuntu:~ $ sudo addgroup healthcare
Adding group `healthcare' (GID 1017) ...
Done.
lileabuntu:~ $ sudo addgroup fitness and nutrition
addgroup: Only one of two names allowed.
lileabuntu:~ $ sudo addgroup fitnessnutrition
Adding group `fitnessnutrition' (GID 1018) ...
Done.
lileabuntu:~ $ sudo addgroup customer service
addgroup: The user `customer' does not exist.
lileabuntu:~ $ sudo addgroup customerservice
Adding group `customerservice' (GID 1019) ...
Done.
lileabuntu:~ $ sudo addgroup accounting
Adding group `accounting' (GID 1020) ...
Done.
lileabuntu:~ $ sudo addgroup administration
Adding group `administration' (GID 1026) ...
Done.
lileabuntu:~ $ sudo addgroup
lileabuntu:~ $ administration
lileabuntu:~ $ administration: command not found
lileabuntu:~ $ sudo addgroup administration
Adding group `administration' (GID 1026) ...
Done.
lileabuntu:~ $ sudo addgroup accounting
Adding group `accounting' (GID 1021) ...
Done.
lileabuntu:~ $ sudo usermod -G warehouse zach
lileabuntu:~ $ sudo usermod -G payroll sally
lileabuntu:~ $ sudo usermod -G management renée
usermod: user `renée' does not exist
lileabuntu:~ $ sudo usermod -G management renee
lileabuntu:~ $ sudo usermod -G payroll sally
lileabuntu:~ $ sudo usermod -G hr polly
lileabuntu:~ $ sudo usermod -G healthcare krisen
lileabuntu:~ $ sudo usermod -G fitnessnutrition tracy
lileabuntu:~ $ sudo usermod -G customerservice jacob
lileabuntu:~ $ sudo usermod -G administration iela
lileabuntu:~ $ sudo usermod -G accounting linda
lileabuntu:~ $

```

a.

4. To add a user to a specific group type `sudo usermod -a -G groupname username`

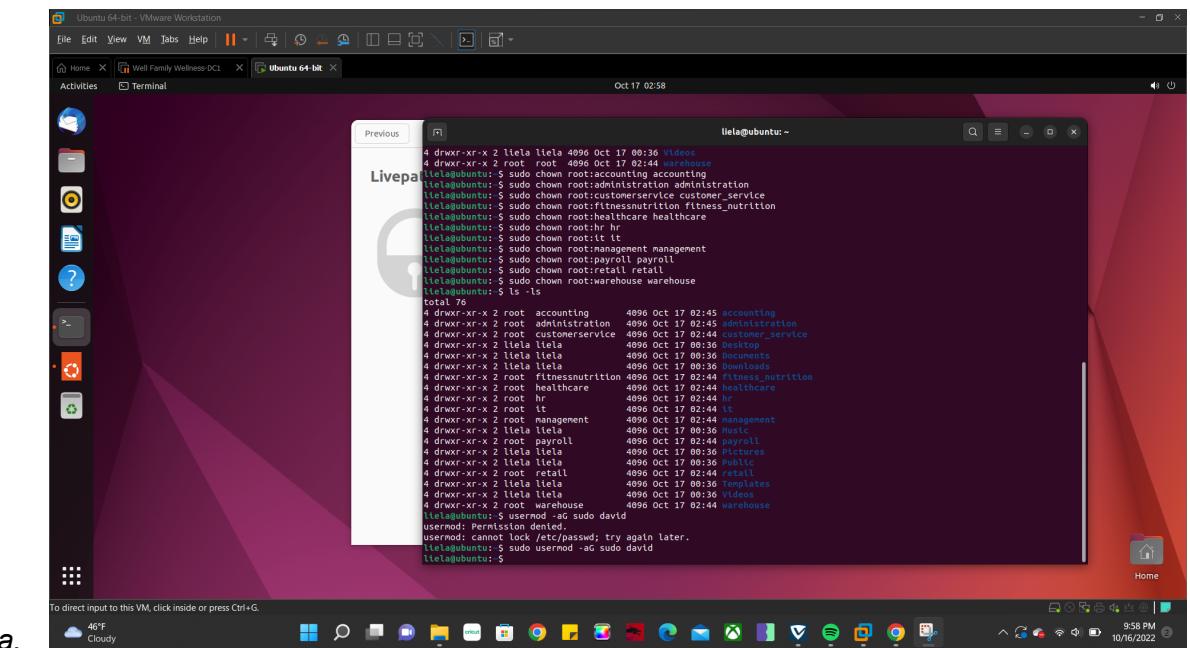
```

lileabuntu:~ $ sudo addgroup hr
Adding group `hr' (GID 1016) ...
Done.
lileabuntu:~ $ sudo addgroup healthcare
Adding group `healthcare' (GID 1017) ...
Done.
lileabuntu:~ $ sudo addgroup fitness and nutrition
addgroup: Only one of two names allowed.
lileabuntu:~ $ sudo addgroup fitnessnutrition
Adding group `fitnessnutrition' (GID 1018) ...
Done.
lileabuntu:~ $ sudo addgroup customer service
addgroup: The user `customer' does not exist.
lileabuntu:~ $ sudo addgroup customerservice
Adding group `customerservice' (GID 1019) ...
Done.
lileabuntu:~ $ sudo addgroup accounting
Adding group `accounting' (GID 1021) ...
Done.
lileabuntu:~ $ sudo usermod -G warehouse zach
lileabuntu:~ $ sudo usermod -G payroll sally
lileabuntu:~ $ sudo usermod -G management renée
usermod: user `renée' does not exist
lileabuntu:~ $ sudo usermod -G management renee
lileabuntu:~ $ sudo usermod -G payroll sally
lileabuntu:~ $ sudo usermod -G hr polly
lileabuntu:~ $ sudo usermod -G healthcare krisen
lileabuntu:~ $ sudo usermod -G fitnessnutrition tracy
lileabuntu:~ $ sudo usermod -G customerservice jacob
lileabuntu:~ $ sudo usermod -G administration iela
lileabuntu:~ $ sudo usermod -G accounting linda
lileabuntu:~ $

```

a.

5. To give an alternate user superuse permission type in this command `sudo usermod -aG sudo username`



The screenshot shows a desktop environment with a terminal window open. The terminal window title is "Livepatch" and the prompt is "lilela@ubuntu:~". The terminal displays a command history where the user has run several "sudo chown" commands to change file ownership from root to various users like accounting, customer_service, management, etc. At the bottom of the terminal, the user runs the command "sudo usermod -aG sudo david". The terminal window is part of a larger desktop interface with a dock at the bottom containing various application icons.

```
lilela@ubuntu:~$ ls -l
lilela@ubuntu:~$ sudo chown root:accounting accounting
lilela@ubuntu:~$ sudo chown root:customer_service customer_service
lilela@ubuntu:~$ sudo chown root:fitness_nutrition fitness_nutrition
lilela@ubuntu:~$ sudo chown root:healthcare healthcare
lilela@ubuntu:~$ sudo chown root:hr hr
lilela@ubuntu:~$ sudo chown root:it it
lilela@ubuntu:~$ sudo chown root:management management
lilela@ubuntu:~$ sudo chown root:payroll payroll
lilela@ubuntu:~$ sudo chown root:retail retail
lilela@ubuntu:~$ sudo chown root:warehouse warehouse
lilela@ubuntu:~$ ls -l
lilela@ubuntu:~$ sudo usermod -aG sudo david
usermod: cannot lock /etc/passwd: try again latter.
lilela@ubuntu:~$ sudo usermod -aG sudo david
lilela@ubuntu:~$
```

a.