AUN1 — AUN1 TASK 2: SERVER ROLES AND FEATURES

SCRIPTING AND AUTOMATION — D411
PRFA — AUN1

Preparation Task Overview Submissions Evaluation Report

COMPETENCIES

4049.2.2 : Writes Automations Scripts

The learner writes scripts that automate configuration tasks.

INTRODUCTION

In this task, you will create two PowerShell scripts. PowerShell enables administrators to perform administrative tasks on both local and remote systems. You will be expected to manage an Active Directory and SQL Server within the PowerShell environment. This management will include the configuration and administration of the servers.

For this task, you will use the "Performance Assessment Lab Area" web link to access the virtual lab environment to complete this task.

SCENARIO

You have been hired as a consultant at a company. The company previously had an SQL server and Active Directory server configured throughout two separate Windows 2012 servers. However, all the drives (including backups) were destroyed due to unforeseen circumstances, and you need to write PowerShell scripts that can accomplish all the required tasks from the local server.

REQUIREMENTS

Your submission must represent your original work and understanding of the course material. Most performance assessment submissions are automatically scanned through the WGU similarity checker. Students are strongly encouraged to wait for the similarity report to generate after uploading their work and then review it to ensure Academic Authenticity guidelines are met before submitting the file for evaluation. See Understanding Similarity Reports for more information.

Grammarly Note:
Professional Communication will be automatically assessed through Grammarly for Education in most performance assessments before a student submits work for evaluation. Students are strongly encouraged to review the Grammarly for Education feedback prior to submitting work for evaluation, as the overall submission will not pass without this aspect passing. See Use Grammarly for Education Effectively for mo

Help

information.

Microsoft Files Note:
Write your paper in Microsoft Word (.doc or .docx) unless another Microsoft product, or pdf, is specified in the task directions. Tasks may not be submitted as cloud links, such as links to Google Docs, Google Slides, OneDrive, etc. All supporting documentation, such as screenshots and proof of experience, should be collected in a pdf file and submitted separately from the main file. For more information, please see Computer System and Technology Requirements.

You must use the rubric to direct the creation of your submission because it provides detailed criteria that will be used to evaluate your work. Each requirement below may be evaluated by more than one rubric aspect. The rubric aspect titles may contain hyperlinks to relevant portions of the course.

A. Create a PowerShell script named "Restore-AD.ps1" within the attached "Requirements2" folder. Create a comment block and include your first and last name along with your student ID.

B. Write the PowerShell commands in "Restore-AD.ps1" that perform all the following functions without user interaction:
   1. Check for the existence of an Active Directory Organizational Unit (OU) named "Finance." Output a message to the console that indicates if the OU exists or if it does not. If it already exists, delete it and output a message to the console that it was deleted.
   2. Create an OU named "Finance." Output a message to the console that it was created.
   3. Import the financePersonnel.csv file (found in the attached "Requirements2" directory) into your Active Directory domain and directly into the finance OU. Be sure to include the following properties:
      - First Name
      - Last Name
      - Display Name (First Name + Last Name, including a space between)
      - Postal Code
      - Office Phone
      - Mobile Phone
   4. Include this line at the end of your script to generate an output file for submission:
```
Get-ADUser -Filter * -SearchBase "ou=Finance,dc=consultingfirm,dc=com"
 -Properties DisplayName,PostalCode,OfficePhone,MobilePhone >
 .\AdResults.txt
```

C. Create a PowerShell script named "Restore-SQL.ps1" within the attached "Requirements2" folder. Create a comment block and include your first and last name along with your student ID.

D. Write the PowerShell commands in a script named "Restore-SQL.ps1" that perform the following functions without user interaction:
   1. Check for the existence of a database named ClientDB. Output a message to the console that indicates if the database exists or if it does not. If it already exists, delete it and output a message to the console that it was deleted.
   2. Create a new database named "ClientDB" on the Microsoft SQL server instance. Output a message to the console that the database was created.
   3. Create a new table and name it "Client_A_Contacts" in your new database. Output a message to the console that the table was created.

4. Insert the data (all rows and columns) from the “NewClientData.csv” file (found in the attached “Requirements2” folder) into the table created in part D3.
5. Include this line at the end of your script to generate the output file SqlResults.txt for submission:

```
Invoke-Sqlcmd -Database ClientDB –ServerInstance .\SQLEXPRESS -Query
  ‘SELECT * FROM dbo.Client_A_Contacts’ > .\SqlResults.txt
```

E. Apply exception handling using try-catch. Output any error messages to the console.

F. Run your Restore-AD.ps1 script from this console and take a screenshot of the output.
   1. Run your Restore-SQL.ps1 script from this console and take a screenshot of the output.

G. Compress the “Requirements2” folder as a ZIP archive. When you are ready to submit your final task, run the Get-FileHash cmdlet against the “Requirements2” ZIP archive. Note the hash value and place it into the comment section when you submit your task.
   1. Include *all* the following files intact within the “Requirements2” folder, including the original files and any additional files you created to support your script:
      i. “Restore-AD.ps1”
      ii. “Restore-SQL.ps1”
      iii. “AdResults.txt”
      iv. “SqlResults.txt”
      v. Screenshots from Parts F and F1

H. Apply scripting standards throughout your script, including the addition of comments that describe the behavior of the script.

File Restrictions

File name may contain only letters, numbers, spaces, and these symbols: ! - _ . * ' ( )
File size limit: 200 MB
File types allowed: doc, docx, rtf, xls, xlsx, ppt, pptx, odt, pdf, csv, txt, qt, mov, mpg, avi, mp3, wav, mp4, wma, flv, asf, mpeg, wmv, m4v, svg, tif, tiff, jpeg, jpg, gif, png, zip, rar, tar, 7z

RUBRIC

A:SCRIPT AD CREATION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The response does not include a script named “Restore-AD.ps1” and does not include student name and ID.	The PowerShell script is not created within the “Requirements2” folder, or it is incorrectly named, or student name and ID are not included as a comment block.	The PowerShell script is created within the “Requirements2” folder and includes a comment block that includes the first and last name and student ID number.

B1:ACTIVE DIRECTORY FINANCE OU OUTPUT MESSAGE

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The script does not check for the existence of a Active Directory OU named "Finance."	The script does not check for the existence of an Active Directory Organizational Unit (OU) named "Finance" or does not output a message of its existence. If the OU exists, the script does not delete the OU or does not confirm the deletion with an output message.	The script checks for the existence of an Active Directory Organizational Unit (OU) named "Finance" and outputs a message of its existence. If the OU exists, the script deletes the OU and confirms the deletion with an output message.

B2:ACTIVE DIRECTORY FINANCE OU

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
An OU is not created.	The script successfully creates an OU but it is not named "Finance" or a message is not output to the console that it was created.	The script successfully creates an OU named "Finance" and a message is output to the console that it was created.

B3:DATA INSERTION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The script logic does not import *any* file.	The script imports the correct file into the wrong OU, or the script logic does not import the correct file, or the script does not import all the data.	The script imports the correct file including all rows and attributes into the correct OU.

B4:VERIFY AD OBJECT CREATION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The ADResults.txt file is not submitted, or it is blank.	The ADResults.txt file includes some but not *all* of the rows and attributes in the OU.	The ADResults.txt file includes *all* of the rows and attributes in the OU.

C:SCRIPT SQL CREATION

NOT EVIDENT

The response does not include a script named “Restore-SQL.ps1” and does not include student name and ID.

APPROACHING COMPETENCE

The PowerShell script is not created within the “Requirements2” folder, or it is incorrectly named, or student name and ID are not included as a comment block.

COMPETENT

The PowerShell script is created within the “Requirements2” folder and includes a comment block that includes the first and last name and student ID number.

D1:DATABASE NAMED CLIENTDB OUTPUT MESSAGE

NOT EVIDENT

The script does not check for the existence of a database named ClientDB.

APPROACHING COMPETENCE

The script does not check for the existence of a database named ClientDB or does not output a message of its existence. If the database exists, the script does not delete the database or does not confirm the deletion with an output message.

COMPETENT

The script checks for the existence of a database named ClientDB and outputs a message of its existence. If the database exists, the script deletes the database and confirms the deletion with an output message.

D2:NEW DATABASE

NOT EVIDENT

The script logic does not create a database.

APPROACHING COMPETENCE

The script logic partially creates a database, or the solution is not on the SQL server named “ClientDB.”

COMPETENT

The script logic correctly creates a new database on the SQL server named “ClientDB.”

D3:NEW TABLE

NOT EVIDENT

The script logic does not create a table.

APPROACHING COMPETENCE

The script logic creates a new table that does not include each of the required attributes, or the new table is not named “Client_A_Contacts,” or the table

COMPETENT

The script logic correctly creates a new table named “Client_A_Contacts” and the table is added to the new database.

is not added to the new database.

D4:DATA INSERTION

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The script logic does not insert *any* data.	The script logic inserts the correct data into the wrong location or inserts data from the wrong file to the correct location, or the script logic does not insert all the data.	The script logic inserts all the correct data from the “NewClientData.csv” file into the "Client_A_Contacts" SQL database.

D5:VERIFY SQL DATABASE AND TABLE CREATIOND1. CMDLETS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The SqlResults.txt file is not submitted, or it is blank. The cmdlets has not been run.	The SqlResults.txt file includes some but not *all* of the rows and attributes in the database table. The cmdlets is run incorrectly.	The SqlResults.txt file includes *all* the rows and attributes in the database table. The cmdlets is run correctly.

E:EXCEPTION HANDLING

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The script does not apply *any* exception handling.	The exception handling does not cover the appropriate part of the script or the error message is either missing or does not provide relevant details of the exception.	The exception handling covers the appropriate part of the script and the error message with relevant details of the exception is provided.

F:RESTORE-AD.PS1 SCRIPT EXECUTION RESULTS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
Screenshots are not included for the output.	Screenshots are provided for some but not *all* of the actions in the script. Or the screenshots do	Accurate screenshots are provided for *each* action in the script.

	not show appropriate results from the script.	

F1:RESTORE-SQL.PS1 SCRIPT EXECUTION RESULTS

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
Screenshots are not included for the output.	Screenshots are provided for some but not *all* of the actions in the script. Or the screenshots do not show appropriate results from the script.	Accurate screenshots are provided for *each* action in the script.

G:POWERSHELL FILE HASH

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
A calculated hash value is not provided.	A calculated hash value is not consistent with the provided hash value.	A hash value is provided and evidences integrity of the zipped file.

G1:INCLUDED FILES

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
No files are included within the “Requirements2” folder.	Some but not *all* files are included within the “Requirements2” folder.	*All* files are included within the “Requirements2” folder.

H:SCRIPTING STANDARD

NOT EVIDENT	APPROACHING COMPETENCE	COMPETENT
The script does not apply any scripting standards.	The script contains errors in scripting standards, or comments are not added throughout the script.	The script accurately applies scripting standards throughout the script, including added comments that describe the behavior of the script.

WEB LINKS

Performance Assessment Lab Area

Skillable Labs Knowledge Base Article
Please consult this WGU Knowledge Base article for general FAQs regarding your Skillable lab environment.

SUPPORTING DOCUMENTS

Requirements2.zip