# IS 531 – Group Project 2 – PowerShell Administration Script

For this group project, you will use your PowerShell knowledge to create two scripts that 1) create Active Directory user accounts and 2) gather information about Windows computers in your enterprise.

You will be graded on the comprehensiveness of the scripts, the ease of understanding your scripts, and the number and quality of features included (though this should be balanced with the value of the information from those features), etc.

Each group member must write approximately 1/N of the total script where N is the number of individuals in your group. In other words, you **should not** delegate the group project to a limited number of individuals in your group! [Even though the scripting portion of the exam will be open note, your upcoming exam will *expose* any member that does not know how to do the things in this project].

# Script 1 – Creating Loads of Active Directory Users

The following are guidelines as you create this script:

* You should create a script that creates Active Directory accounts from information in a csv file.
  + This CSV file will have first names, last names, and usernames in it
  + You should use the CSV file named GroupProject2Usernames.csv that can be found in LearningSuite.
  + Hint: You can loop through each row contained in the file returned from the CSV import using the foreach command, e.g. “foreach ($user in $usersfile)”.
* Your script should be able to be run from the command prompt and should have a parameter so that the CSV file (name) can be passed to it
* You should have a function that creates an Active Directory user and sets the first and last name
* After user accounts are created, you should output the SAMAccountName and SID to a CSV file
* You should pass the output file name and whether or not the output file should be created as parameters to the script
* You should run your script on an AWS Windows Server 2019 with Active Directory installed so that user accounts can actually be created. You should develop your script on the same machine that you have active directory installed on.
  + ***The appendix in this document shows how to install Active Directory.***
  + ***The appendix in this document show how to start Windows Server in AWS***
* Make sure that you comment your script.

# Script 2 – Gathering Information from Computers in the Enterprise

The following are guidelines as you create this script:

* You should primarily use WMI classes to gather information.
  + Quick teaching moment here: Windows has been around for decades, and if you look at the PowerPoint I uploaded in a previous class as optional reading called PowerShell Day 3, it shows how there is something called the WMI Architecture which shows you can access many of the objects in the windows operating system, regardless of when/how they were created over the years, through the WMI Scripting API. Do a Google search and briefly learn about WMI classes and accessing WMI using PowerShell.
  + Get-Wmi-Object is a cmdlet that provides access to WMI classes. One command to list the processes on the local computer is: “Get-WmiObject –Class Win32\_process”. If you were getting the processes listed on the local computer, you would type: “Get-WmiObject –Class Win32\_process -ComputerName .” (notice the period to mean ‘this’ computer. If you wanted to ask the same questions about a neighboring computer, and you knew its local IP address (10.0.0.3 or 172.31.84.233), you could type this command   
    “Get-WmiObject –Class Win32\_process -ComputerName 10.0.0.3”
* Your script should be as comprehensive as possible.
* Example information may include BIOS information, programs installed on the computer, current programs running, accounts that exist on the computer, etc.
* While the script by default should run on the machine where it is executed, it should also be able to be executed for a remote machine (assuming you have rights).
  + Note: We used to do this project in the lab, where you could run a script against against all the lab computers. I haven’t taken the time to figure out how to do this in AWS (get two Windows instances to talk to each other in the same network). Nevertheless, code your script to allow this possibility as if the networking was working and you could issue a command to a neighboring computer such as: “Get-WmiObject -Class Win32\_Service -ComputerName 10.1.4.62”
* Your script should be able to be called from a PowerShell console and appropriate parameters passed to it.
* Feel free to use the internet to find potential solutions; however, you should not copy scripts exactly as you find them online. You should write them yourself but can be guided by what you find.
* Make sure that you comment your script.
* Your script should save the output of the information that it gathers to a file. You can determine the type and organization of the file.

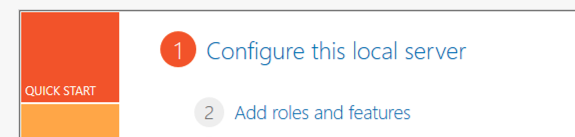
# What to turn in

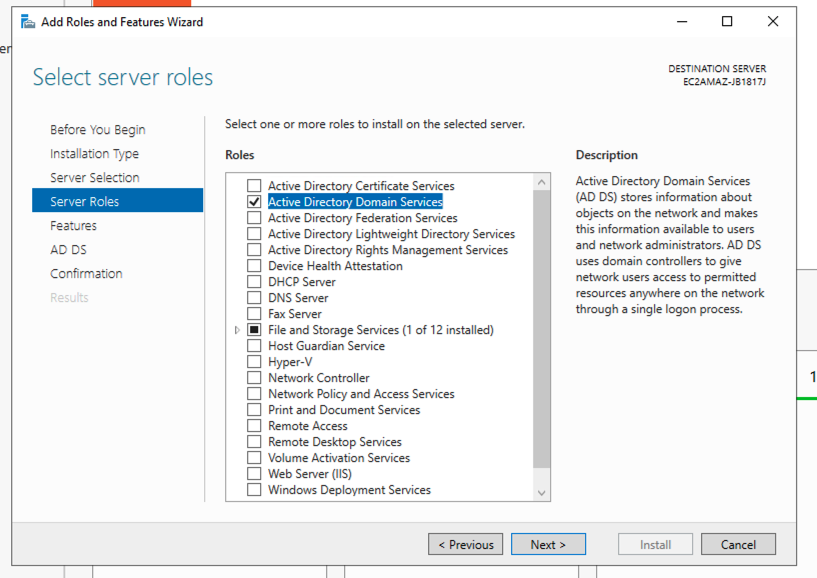
Submit a zip file with the following (using the Group Project 2 link on learningsuite.byu.edu):

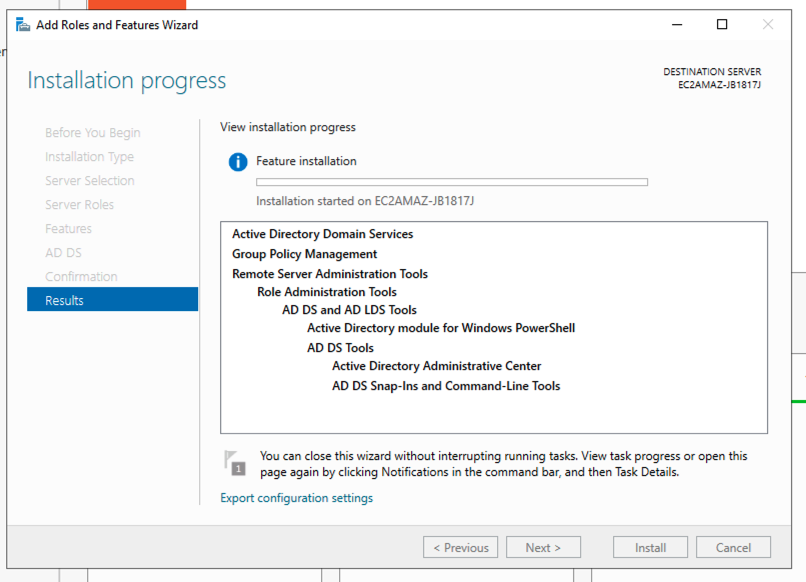
* Script 1 (with comments)
* Script 2 (with comments)
* A document that contains
  + Lessons learned
  + Output from Script 1 (after running it on your Server)
    - This could be either a screenshot of your csv file or the PowerShell screen after you run the script
  + Sample output from Script 2
    - Can be screenshots if you prefer
* A 1-page tutorial highlighting something interesting you discovered while doing this project that you would like to share with your peers

# Appendix 1 – Set up Active Directory

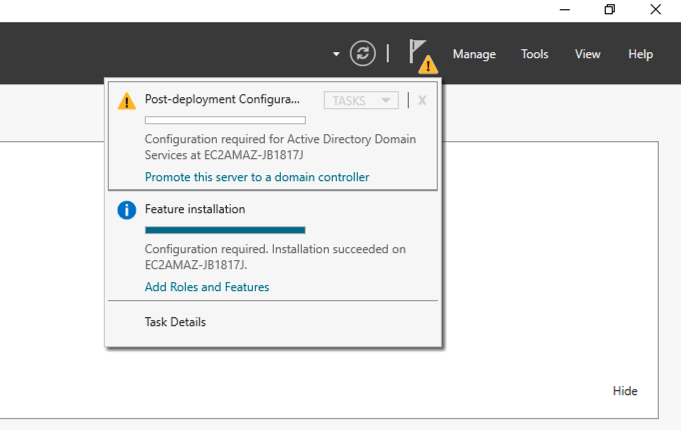
1. Open Server Manager from the Start menu
2. In the quickstart menu at the top of the page, click on “Add roles and features”



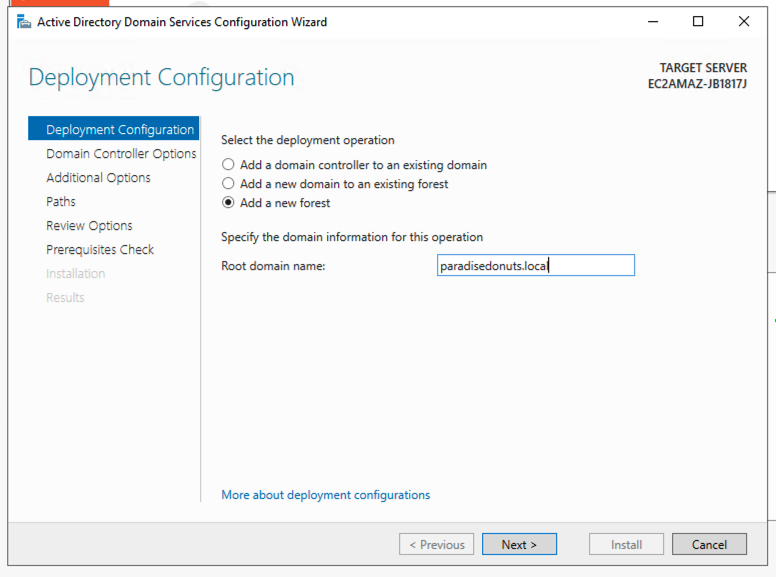
1. Click Next on the first page, leave the default “Role-based or feature-based installation” on the second page and click Next, leave the default on the Server Selection page, on the fourth page (Server Roles) select Active Directory Domain Services and confirm. Click Next.  
     
   
2. On the Features page, click Next, continue accepting defaults and then click install when prompted. You will see a progress window like this one:



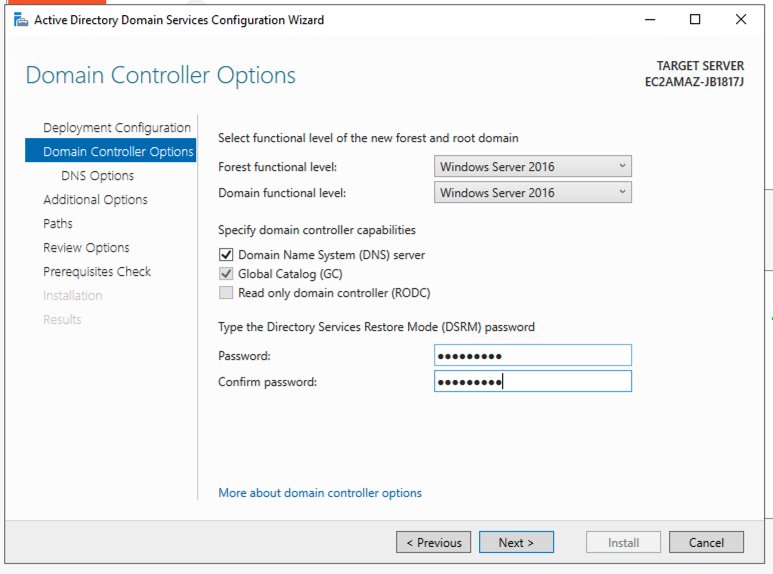
1. In the upper right of Server Manager, you will see a yellow information icon. Click on it, and it will prompt you to configure your Active Directory installation. Click on the link that says “Promote this server to a domain controller”



1. In the Active Directory Domain Services Configuration Wizard that pops up, select “Add a new forest” and specify a root directory using the suffix “.local”. For example, paradisedonuts.local. Click Next.



1. Enter a Directory Services Restore Mode (DSRM) password. Click Next.



1. Ignore the warning message on the DNS Options, page, click Next.
2. Accept all the other defaults (ignore error messages) and then click Install. After it installs, the server will reboot.

# Appendix 2 – Launch a Windows Server Instance

1. Select 2019 Base (free tier)

Graphical user interface, text, application, email

Description automatically generated

1. Make sure the security group supports 3389 so you can connect via remote desktop

Graphical user interface, text, application, email

Description automatically generated

1. Click on the Connect button

Graphical user interface, text, application

Description automatically generated

1. Wait 4 minutes for the password to be available.
2. On the RDP client tab, click the Get password link

Graphical user interface, text, application, email

Description automatically generated

A picture containing text

Description automatically generated

1. Click on the Browse button to find your key pair for this server.

Graphical user interface, application

Description automatically generated

1. Click Decrypt Password.

A picture containing shape

Description automatically generated

1. You now have a user (Administrator), a URL, and a Password to long in with.
2. Log in with RDP (remote desktop protocol) on Windows or Microsoft Remote Desktop App on Mac.
3. You can find Server Manager from the Start Menu to install things

A screenshot of a computer

Description automatically generated with medium confidence