Advanced Scripting   
Remoting SSH

Last Updated: 6/29/2021 1:42 PM Version 1  
Document Prepared for: CIT361 Student

# Name Zach Lucas ID 895649438

# Instructions

Save a copy of this document. Answer all questions directly in this document. You will save and upload this completed document as your homework submission.

# Overview

PowerShell can use SSH for remoting as well. This is necessary since Linux does not support WMI which is the technology that PowerShell uses natively for remoting. You will need ssh installed. Open SSH is now included with Windows (Win 10 1004 update, Server 2019)

# Requirements

* SSH Client.
* Internet access.

# Setup

SSH to the lab:  
Computer: **cit361-lab.citwdd.net**Port: **443**  
User: Your mailbox portion of your BYU-I email address. If your BYU-I email is [lin87690@byui.edu](mailto:lin87690@byui.edu) you would use **lin87690** for your username  
Password: Your I-Number (If you changed your password earlier in the semester you will need to use that one)

# Task 1—Creating a Session via SSH

Creating SSH sessions is very similar. The main difference is you use the -HostName parameter for the machine to connect to, and the -UserName parameter for the user you want to connect as.

## Steps

1. Connect to the lab computer, then Open **PowerShellCore**.
2. Create a SSH session to a Linux machine  
   $s=New-PSSession -HostName eric -UserName ps
   1. Enter the password **PowerShellRocks!** When prompted.
3. Get a list of sessions.
   1. What is the transport type?SSH
   2. What is the Name? eric
4. Execute a command on the remote machine  
   invoke-command -Session $s -ScriptBlock {ls -l}
   1. What was returned? The home directory of eric
5. View the PowerShell info  
   invoke-command -Session $s -ScriptBlock {$PSVerstionTable}
   1. What version of PowerShell is running? 7.0.2
   2. What is the OS? Linux
   3. Which PowerShell Edition? Core

# Task 2—Interactive Session

You can also do interactive session with Enter-PSSession

## Steps

1. Use your existing session to connect interactively  
   Enter-PSSession $s
2. Notice the prompt now includes the username and machine
3. Explore the machine with PowerShell
4. When you are done use exit to close the interactive session.
5. NOTE: Prior to PS Version 7.0.2 you had to do interactive sessions this way. As of 7.0.2 you can use the Enter-PSSession with the -Hostname and -UserName parameters. (this is under active development at the time of this writing)

# Deliverable

Upload this document with completed answers to i-learn.