**Configure Self Hosted agent on Windows**

**Pre-requirement:**

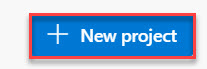
Java Application we are deploying so configure Maven, Java directory. Also Set path in System Environment.

**Steps:**

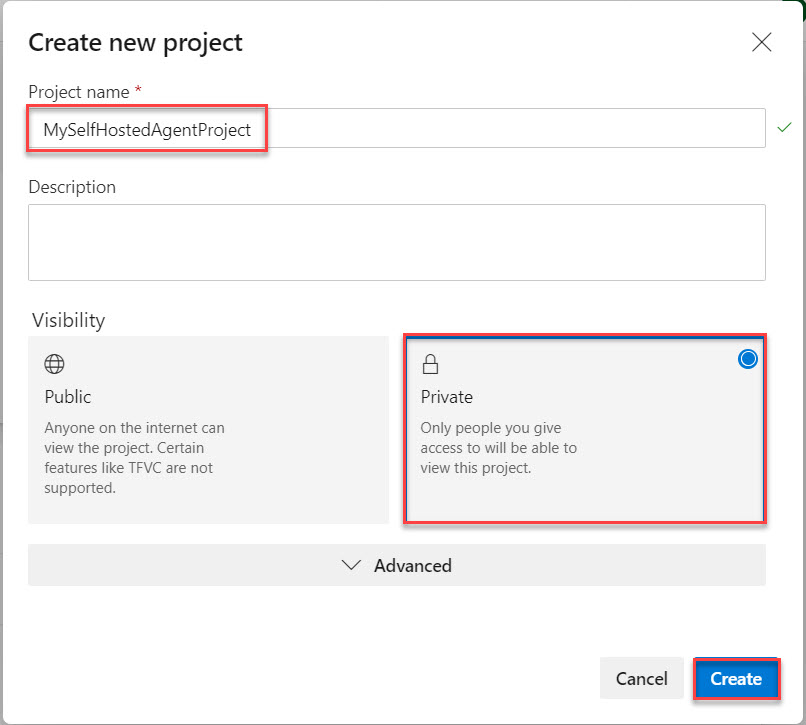
Step 1: Open Azure DevOps portal

[https://dev.azure.com](https://dev.azure.com/)

Step 2: Create New Project by click on + New Project



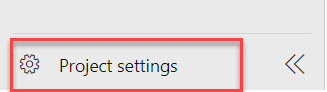
Step 3: Create New Project



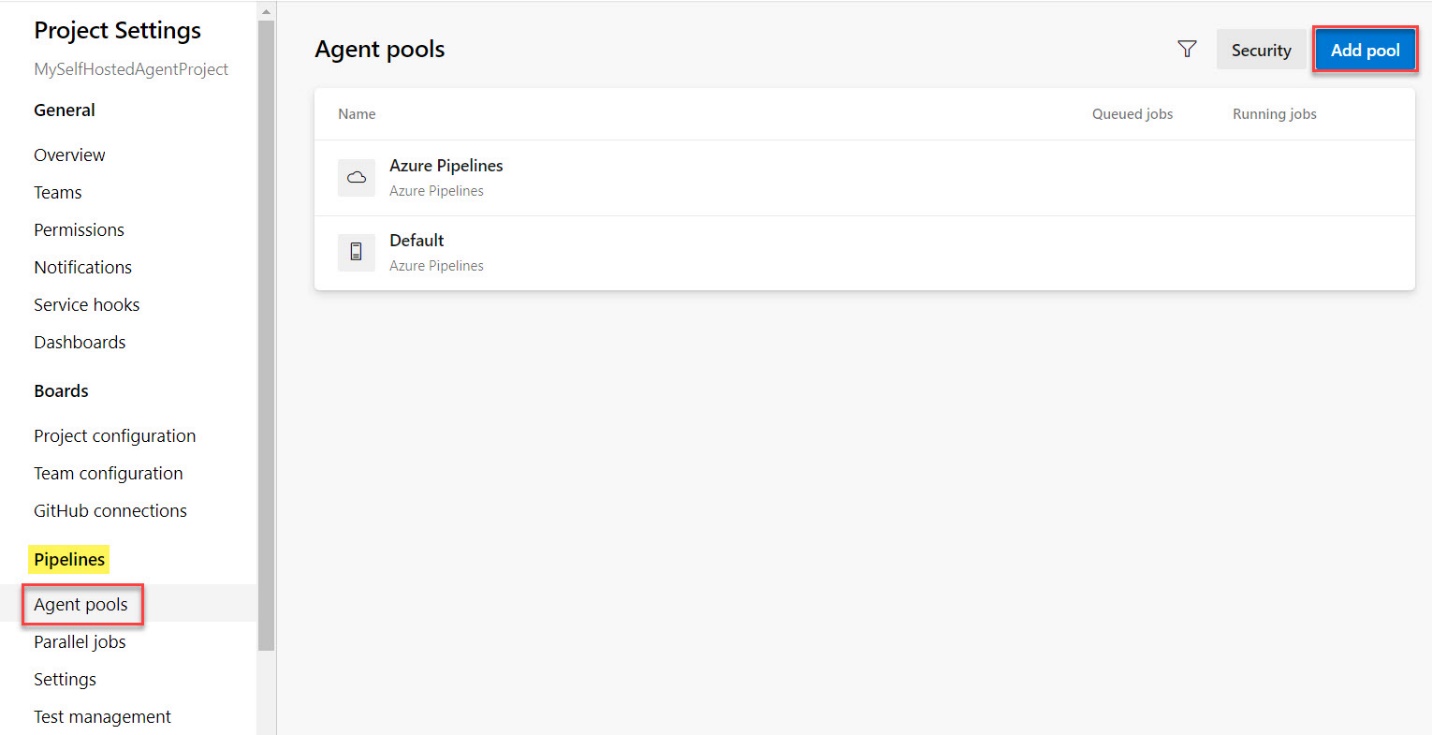
Project Name: **MySelfHostedAgentProject**

Visibility: Choose any one of them

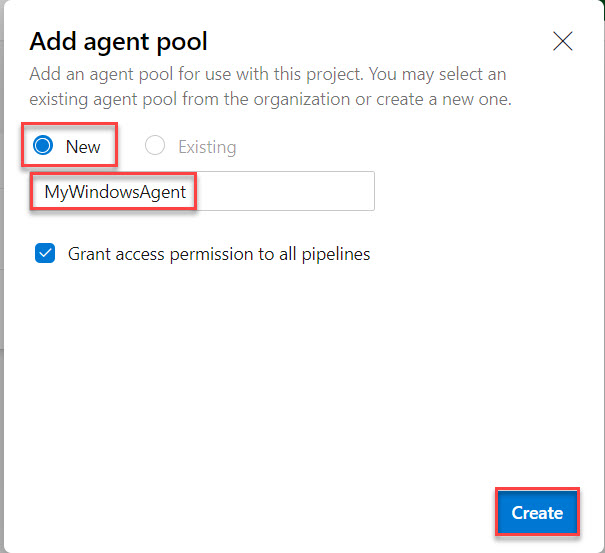
Step 4: Click on Project Settings



Step 5: Click on Agent Pools option and select Add pool



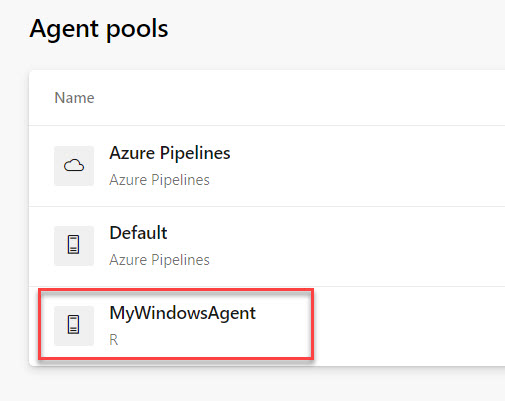
Step 6: Add Agent Pool



Agent Pool: **New**

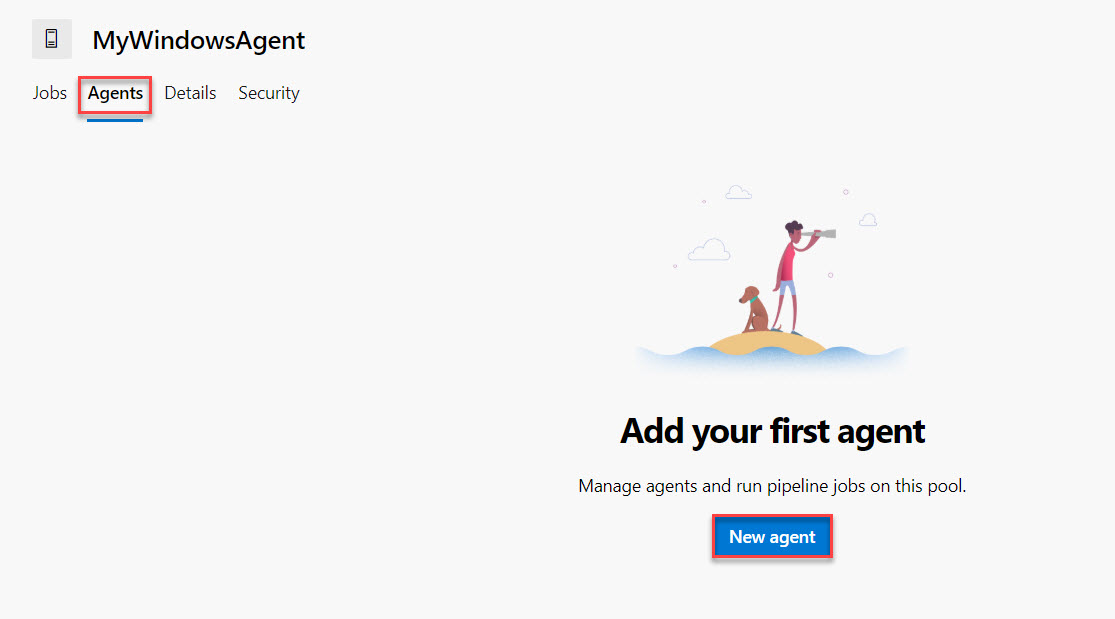
Name: **MyWindowsAgent**

Step 7: Click on Newly created Agent: MyWindowsAgent



Step 8: Select Agents tab

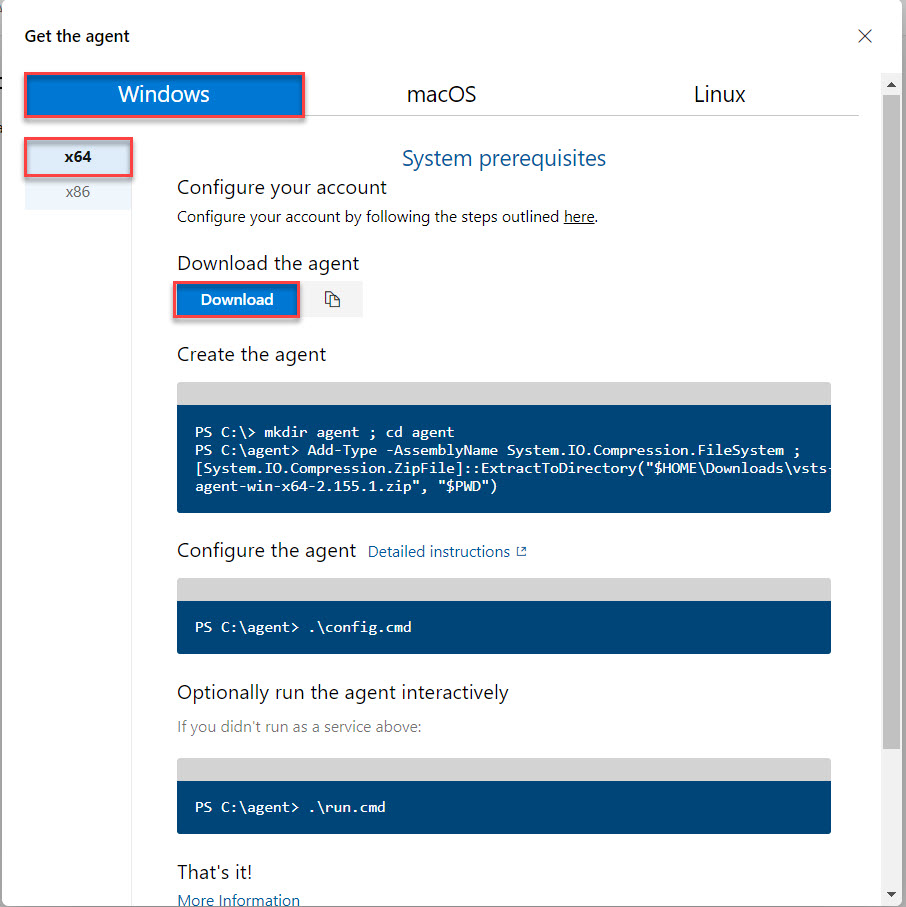
Click on New agent button



Step 9: Select Operating system option.

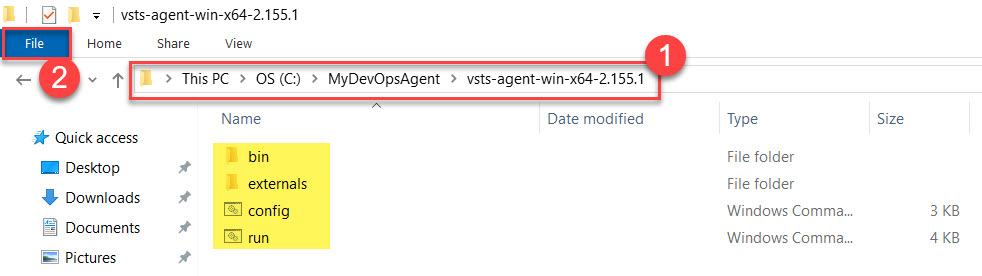
For this demo we are going with Windows and x64 version

Click on Download button.



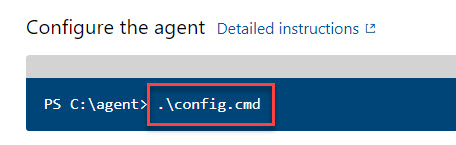
Step 10: Create new folder and extract that agent.

Click on File menu



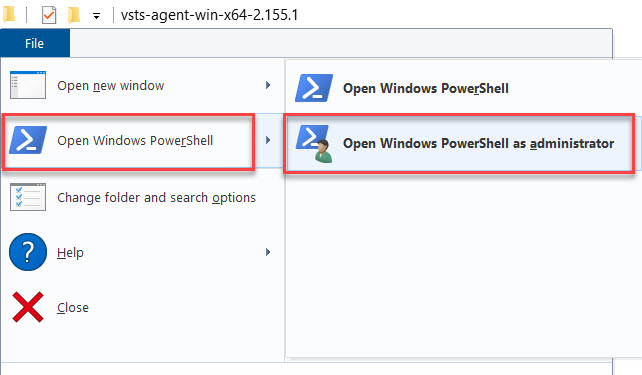
Step 11: Navigate to Azure DevOps Portal

Copy Config command: .\config.cmd

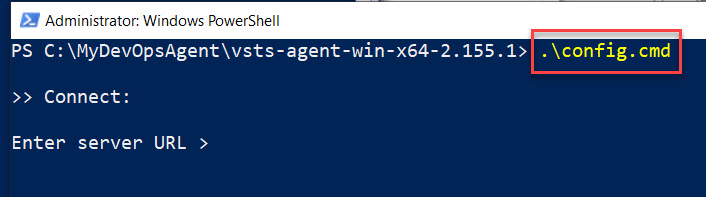


Step 12: Start PowerShell with Administrator option

Open Windows PowerShell -> Open Windows PowerShell as administrator



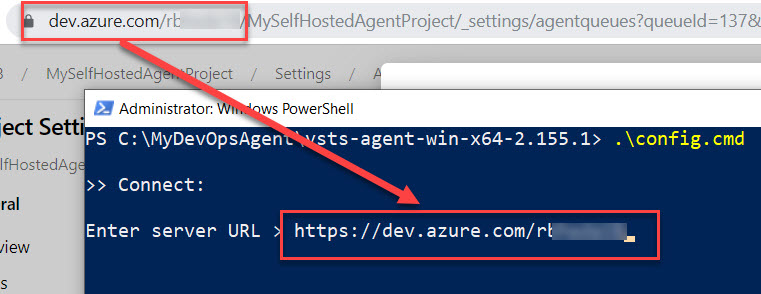
Step 13: Paste that config command or type: .\config.cmd



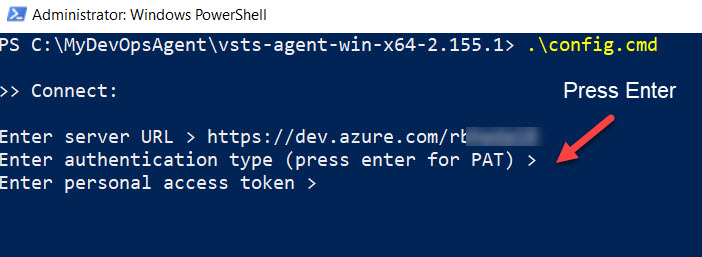
Step 14: Enter Server URL

Collect from Azure DevOps Portal with organization name

Ex. <https://dev.azure.com/contoso>

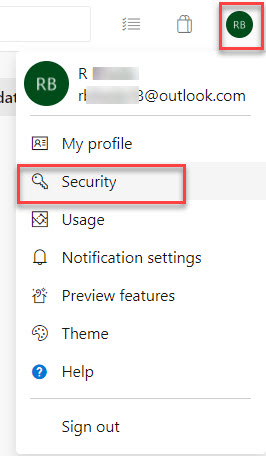


Step 15: Go with Personal Access Token (PAT) authentication type. Press Enter for that.

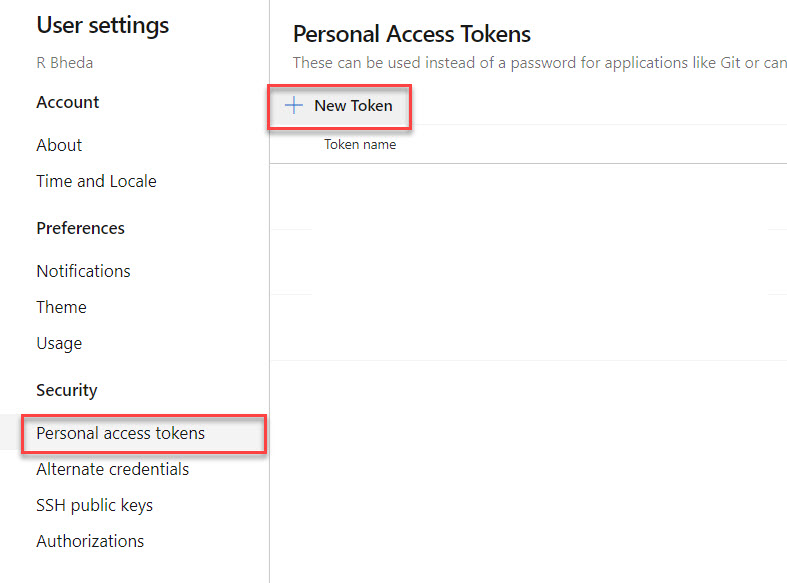


Step 16: To collect PAT follow below steps

Click on User icon and select Security option



Step 17: Click on + New Token under Personal Access tokens option



Step 18: Create New Personal Access Token

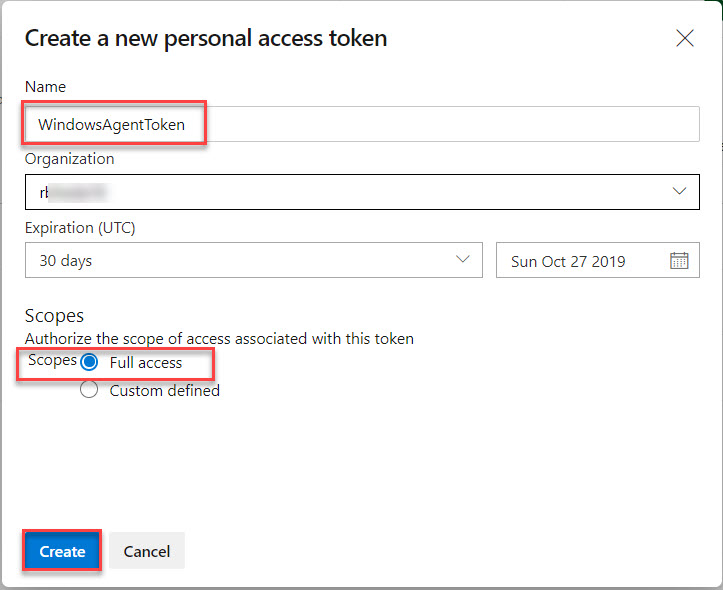
Name: WindowsAgentToken

Organization: Choose current organization

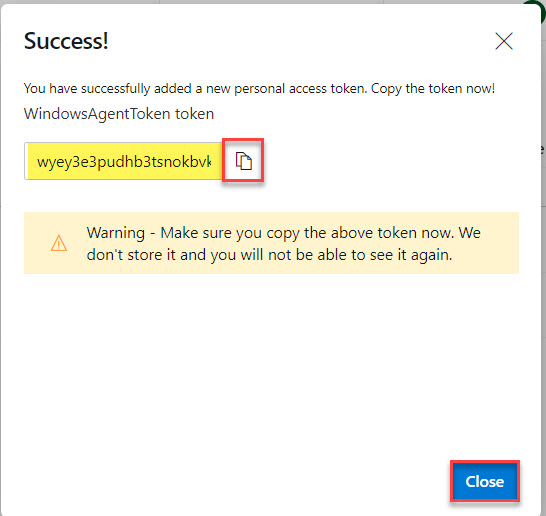
Expiration: According to your requirement.

Scopes: Full access

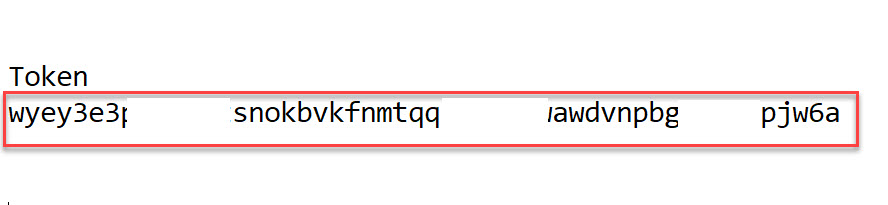
Click on Create button.



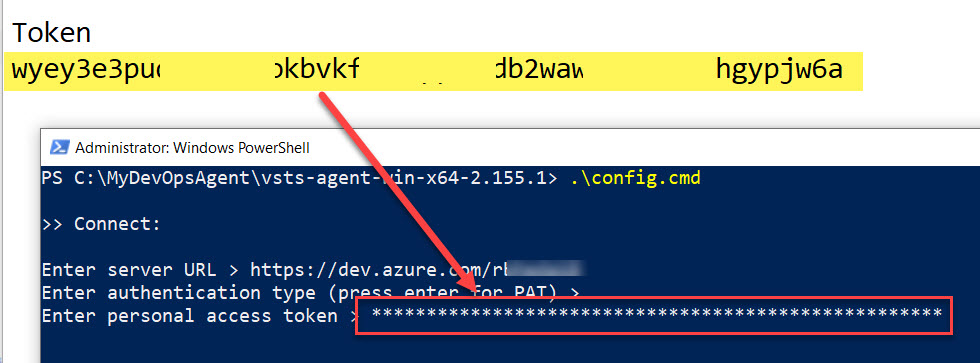
Step 19: Copy PAT into notepad for further steps.



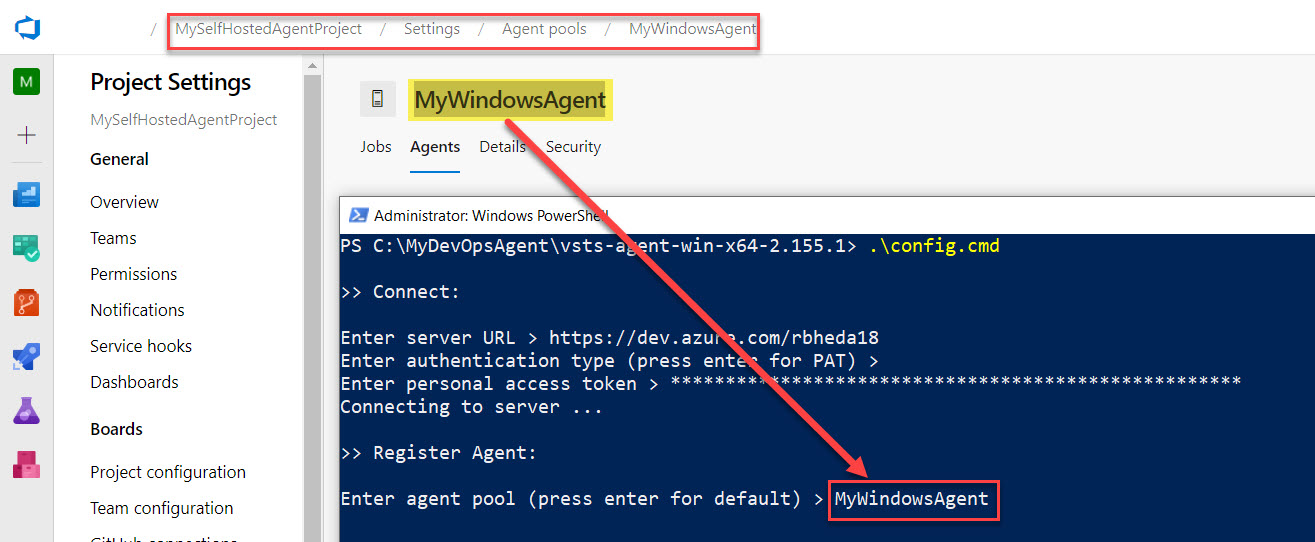
Paste into Notepad



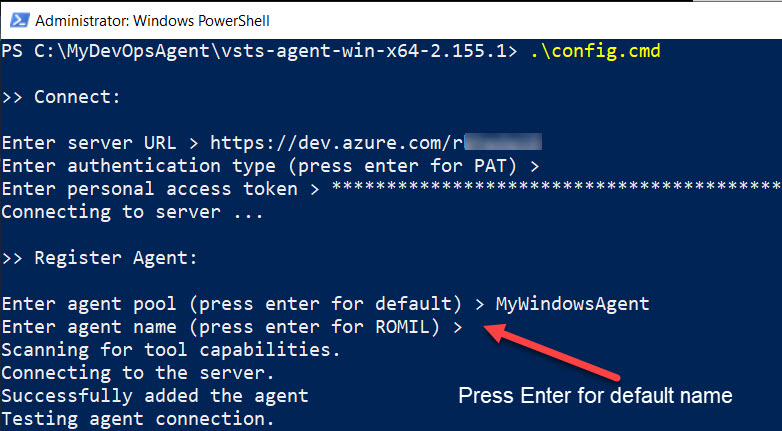
Step 20: Paste token into PowerShells



Step 21: Enter Agent Pool Name



Step 22: Agent Name: go with default so press enter key

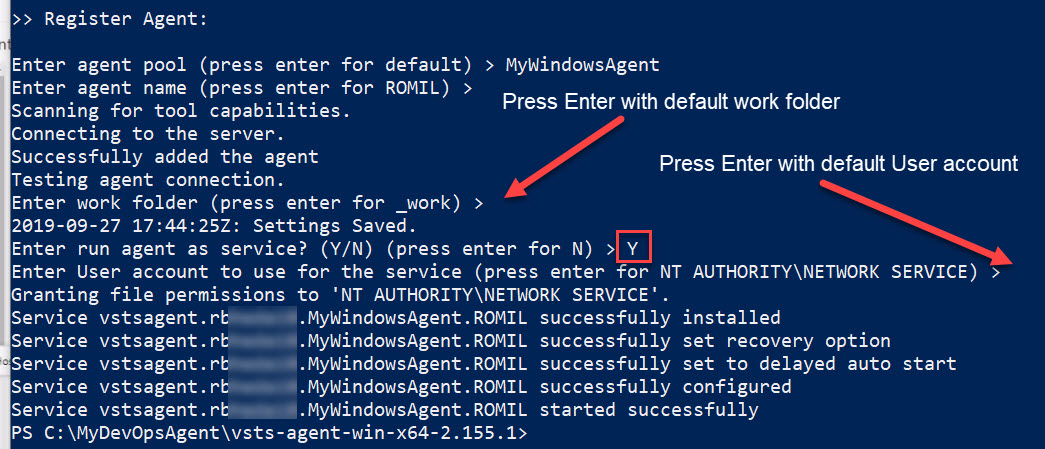


Step 23: Choose Work folder: Press enter for default folder or enter any folder name

Run agent as service: Y

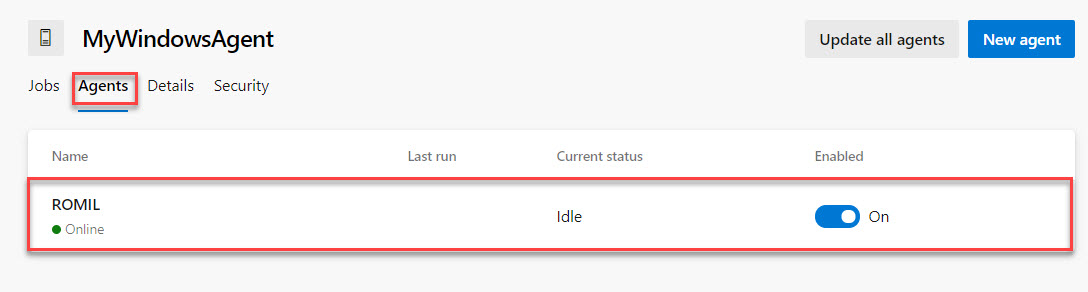
User account: Press enter for default user

Service will configure successfully.

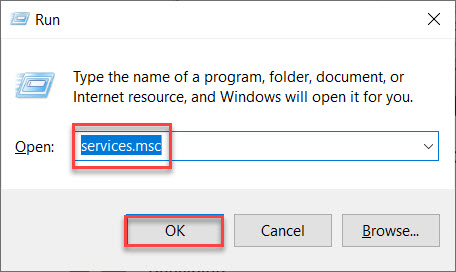


Step 24: Navigate to Azure DevOps portal and select Agents tab.

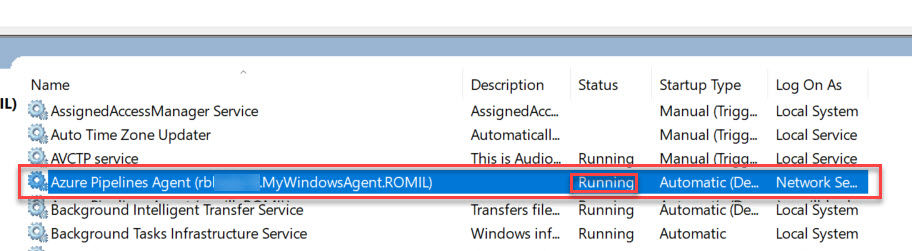
Agent name available with Online status.



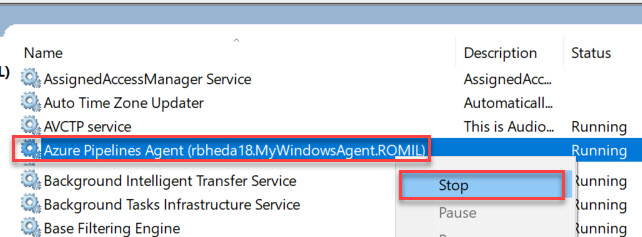
Step 25: Open Run window (Windows Key + R) and type services.msc



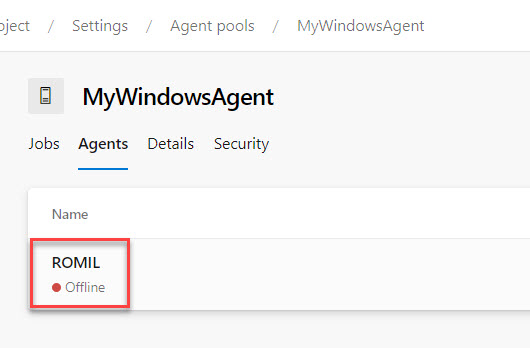
Step 26: Search for Azure Pipeline Agent and you can see status is running.



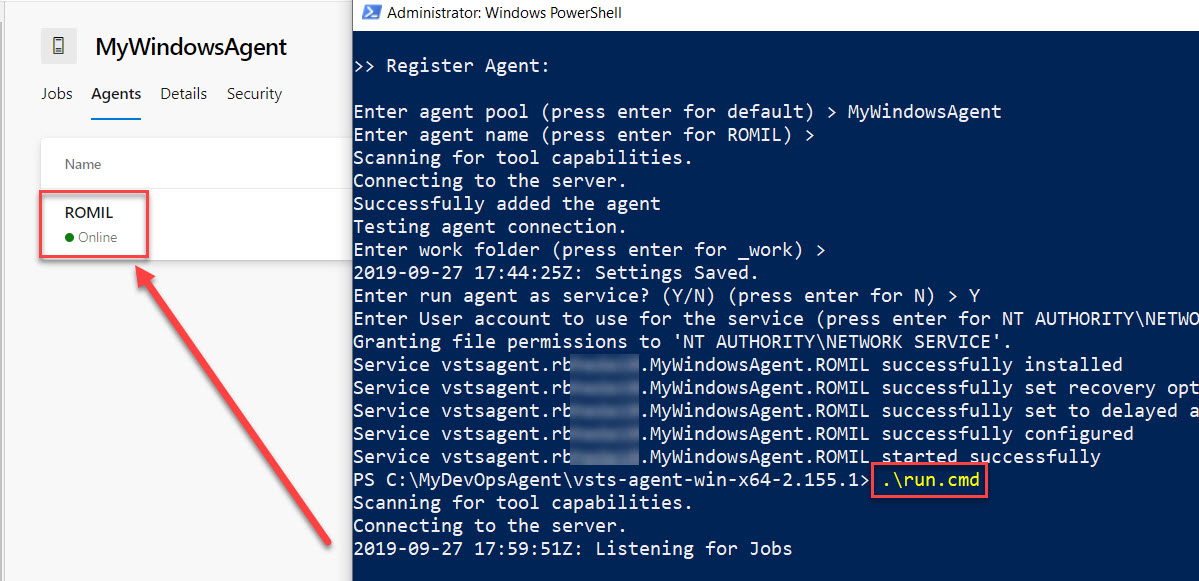
Step 27: Right Click on Azure Pipeline Agent and select Stop option



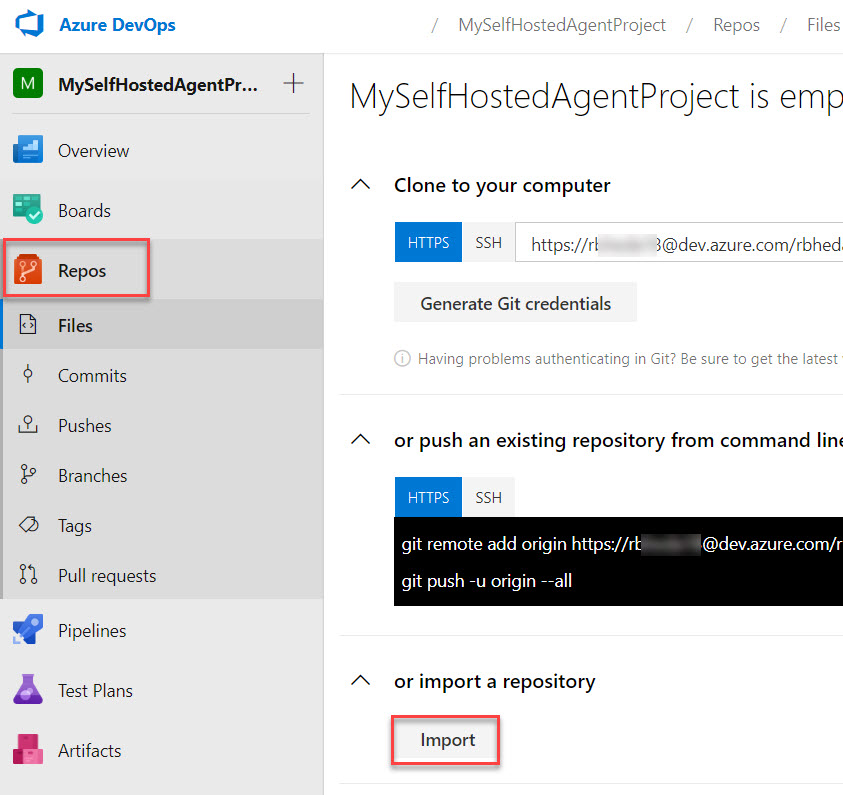
Step 28: Check status of agent: Offline



Step 29: Navigate to PowerShell and type .\run.cmd

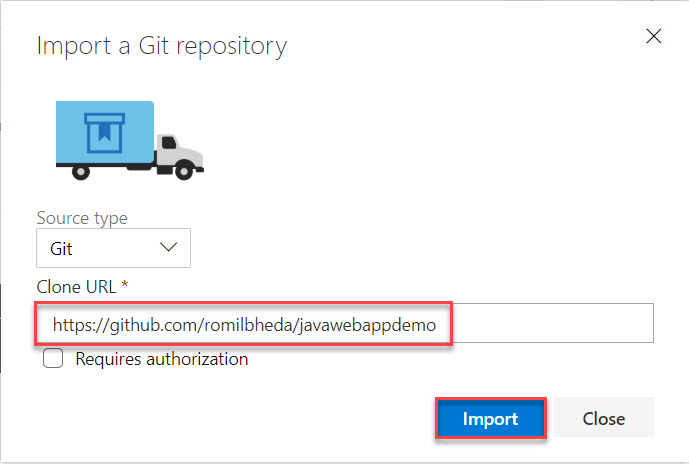


Step 30: Select Repos and click on Import button.

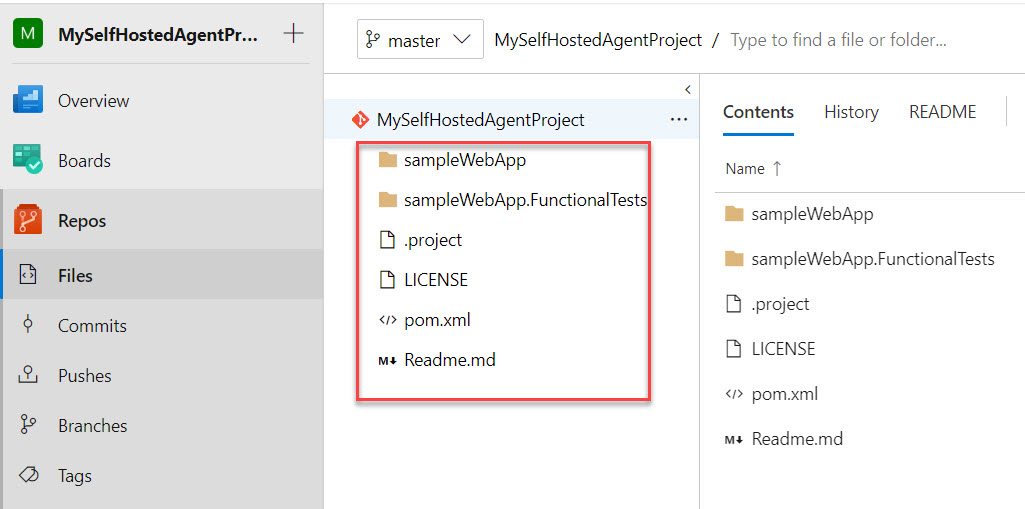


Step 31: Enter Clone URL: <https://github.com/romilbheda/javawebappdemo>

Click on Import button.



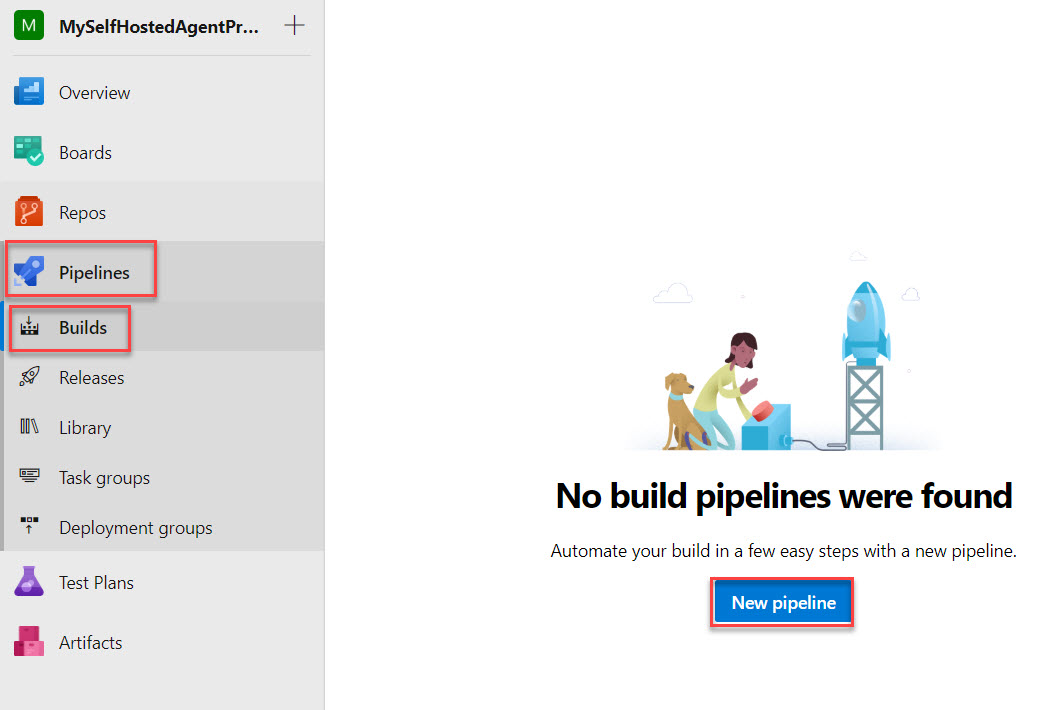
Step 32: Wait for few seconds to load all files as below:



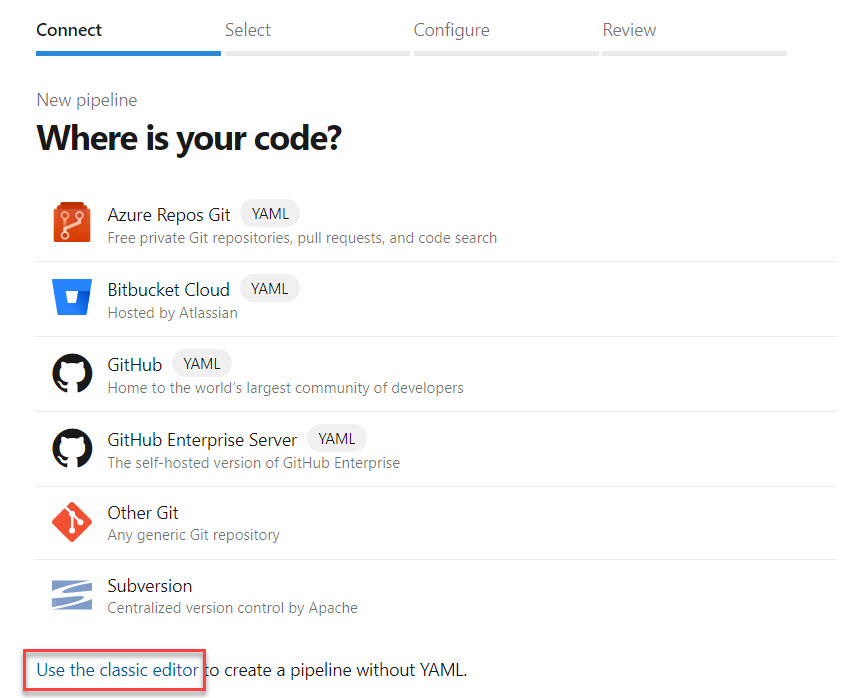
Step 33: Now time to start work with Pipelines

Pipelines -> Builds

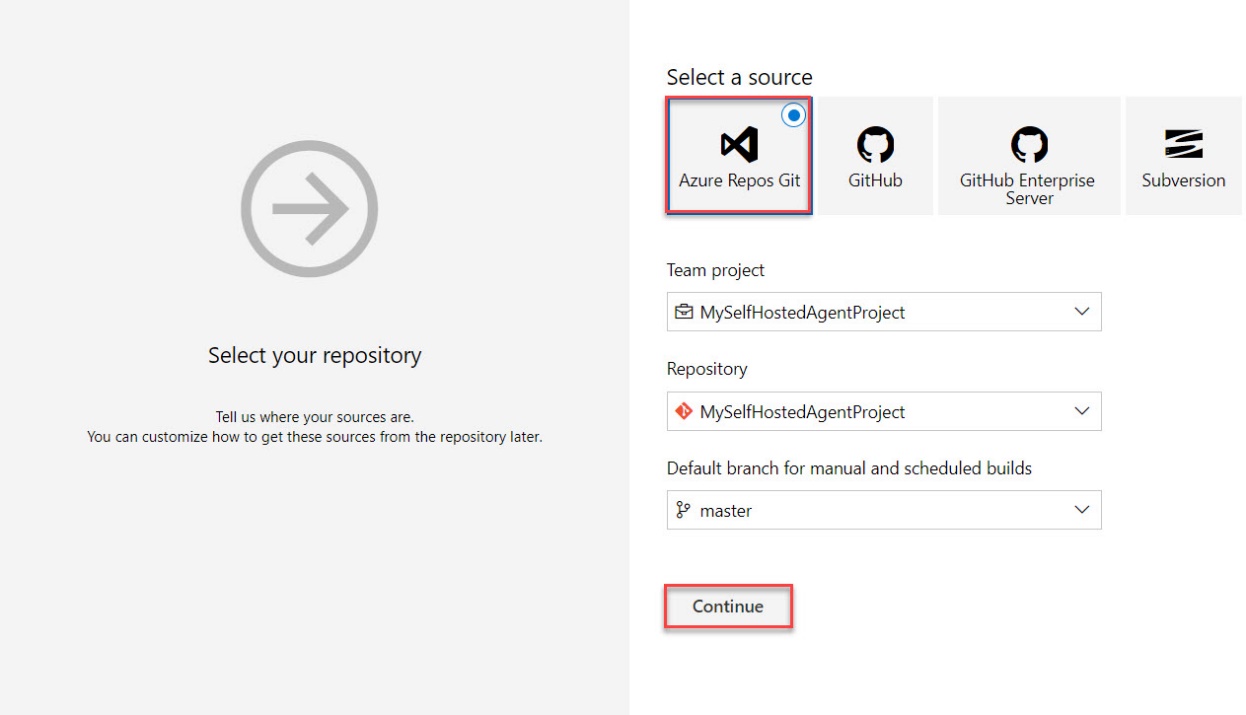
Click on New pipeline



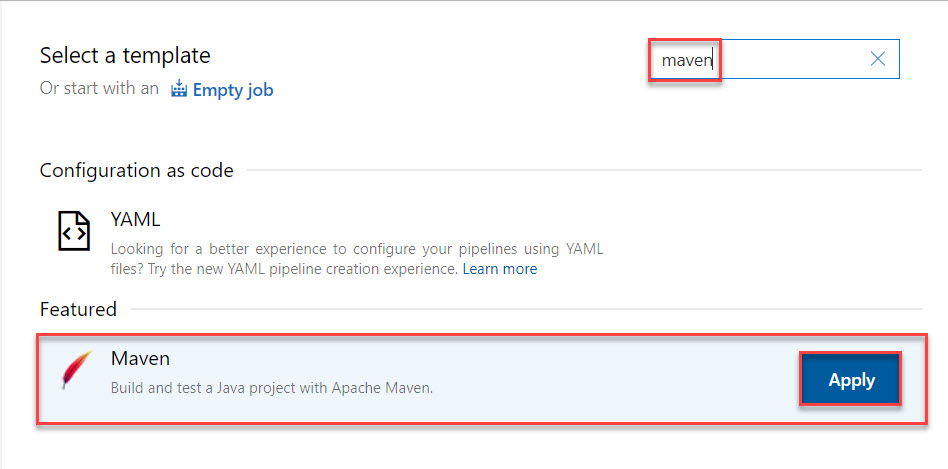
Step 34: Click on Use the classic editor option



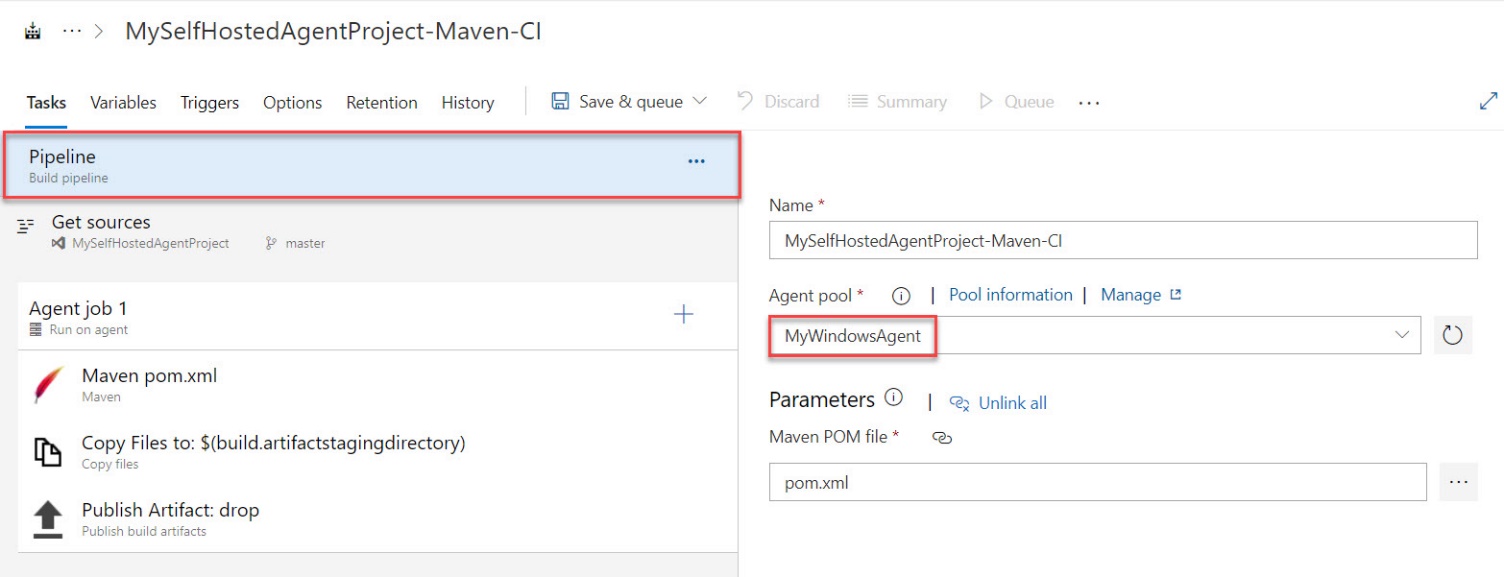
Step 35: Select Source: Azure Repos Git and click on Continue button.



Step 36: Search for maven and Click on Apply button.

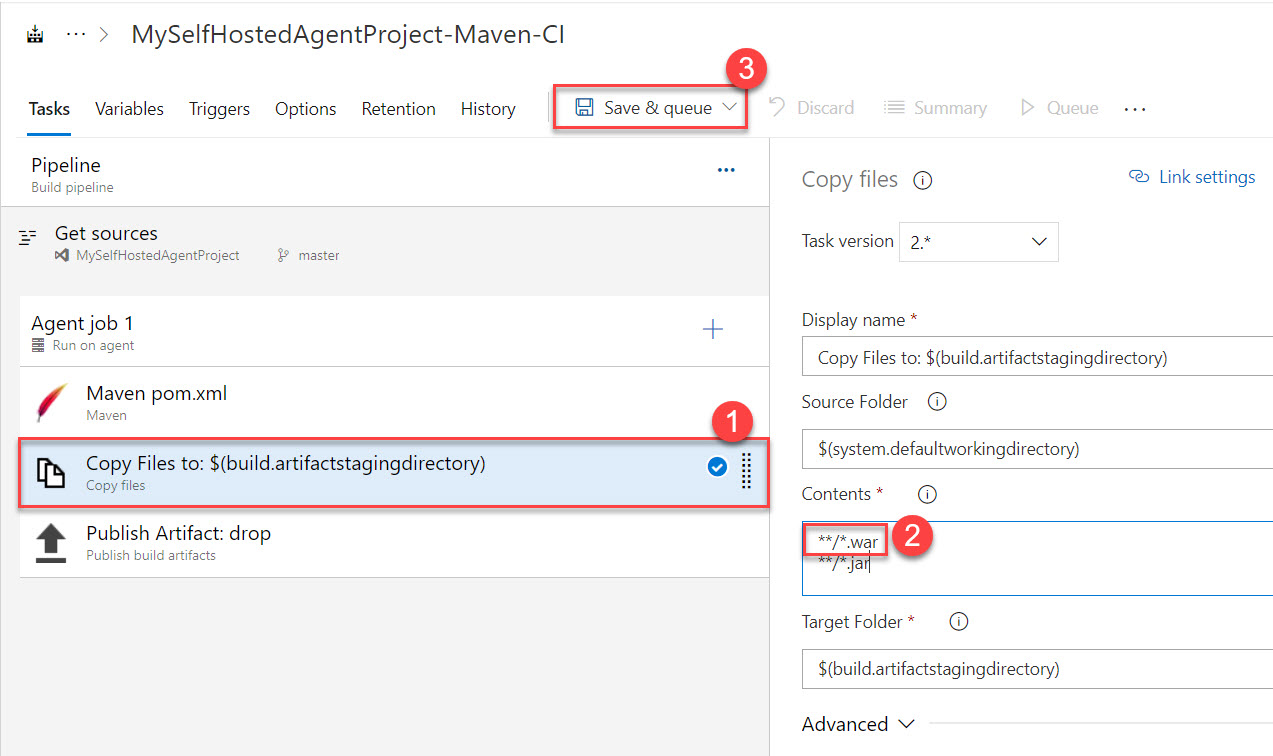


Step 37: Select Pipeline and Choose Agent Pool: MyWindowsAgent

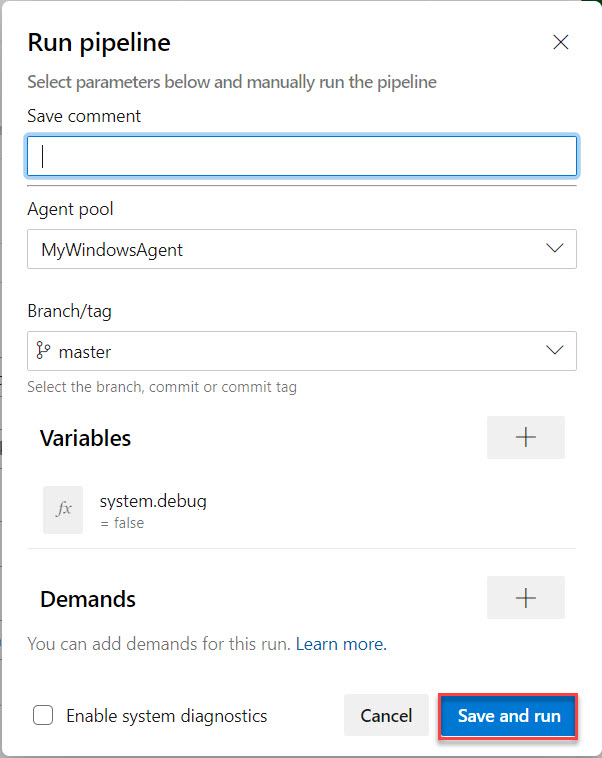


Step 38: Choose Copy Files to Job and Add one line in Contents: \*\*/\*.war

Click on Save & Queue option

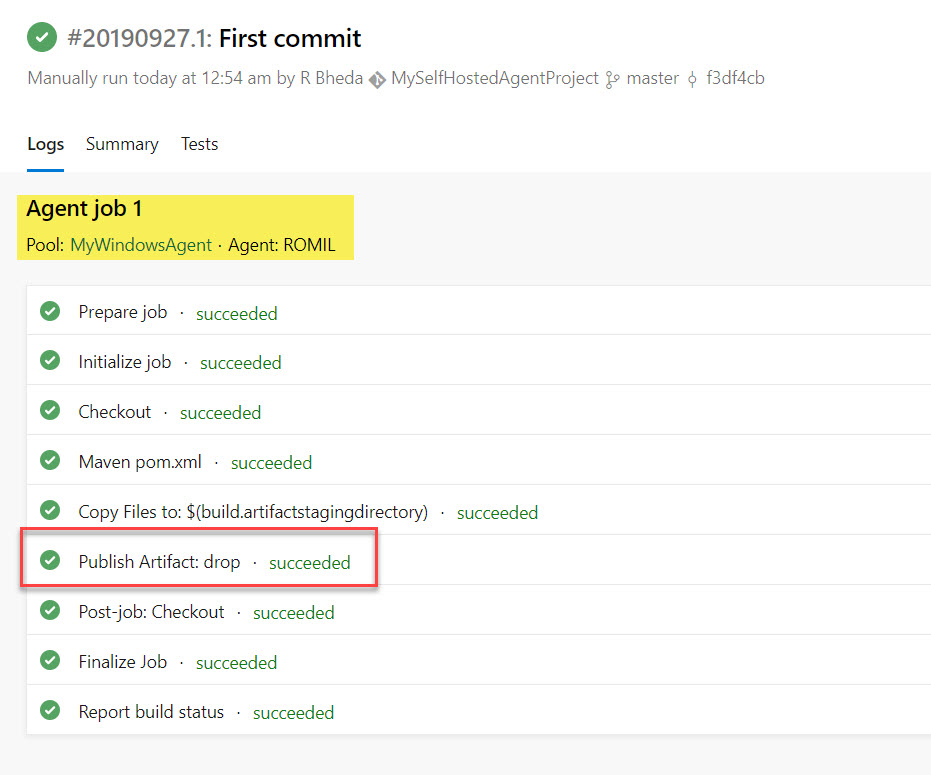


Step 39: Click on Save and run button.

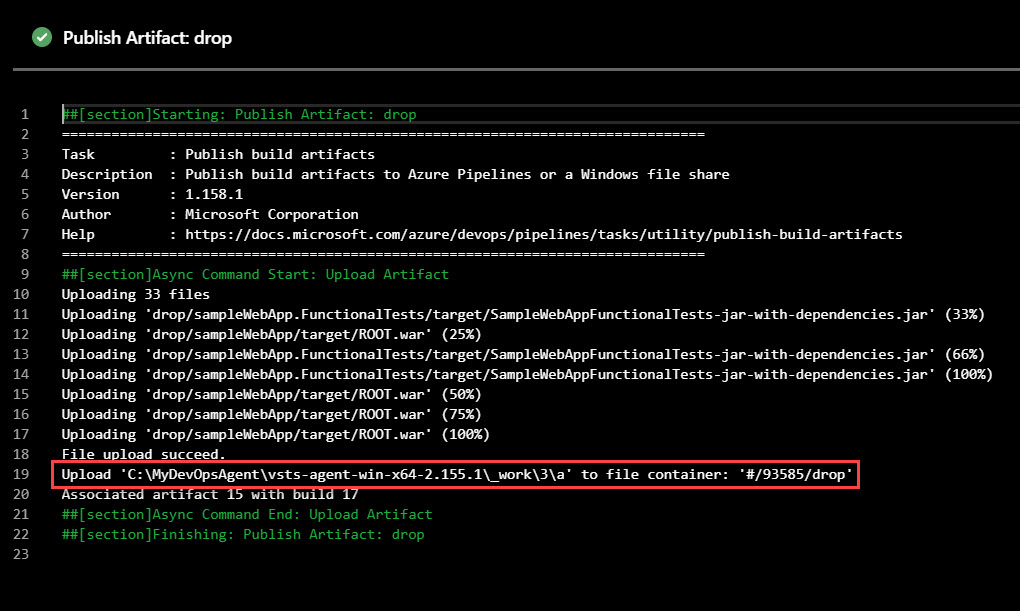


Step 40: Wait for few minutes to complete build

Click on Publish Artifact: drop



Step 41: Check Upload line for location



Step 42: Check into your system for Build

