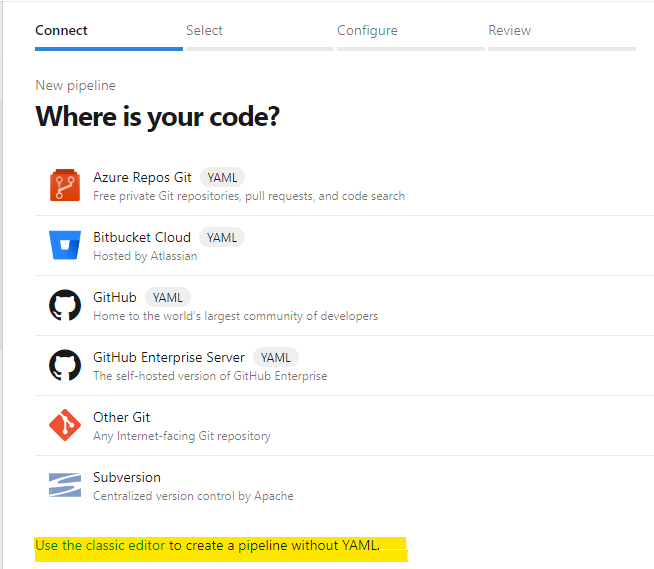
# Azure DevOps

Prerequisites

* A GitHub account.
* An Azure DevOps organization.

Build Pipeline

1. Sign into your Azure DevOps organization and navigate to your project.
2. In your project, navigate to the **Pipelines** page. Then choose the action to create a new pipeline. Click on classic editor as shown below.



1. Walk through the steps of the wizard by first selecting **GitHub** as the location of your source code.
2. You might be redirected to GitHub to sign in. If so, enter your GitHub credentials.
3. When the list of repositories appears, select your desired sample app repository.
4. Azure Pipelines will analyse your repository. Select Save and run, then select Commit directly to the master branch, and then choose Save and run again.
5. A new run is started. Wait for the run to finish.
6. There are many templates available to build common project types. Everything is customizable, and you can even start with an empty pipeline.

## PowerShell Scripts



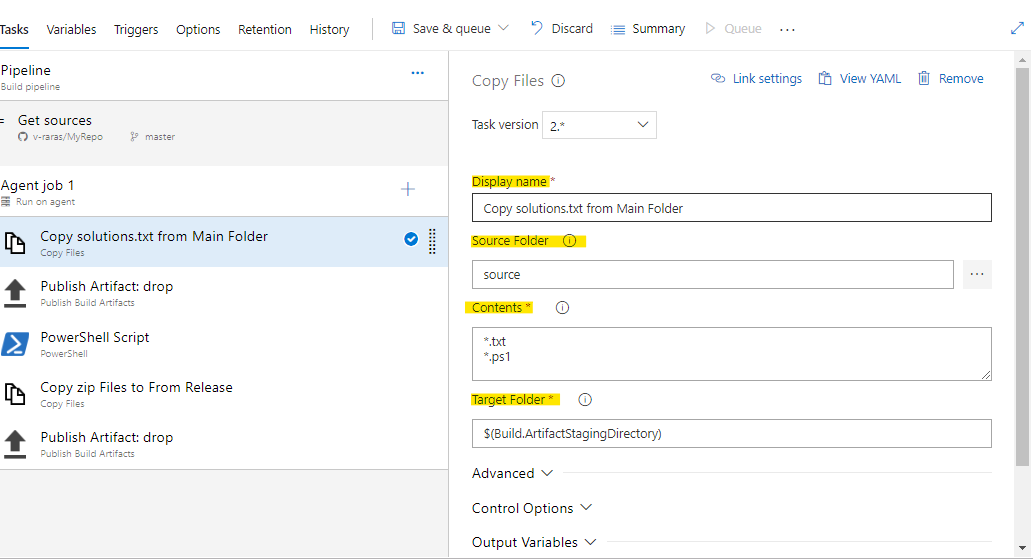
## Add Task to Build Pipeline:

* 1. Click the Add task button.
  2. Search for **Copy Files** task.

Select Copy Files task to copy files from a source folder to a target folder.

|  |  |
| --- | --- |
| **Argument** | **Description** |
| Source Folder | Folder that contains Solution.txt and PowerShell script |
| Contents | Specify match pattern filters that you want to apply to the list of files to be copied.  **Example**:  \*.txt  \*.ps1 |
| Target Folder | Folder where the files will be copied.  **Example**:  $(Build.ArtifactStagingDirectory) The local path on the agent where any artifacts are copied to before being pushed to their destination. |

Please refer below image.

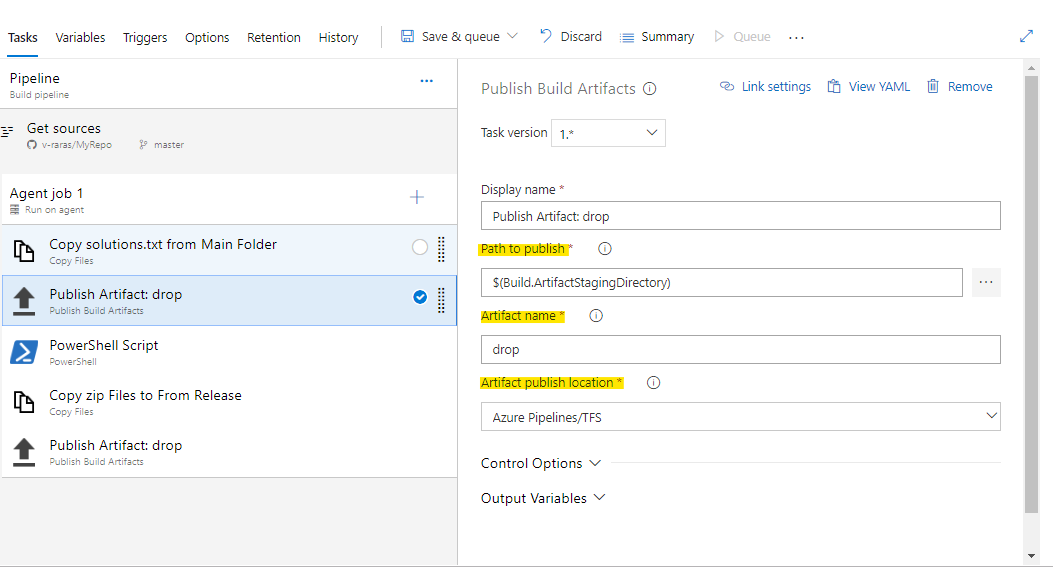


* 1. Search for **publish build artifacts** task.

Select **publish build artifacts** to build outputs to a staging directory and Publish staged artifacts.

|  |  |
| --- | --- |
| **Argument** | **Description** |
| Path to publish | Path to the folder or file you want to publish.  **Example**:  $(Build.ArtifactStagingDirectory) The local path on the agent where any artifacts are stored. |
| Artifact name | Specify the name of the artifact that you want to create.  **Example**:  drop |
| Artifact publish location | Choose Azure Pipelines/TFS |

Please refer below image.

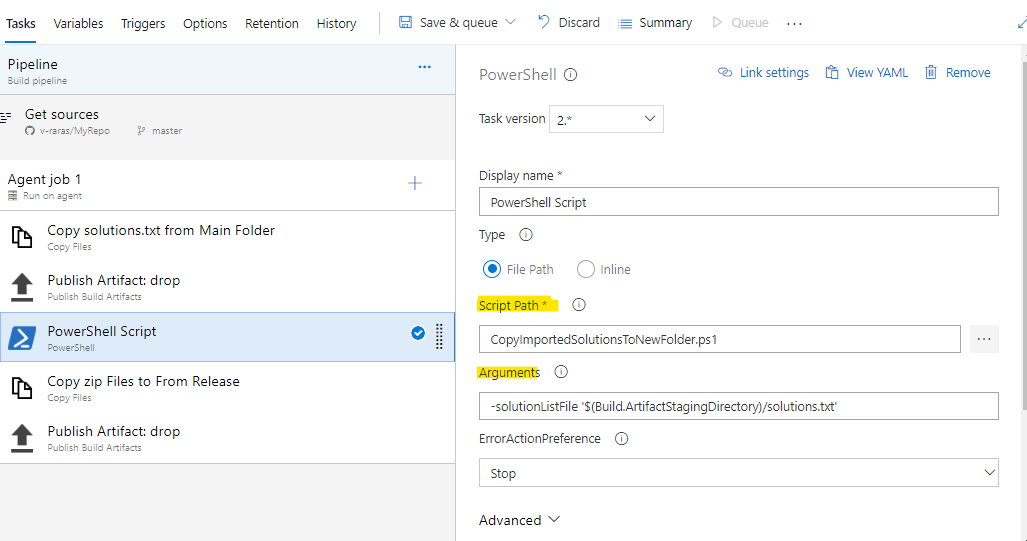


* 1. Search for **PowerShell** task.

Select **PowerShell** task to run a PowerShell script

|  |  |
| --- | --- |
| **Argument** | **Description** |
| Type | Set as an inline script |
| Script Path | Contents of the script, Choose PowerShell script CopyImportedSolutionsToNewFolder.ps1 |
| Arguments | Arguments passed to the PowerShell script.  **Example:**  -solutionListFile'$(Build.ArtifactStagingDirectory)/solutions.txt' |

Please refer below image.

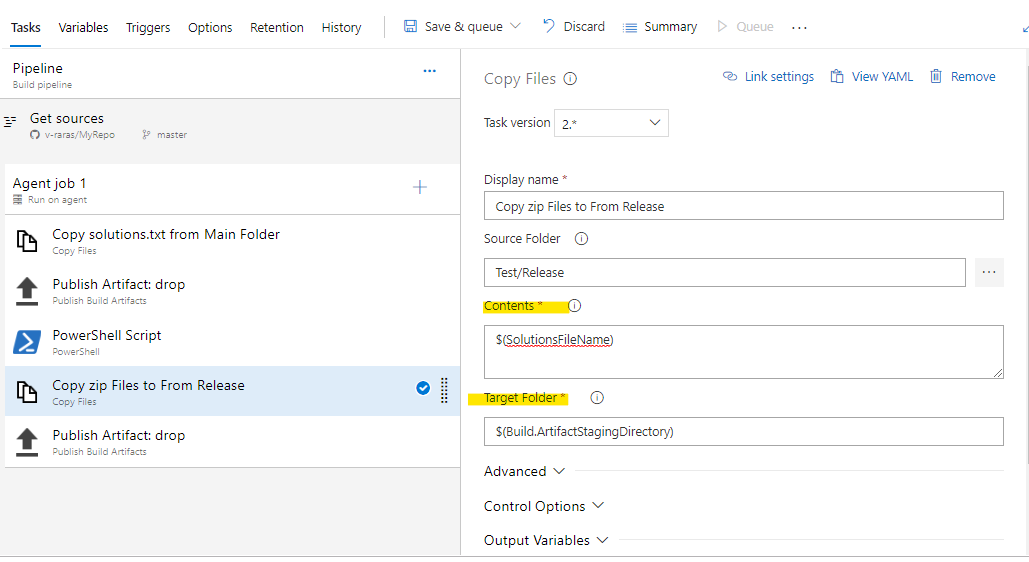


* 1. Search for **Copy Files** task.

Select Copy Files task to copy files from a source folder to a target folder.

|  |  |
| --- | --- |
| **Argument** | **Description** |
| Source Folder | Folder that contains CRM Solutions to be deployed |
| Contents | Specify match pattern filters that you want to apply to the list of files to be copied.  **Example**:  $(SolutionsFileName) |
| Target Folder | Folder where the files will be copied.  **Example**:  $(Build.ArtifactStagingDirectory) The local path on the agent where any artifacts are copied to before being pushed to their destination. |

Please refer below image.

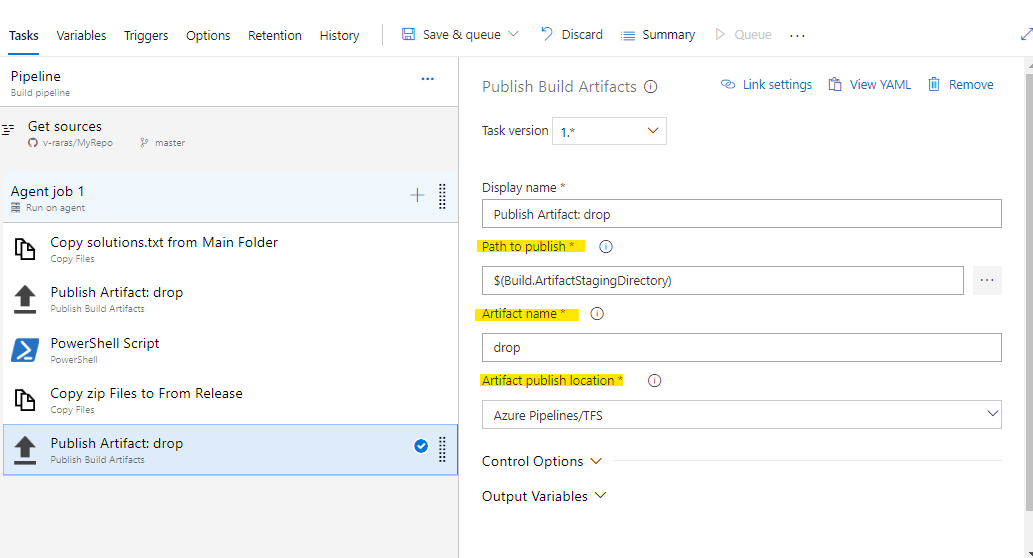


* 1. Search for **publish build artifacts** task.

Select **publish build artifacts** to build outputs to a staging directory and Publish staged artifacts.

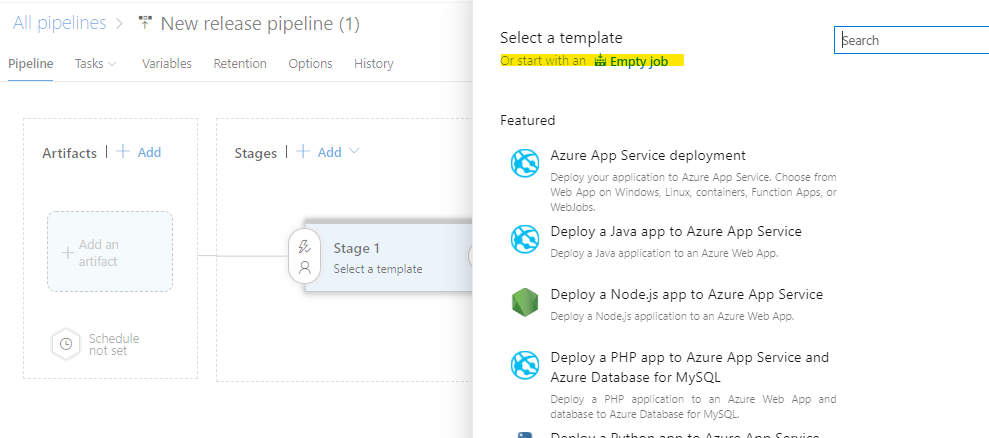
|  |  |
| --- | --- |
| **Argument** | **Description** |
| Path to publish | Path to the folder or file you want to publish.  **Example**:  $(Build.ArtifactStagingDirectory) The local path on the agent where any artifacts are stored. |
| Artifact name | Specify the name of the artifact that you want to create.  **Example**:  drop |
| Artifact publish location | Choose Azure Pipelines/TFS |

Please refer below image.

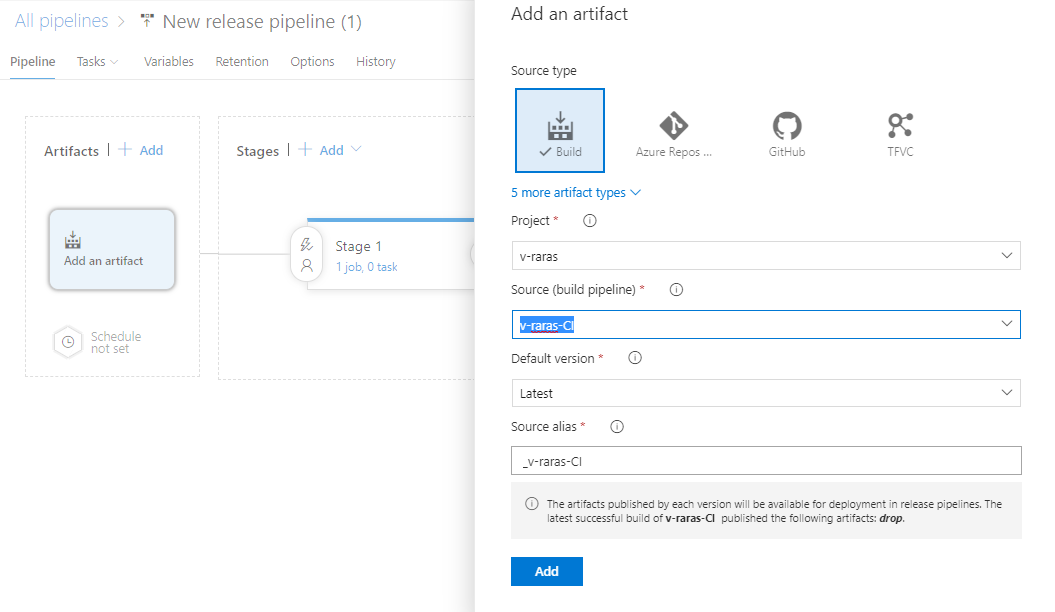


Release Pipeline:

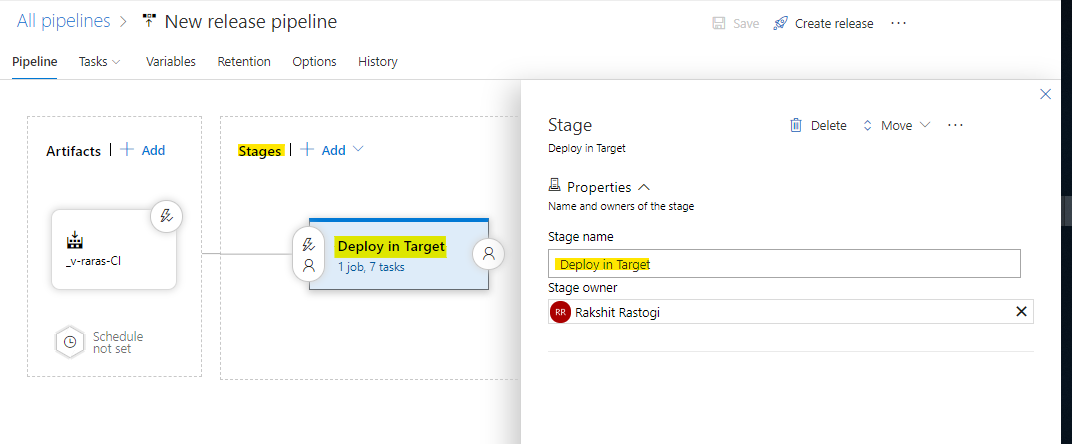
1. In your project, navigate to the **Releases** page. Then choose the action to create a new pipeline. In Select a Template as **Empty Job**.



1. Click on the Artifacts within the release pipeline. Choose the Build Pipeline that you have created.

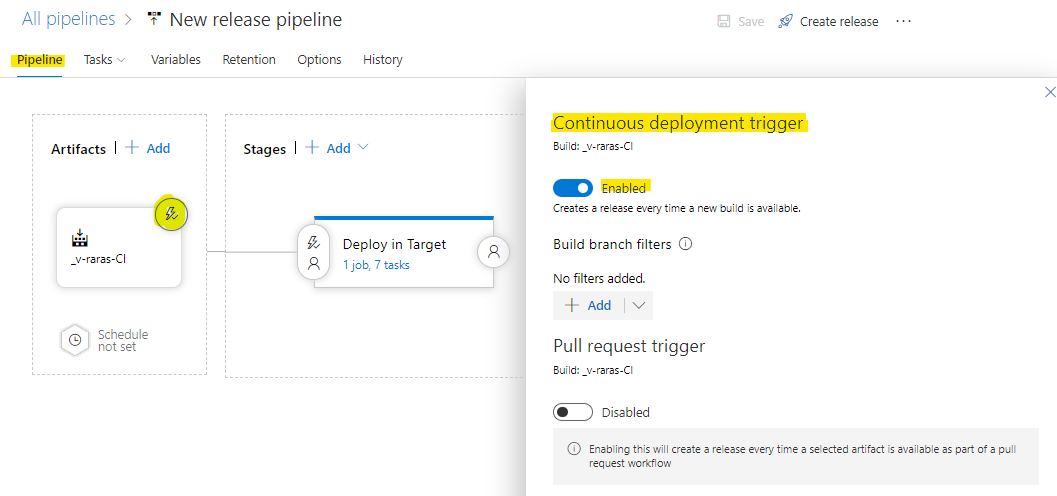


1. Click on the stage within the release pipeline and rename the Stage Name.



1. To set continuous deployment trigger. Click on the trigger as shown in below image.

Set that continuous deployment to enabled.

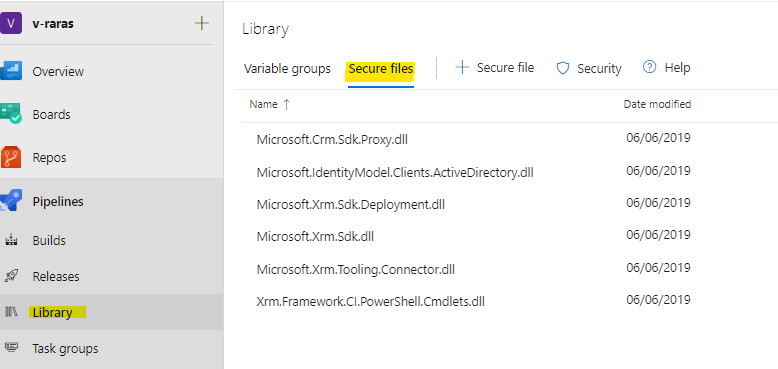


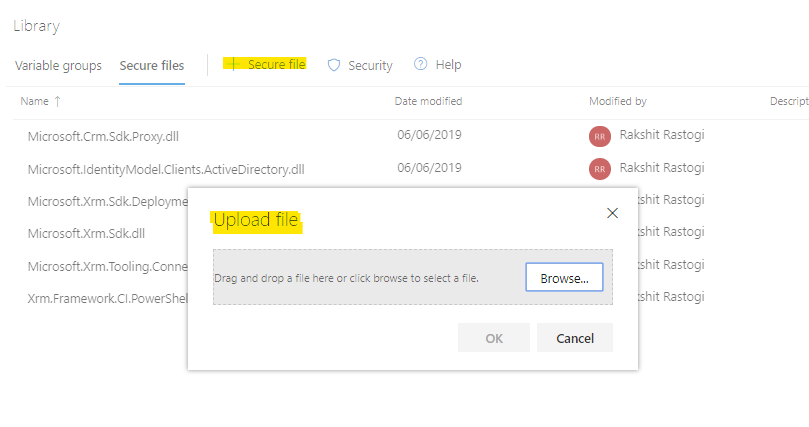
## Add Secure File

1. Follow the steps to add listed .dll in Library.

* Microsoft.Xrm.Sdk.dll
* Microsoft.Crm.Sdk.Proxy.dll
* Microsoft.IdentityModel.Clients.ActiveDirectory.dll
* Microsoft.Xrm.Sdk.Deployment.dll
* Microsoft.Xrm.Tooling.Connector.dll
* Xrm.Framework.CI.PowerShell.Cmdlets.dll

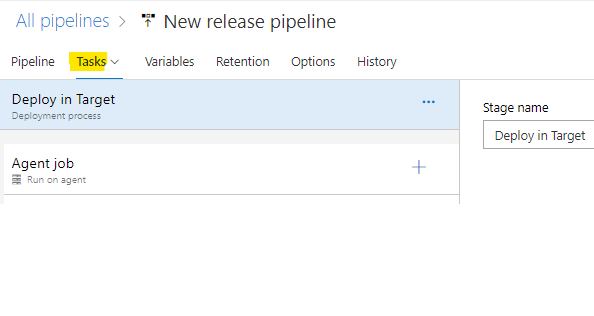


1. Navigate to the Library tab in Azure Pipelines and Select the Secure files tab at the top.
2. Click on the + Secure file and upload .dll and click ok.



## Add Task to Release Pipeline:

1. In Release Pipeline, Click on Task as shown as below.

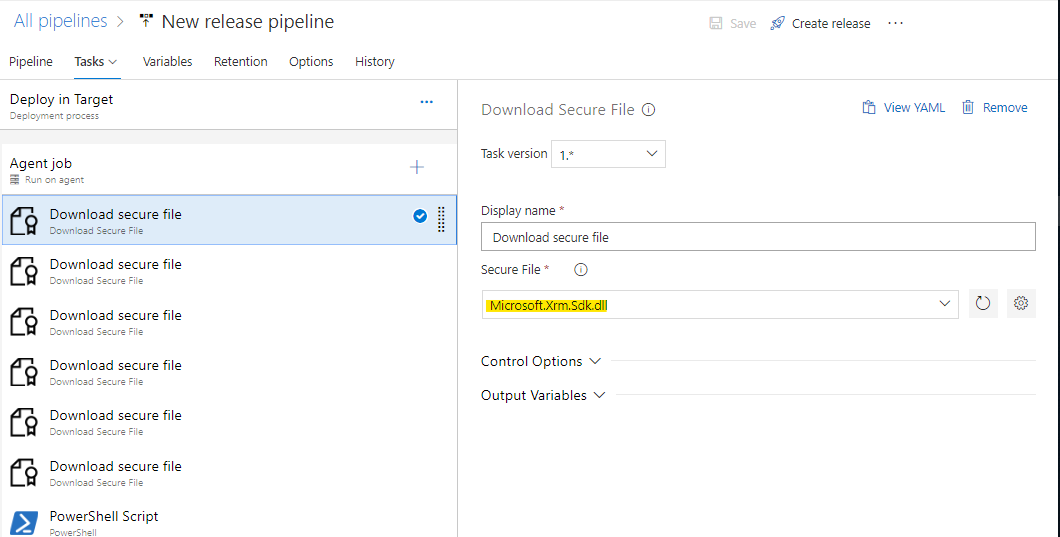


1. Search for **Download Secure File** task.

Select Download Secure File to download a secure file to a temporary location on the release agent.

|  |  |
| --- | --- |
| **Argument** | **Description** |
| Secure File | Select the secure file to download to a temporary location.  **Example**: Microsoft.Xrm.Sdk.dll  **Note**: The file will be deleted after the build or release. |

1. Please refer below image.



1. Follow the Step:2 and add below secure files from Library.

* Microsoft.Crm.Sdk.Proxy.dll
* Microsoft.IdentityModel.Clients.ActiveDirectory.dll
* Microsoft.Xrm.Sdk.Deployment.dll
* Microsoft.Xrm.Tooling.Connector.dll
* Xrm.Framework.CI.PowerShell.Cmdlets.dll
  + Search for **PowerShell** task.

1. Select **PowerShell** task to run a PowerShell script

|  |  |
| --- | --- |
| **Argument** | **Description** |
| Type | Set as an inline script |
| Script Path | Contents of the script, Choose PowerShell script from the drop location.  **Example**: $(System.DefaultWorkingDirectory)/\_v-raras-CI/drop/MultilpleSolutionsImport.ps1 |
| Arguments | Arguments passed to the PowerShell script.  **Example:**  -solutionListFile '$(System.DefaultWorkingDirectory)/\_v-raras-CI/drop/  solutions.txt'  -solutionImportPath '$(System.DefaultWorkingDirectory)/\_v-raras-CI/drop'  -solutionContainer '$(System.DefaultWorkingDirectory)/\_v-raras-CI/drop/SolutionsToBeImported'  -crmConnectionString 'AuthType=Office365;Username=\*\*\*\*\*\*;  Password=\*\*\*\*\*\*;Url=\*\*\*\*\* '  -override 1  -publishWorkflows 1  -overwriteUnmanagedCustomizations 1  -skipProductUpdateDependencies 1  -convertToManaged 0  -holdingSolution 0  -AsyncWaitTimeout 0  -logsDirectory ''  -logFilename '' |

Please refer below image.

