

# Collect Data from Multiple Machines for Analysis



**Liam Cleary**

Microsoft MVP and Microsoft Certified Trainer at SharePlicity

@helloitsliam | [www.helloitsliam.com](http://www.helloitsliam.com)



# Overview

## Goal: Collect Data from Machines for Analysis

- Understanding PowerShell remoting
- Using SSH for remoting to Windows and Linux
- Exporting log data from remote machines
- Create scheduled tasks using PowerShell



# Understanding PowerShell Remoting



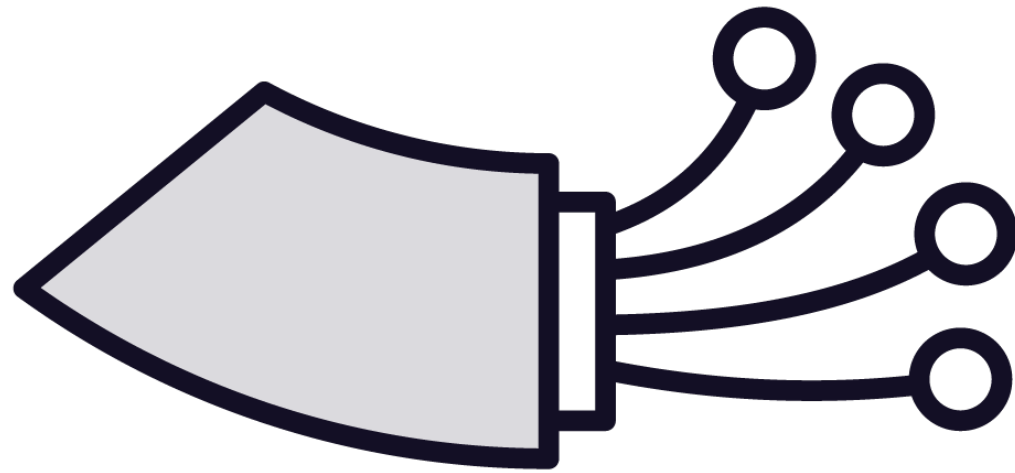


**PowerShell Remoting isn't the same as using a cmdlet's "ComputerName" parameter to run it on a remote computer**





# PowerShell Remoting



Protocol that allows users to run PowerShell commands on remote computers

PowerShell supports remote computing by using various technologies

PowerShell supports WMI, WS-Management, and SSH remoting

PowerShell 7 and above support RPC remoting only within Windows



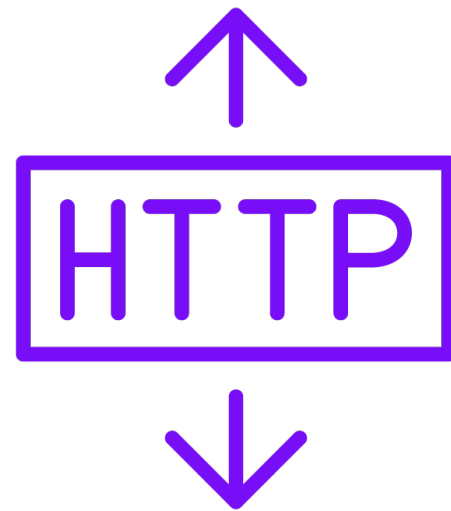
# WinRM

**Windows Remote Management (WinRM) is Microsoft's implementation of the WS-Management Protocol. WSMan is a standard Simple Object Access Protocol (SOAP) based protocol that allows hardware and operating systems, from different vendors to interoperate**



# PowerShell Remoting Connections

PowerShell Remoting is primarily designed for connecting and managing remote devices within an Active Directory environment



**HTTP**

Windows 10 default option



**HTTPS**

Requires IT to acquire, provision, and manage certificates



**SSH**

Available since PowerShell core. Allows cross-platform connecting



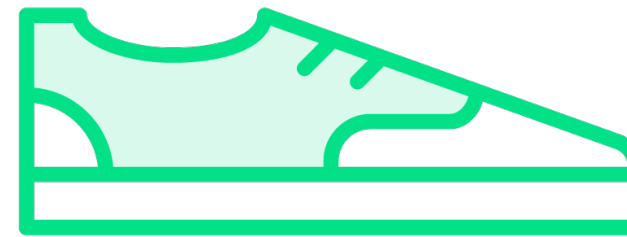
# PowerShell Remoting



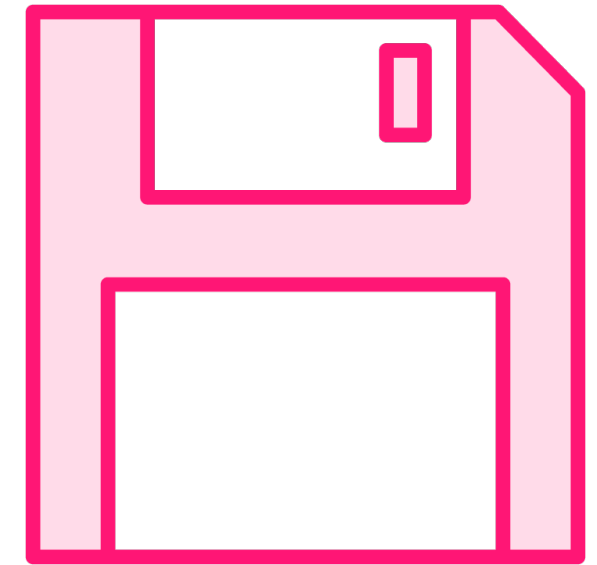
**Connect**



**Authenticate**



**Execute  
Commands**



**Return Output**



# Enable PowerShell Remoting Using WS-MAN



## To enable PowerShell remoting for non-administrators

- Define Active Directory security group
- Add group to the remote management users' group locally

## Group policy should be used for enabling remoting on multiple machines

- Use the computer configuration option
- Navigate to the Windows Remote Management (WinRM) settings
- Set the Allow option for the WinRM service



# Enable PowerShell Remoting Using WS-MAN

**# Enable over HTTP**

```
Set-WSManQuickConfig
```

**# Enable over HTTPS**

```
Set-WSManQuickConfig -UseSSL
```

**# Enable Remoting**

```
Enable-PSRemoting -Force
```

**# Enable Remoting with no Network Profile Check**

```
Enable-PSRemoting -Force -SkipNetworkProfileCheck
```





# Demo

## Enable PowerShell Remoting Using WS-MAN

- Enable remoting within Windows
- Connect to Windows server using PowerShell remoting



# Using SSH for Remoting to Windows and Linux



# PowerShell Remoting over SSH



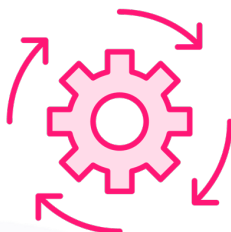
**Multiplatform PowerShell remoting**



**Basic remoting between Windows, Linux, and macOS computers**



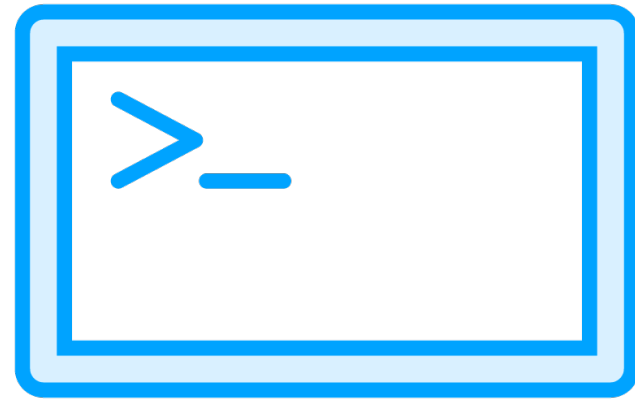
**No support for remote endpoint configuration and Just Enough Administration (JEA)**



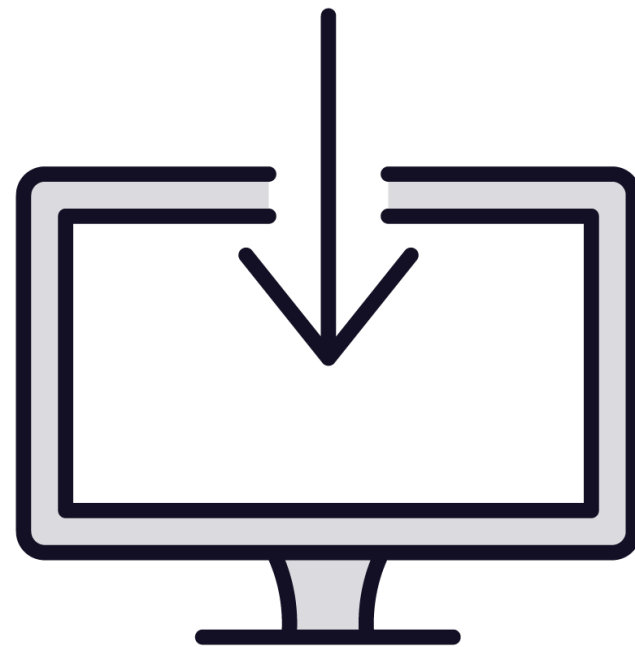
**Creates a PowerShell host process on the target computer as an SSH subsystem**



# Enable PowerShell Remoting Using SSH



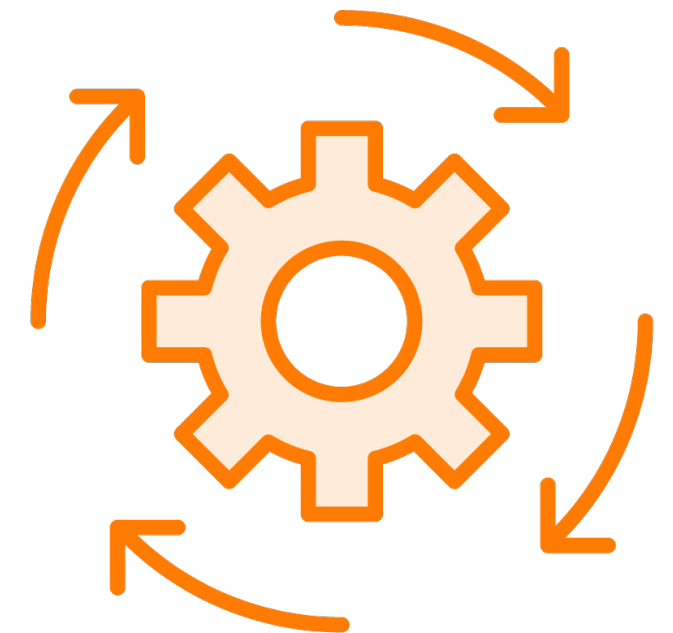
**Install Latest  
version of  
PowerShell**



**Install Latest  
OpenSSH**



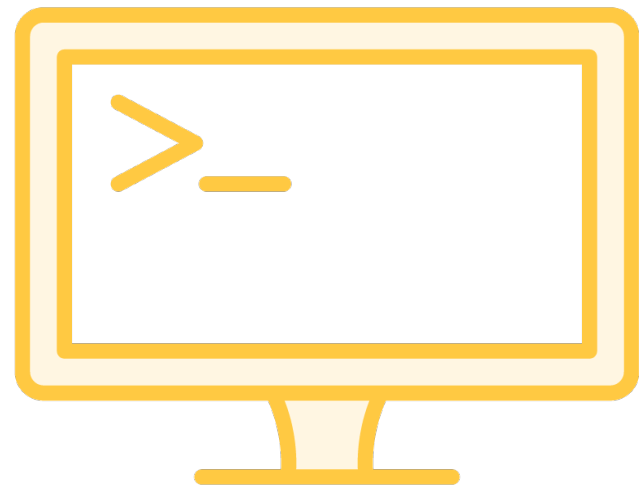
**Define the SSH  
Configuration**



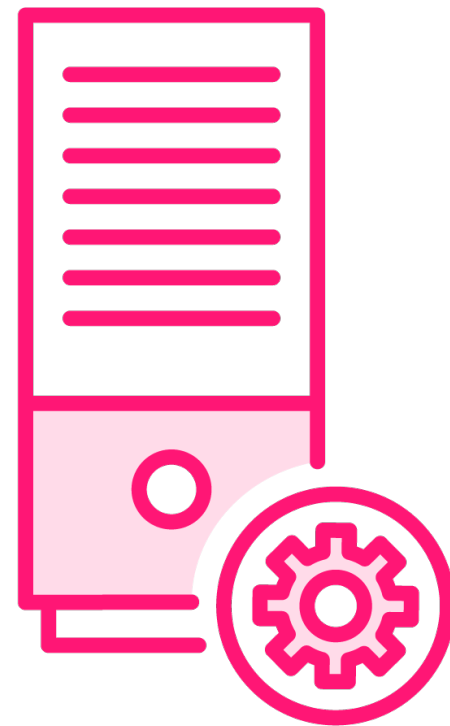
**Restart SSH  
Service**



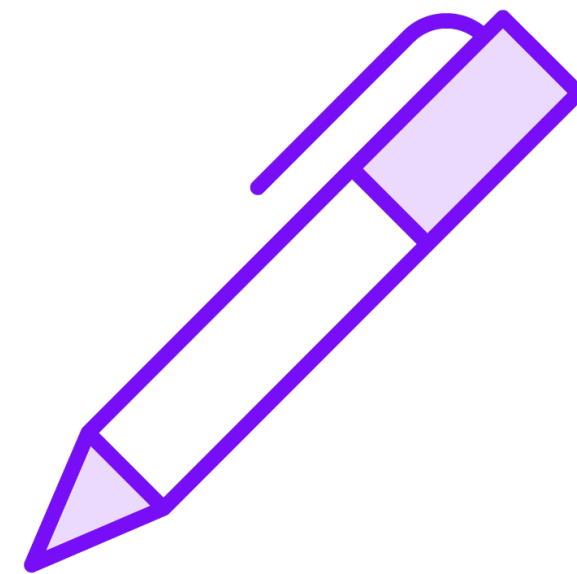
# Enable PowerShell Remoting Using SSH (Linux)



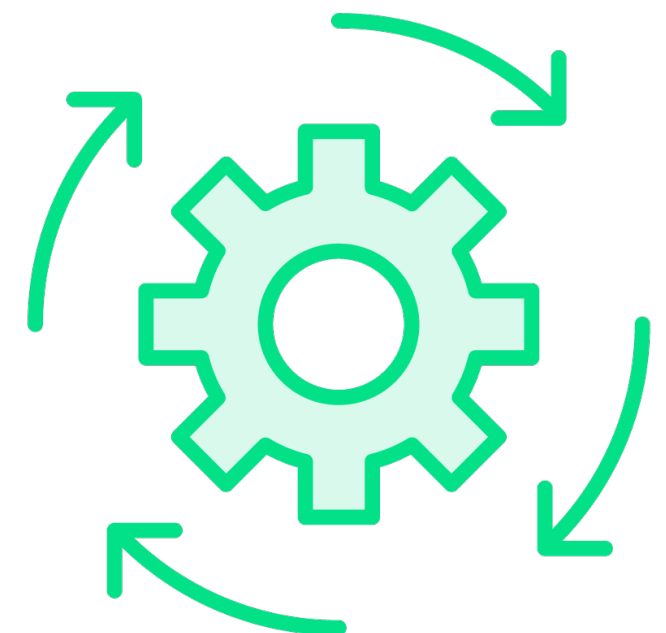
**Install Latest  
version of  
PowerShell**



**Install Latest  
OpenSSH Server**



**Define the SSH  
Configuration**



**Restart SSH  
Service**



# Prepare Windows for SSH Remoting

## # Install OpenSSH Client and Server

```
Add-WindowsCapability -Online -Name OpenSSH.Client  
Add-WindowsCapability -Online -Name OpenSSH.Server
```

## # Set the SSH Server Service to start Automatically

```
Set-Service -Name sshd -StartupType "Automatic"  
Start-Service -Name sshd
```

## # Install and Import Microsoft's PowerShell Remoting Module

```
Install-Module -Name Microsoft.PowerShell.RemotingTools  
Import-Module -Name Microsoft.PowerShell.RemotingTools
```

## # Enable SSH Remoting and Restart the Service

```
Enable-SSHRemoting -Verbose  
Restart-Service -Name sshd
```





# Prepare Linux for SSH Remoting

## # Install OpenSSH Client and Server

```
sudo yum -y install openssh-server openssh-clients
```

## # Start the SSH Server Service

```
sudo systemctl start sshd  
sudo systemctl status sshd
```

## # Enable the OpenSSH Service

```
sudo systemctl enable sshd
```

## # Modify the SSH Server Configuration and Restart

```
sudo vim /etc/ssh/sshd_config  
service sshd restart
```



# Demo

## Using SSH for Remoting to Windows and Linux

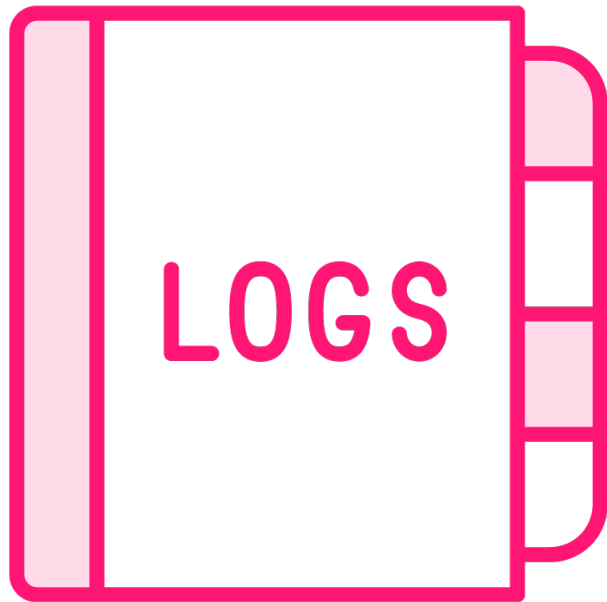
- Prepare Windows client for SSH
- Prepare Windows server for SSH
- Prepare Linux for SSH
- Connect to a Windows and Linux machine using PowerShell remoting over SSH



# Exporting Log Data from Remote Machines



# Exporting Event Logs



## Get-EventLog

Deprecated command  
Not recommended by Microsoft



## Get-WinEvent

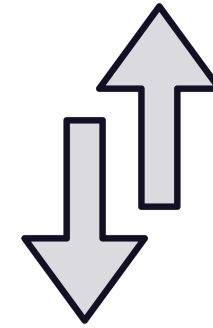
Latest command  
Microsoft recommended  
Supports modern Windows event log capabilities



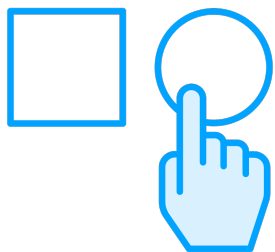
# Get-WinEvent



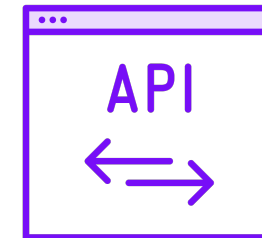
**Supports modern event logs and classic logs**



**Returns event information sorted newest to oldest**



**Retrieve entries from selected logs**



**Retrieve entries from selected log providers**



**Filter support for XPath, XML, and hash table queries**



**Only available on the Windows Platform**



# Retrieve Windows Event Log Entries

**# Retrieve All Event Logs**

```
Get-WinEvent -ListLog *
```

**# Retrieve Event Logs with Entries**

```
Get-WinEvent -ListLog * | Where-Object -Property RecordCount -GT 0
```

**# Retrieve Entries from the Security Event Log**

```
Get-WinEvent -LogName Security
```

```
Get-WinEvent -LogName Security -MaxEvents 5 | Select-Object -Property *
```





# Filtering Windows Event Log Entries

## # Retrieve Entries and Filter by ID

```
Get-WinEvent -LogName Security | Where-Object -Property Id -EQ 4624
```

## # Retrieve Entries and Filter by ID using Hash Table

```
Get-WinEvent -FilterHashtable @{ LogName='Security'; ID=4624 }
```

## # Retrieve Entries and Filter by ID using Hash Table with Variables

```
$logName = "Security"
```

```
$startDate = ""
```

```
$endDate = ""
```

```
Get-WinEvent -FilterHashtable @{
```

```
    LogName = $logName;
```

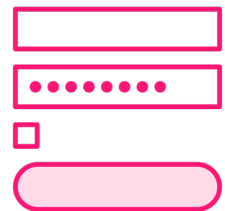
```
    StartTime = $startDate;
```

```
    EndTime = $endDate
```

```
}
```



# Linux Logs



## Authorization Log (/var/log/auth.log)

Keeps track of authorization systems, such as password prompts, the sudo command and remote logins



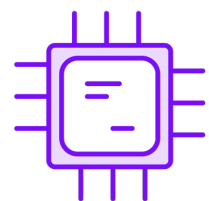
## Daemon Log (/var/log/daemon.log)

Daemons are programs that run in the background, usually without user interaction. For example, display server, SSH sessions, printing services, Bluetooth, and more



## Debug Log (/var/log/debug)

Provides debugging information from the Ubuntu system and applications



## Kernel Log (/var/log/kern.log)

Logs from the Linux kernel



## System Log (/var/log/syslog)

Contains more information about the system



# Retrieve Log Entries from Linux

## # Retrieve Logs

```
Get-ChildItem /var/log
```

## # Retrieve Entry from All Logs

```
Get-ChildItem -Path /var/log -File |  
    foreach { $_ | Get-Content | Select -First 1 -Skip 5 }
```

## # Retrieve Entries from the Auth Log

```
Get-Content /var/log/auth.log
```

## # Retrieve Entries from the Auth Log and Export to CSV

```
$path = "/home/trainer/auth.csv"  
$results = Get-Content /var/log/auth.log | Select-String "trainer"  
$results | Export-Csv -Path $path -NoTypeInfoation
```



# Demo

## Exporting Log Data from Remote Machines

- Connect to a Windows machine and export event log entries
- Connect to Linux machine and export log entries





# **Creating Scheduled Tasks Using PowerShell**



# Creating Scheduled Tasks Cmdlets



**New-ScheduledTaskAction**

**New-ScheduledTaskTrigger**

**New-ScheduledTaskPrincipal**

**New-ScheduledTaskSettingsSet**

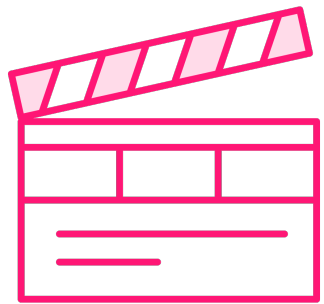
**New-ScheduledTask**

**Register-ScheduledTask**

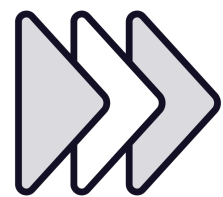




# Creating Scheduled Tasks Cmdlets



**New-ScheduledTaskAction** cmdlet creates an object that contains the definition of a scheduled task action. A scheduled task action represents a command that a task executes when Task Scheduler runs the task



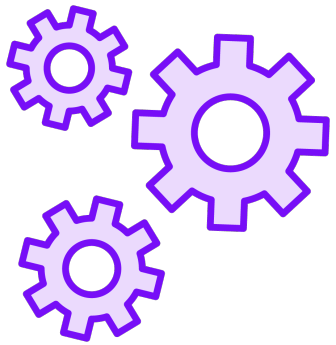
**New-ScheduledTaskTrigger** cmdlet creates and returns a new scheduled task trigger object. You can use a time-based trigger or an event-based trigger to start a task



**New-ScheduledTaskPrincipal** cmdlet creates an object that contains a scheduled task principal. Use a scheduled task principal to run a task under the security context of a specified account.



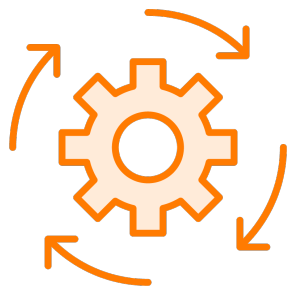
# Creating Scheduled Tasks Cmdlets



**New-ScheduledTaskSettingsSet** cmdlet creates an object that contains scheduled task settings. Each scheduled task has one set of task settings



**New-ScheduledTask** cmdlet creates an object that contains the definition of a scheduled task. Support for Win32 applications, Win16 applications, OS/2 applications, MS-DOS applications, batch files, command files, or any properly registered file type



**Register-ScheduledTask** cmdlet registers a scheduled task definition on a local computer. You can register a task to run executable files, batch files, or any registered file type



# Creating Scheduled Tasks

```
$action = (New-ScheduledTaskAction -Execute 'Query-Logs.ps1')  
$trigger = New-ScheduledTaskTrigger -Daily -At '10:00 AM'
```

```
$principal = New-ScheduledTaskPrincipal `   
    -UserId 'TRAINING\trainer' `   
    -RunLevel Highest
```

```
$settings = New-ScheduledTaskSettingsSet `   
    -RunOnlyIfNetworkAvailable `   
    -WakeToRun
```

```
$task = New-ScheduledTask `   
    -Action $action `   
    -Principal $principal `   
    -Trigger $trigger `   
    -Settings $settings
```

```
Register-ScheduledTask 'TASK: Query Remote Server Logs' -InputObject $task
```



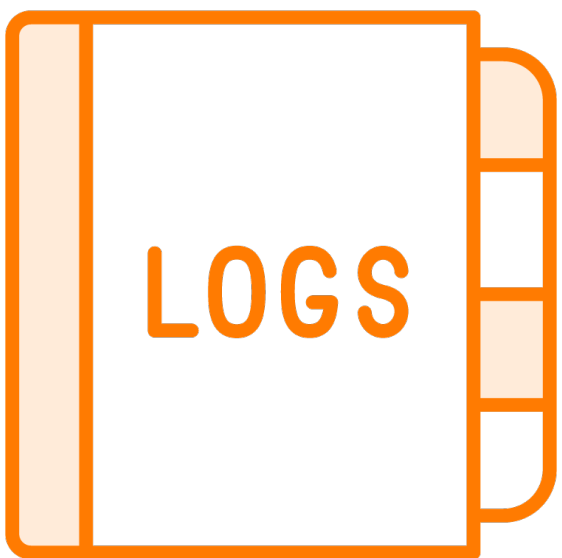
# Query-Logs.ps1



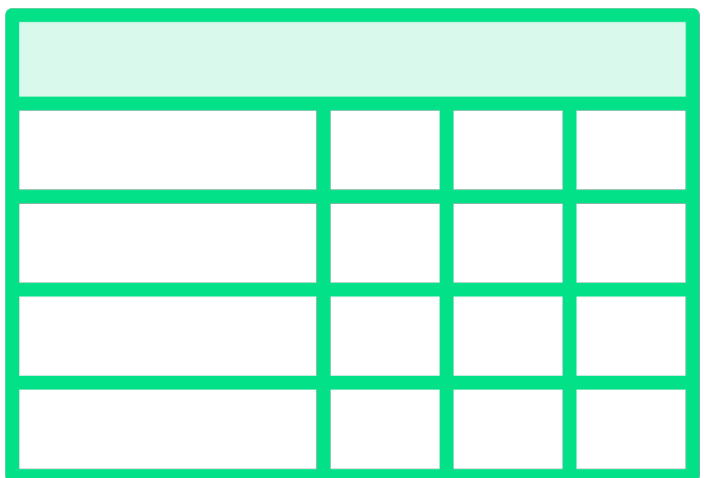
**Define variables**



**Create remote  
sessions**



**Retrieve Logs  
using remote  
sessions**



**Export as CSV  
files**



**Demo**

## **Creating Scheduled Tasks Using PowerShell**



# Summary

## Goal: Collect Data from Machines for Analysis

- Reviewed PowerShell remoting, and used WinRM for PowerShell remoting
- Remotely connected to Windows and Linux using SSH
- Created a Windows scheduled task to export log file data





**Up Next:**

# **Querying Exported Data for Process or Service Anomalies**

---

