Outline

In this lab, we'll discuss how to work with repositories, some best practices, and how to publish to the PowerShell Gallery.

Exercise 1 - Analyze Code for Errors

Before we publish our code, the first thing we'll want to do is check for any errors or issues with the code itself. Microsoft publishes a tool called PSScriptAnalyzer that allows you to do so, and it's available for download from the PSGallery.

```
Install-Module PSScriptAnalyzer
```

Once we've installed it, we can use it to check for any issues that require correction prior to publishing.

```
# Before running this, make sure your PowerShell working directory is the same
folder as the file you're currently reading!
Invoke-ScriptAnalyzer -Path $PWD -Recurse
```

We'll need to fix any errors or warnings that are found, but our module should have a clean return. Next, let's test our module manifest to make sure the required metadata is present.

```
# Before running this, make sure your PowerShell working directory is the same
folder as the file you're currently reading!
Test-ModuleManifest -Path .\PowerShellSummit2025\PowerShellSummit2025.psd1
```

Exercise 2 - Publish to the PowerShell Gallery

Now that we're sure our code is ready, we can publish our module to the PowerShell gallery. First, you'll need to log into the PSGallery using your Microsoft account, accept the terms, and then generate an API key. For the purposes of this lab, we'll use the following settings:

- Expires In: 1 Day
- Scope:
    - Push new packages and package versions
- Glob pattern: *

Copy the key to a secure location for safekeeping; we'll need it in a moment.

You'll need to pick a unique name for your module before publishing, and that name will need to be changed in a number of places if you intend to publish the module in this lab:

- Module File Name
- Module Manifest File Name
- External Help File Name
- Any references to the original module name in the module manifest

Once those changes have been made, all that's left for us to do is use the `Publish-Module` cmdlet in conjunction with our API key.

A quick note before proceeding: Once you publish a module to the PSGallery, there's no way to "un-publish" or delete it. You can un-list the module, which will remove it from all search results, but it will still be publicly accessible if you know the module name.

```
# Before running this, make sure your PowerShell working directory is the same
folder as the file you're currently reading!
$Env:PSModulePath = $ENV:PSModulePath+';'+$PWD

# User-supplied information
$NuGetAPIKey = Read-Host "Enter your NuGet API Key"
$ModuleName = Read-Host "Enter the name of your module"

# Publish module
Publish-Module -Name $ModuleName -NuGetApiKey $NuGetAPIKey -Verbose
```

Troubleshooting

If you encounter issues with publishing to the PSGallery, it's likely one of a few reasons:

- You're not using the right version of TLS
- Your NuGet version is outdated
- Your PowerShellGet version is outdated

All of these are simple fixes:

```
# Use TLS 1.2
[net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12

# Update NuGet
Install-PackageProvider -Name NuGet -Force
```

```
# Update PowerShellGet
Install-Module PowerShellGet -AllowClobber -Force
```

If you receive a 403 error, it's likely because:

- Your API key is invalid, expired, or doesn't have permission
- You chose a module/script name that's already in use
- You don't have all the required metadata in your module manifest

Since we just generated the API key, we can mostly rule that out. You can check whether a module/script exists by going to the following URL: `https://www.powershellgallery.com/packages/<YOUR_MODULE_NAME_HERE>`

In terms of metadata, the following is required in order to publish a module:

- Version
- Description
- Author
- A URI to the license terms of the module, either as part of the PrivateData section of the manifest, or in the LicenseUri parameter of the Publish-Module cmdlet.

If you're still having issues publishing, there are a wealth of resources available to help. Here's a few places to start:
https://learn.microsoft.com/en-us/powershell/gallery/faqs?view=powershellget-3.x
https://learn.microsoft.com/en-us/powershell/gallery/concepts/publishing-guidelines?view=powershellget-3.x
https://learn.microsoft.com/en-us/powershell/gallery/powershellget/update-powershell-51?view=powershellget-3.x
https://learn.microsoft.com/en-us/powershell/gallery/how-to/publishing-packages/publishing-a-package?view=powershellget-3.x

Happy coding!