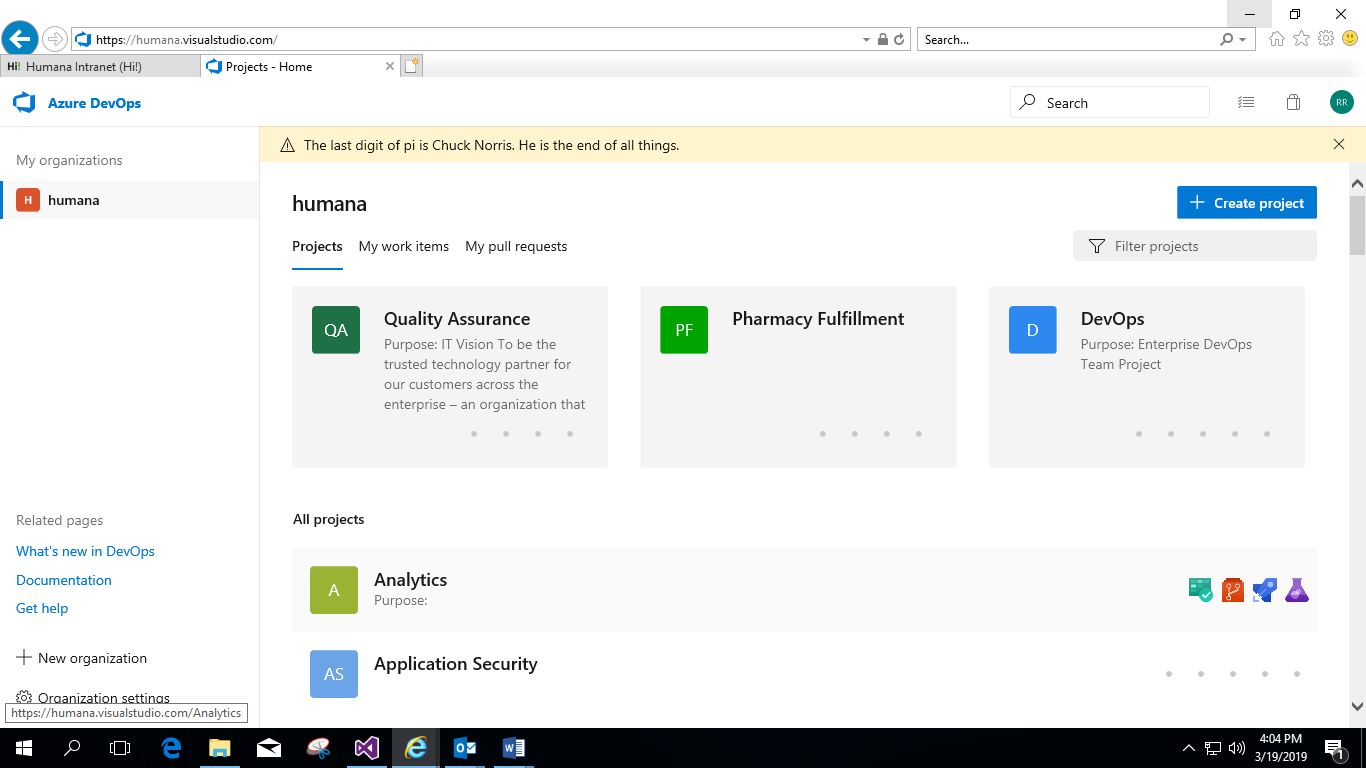
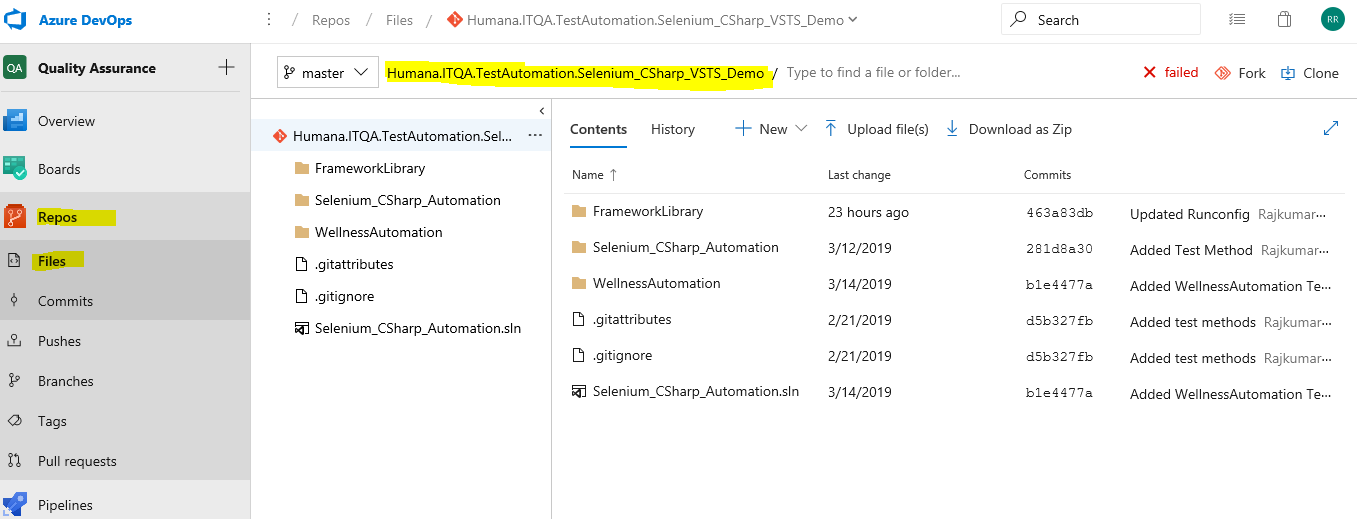
**VSTS Integration with SELENIUM Solutions**

# **Build Pipeline creation**

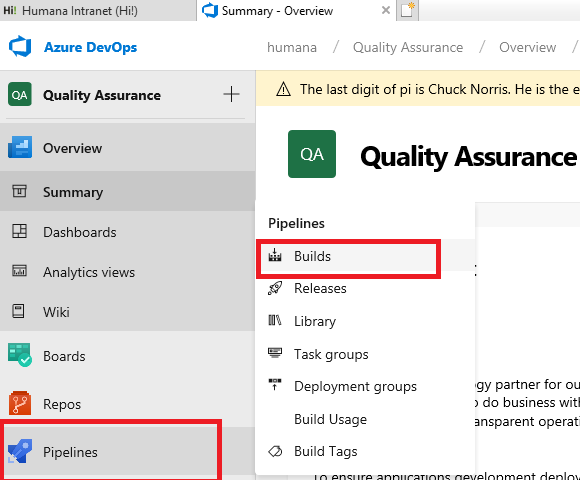
1. Go to the projects, by selecting this link: <https://humana.visualstudio.com/>



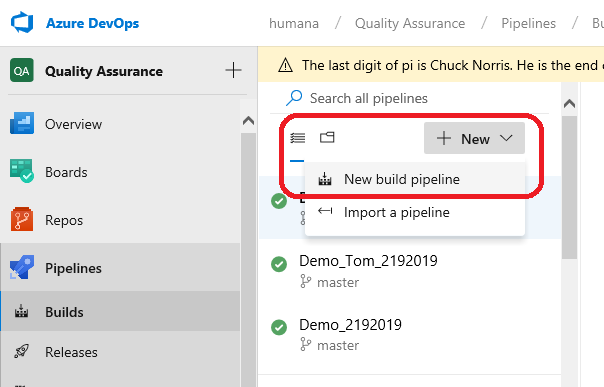
1. Select **Quality Assurance** Project.
2. Selenium Solution repository path –> Repos –> Files –>Choose the correct branch

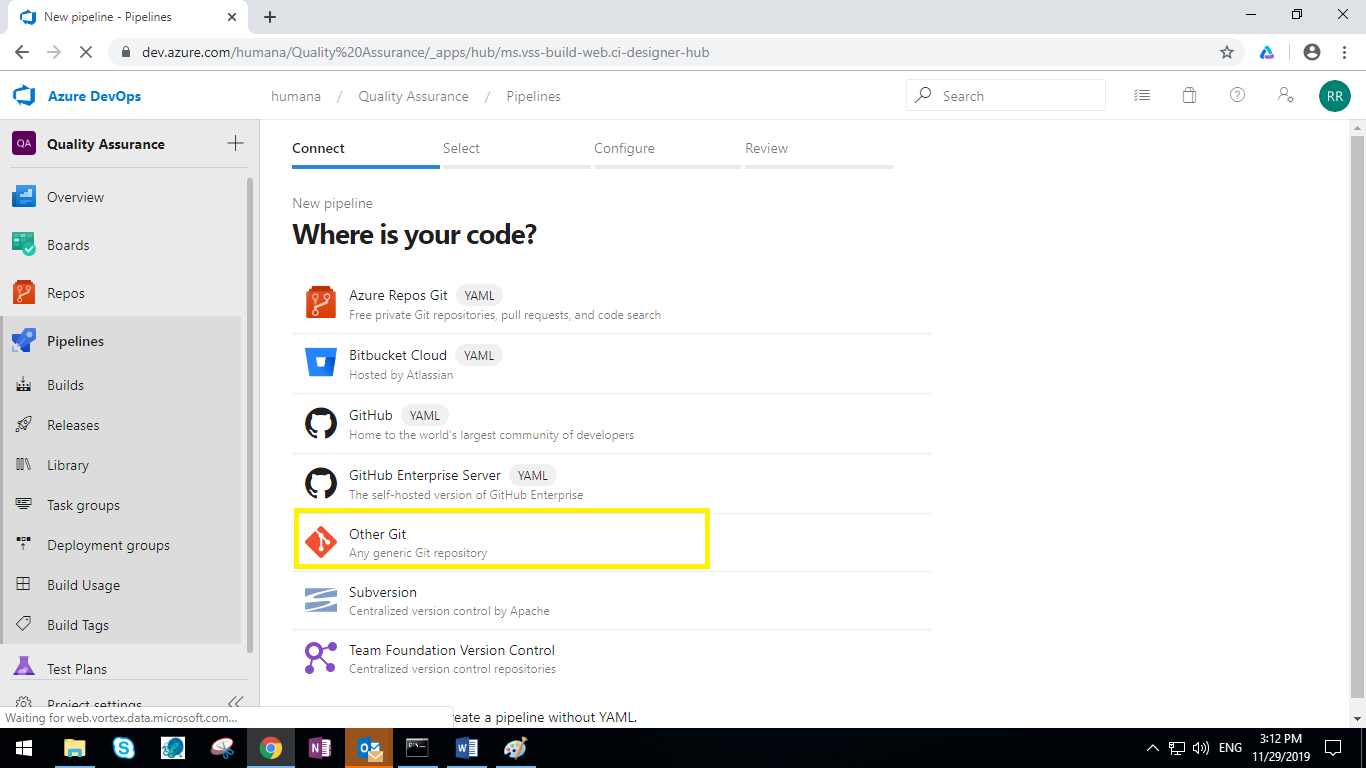


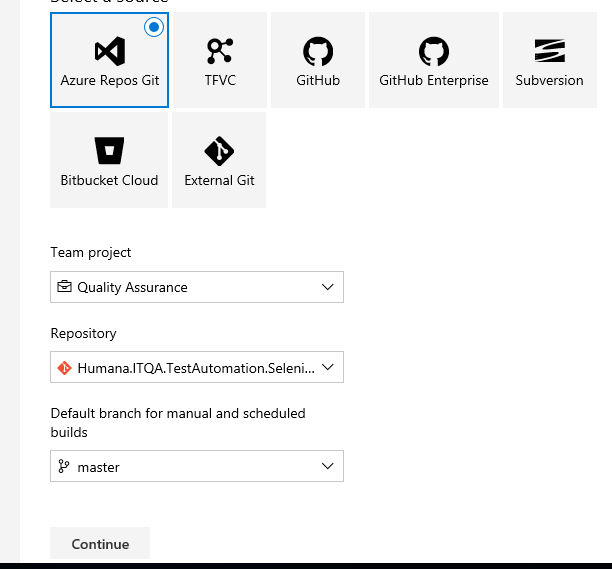
1. Navigate to build creation page. Pipelines -> Builds



1. Click ‘+ ‘button and Select New build pipeline option



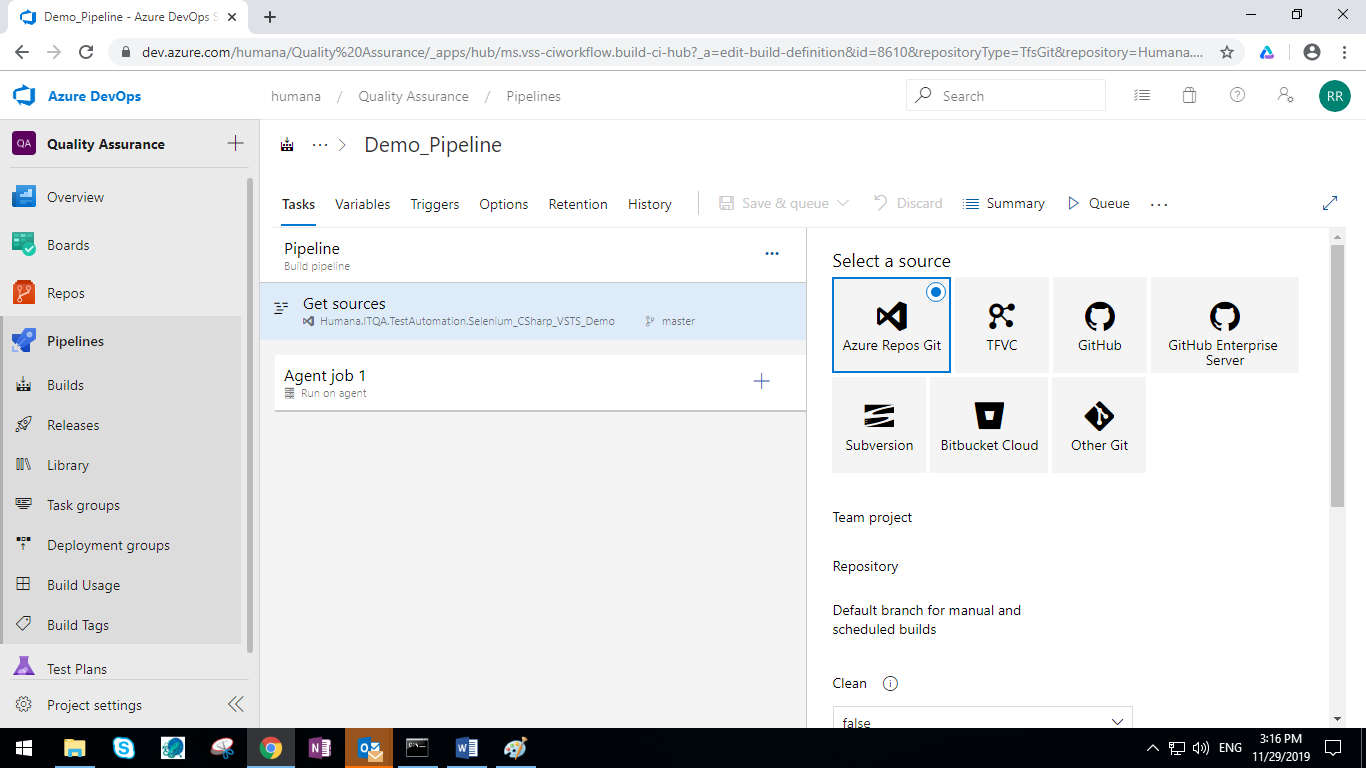
1. Select other git in the below page****
2. Make sure to follow the configurations shown below and click on continue button



1. Select the template as Empty Process

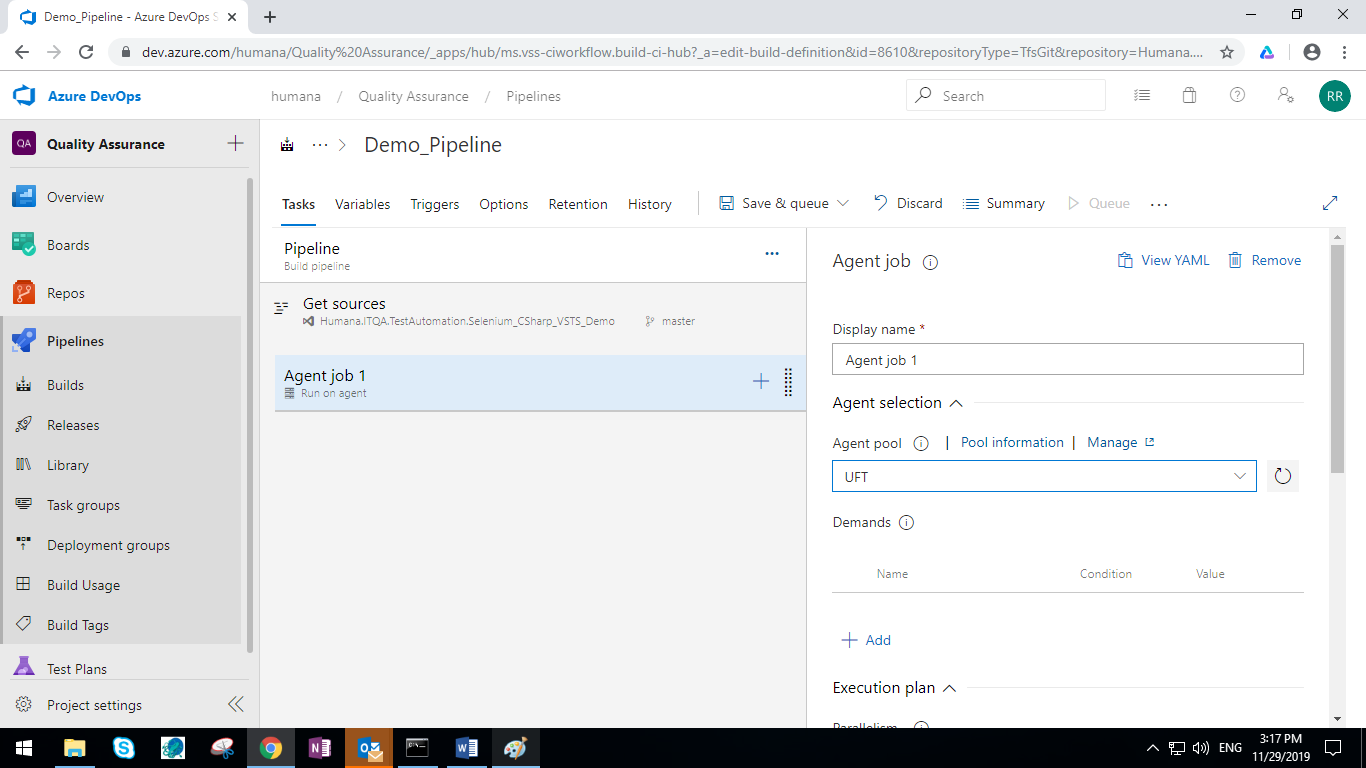


1. Select the **Pipeline** task and follow the configurations shown below.

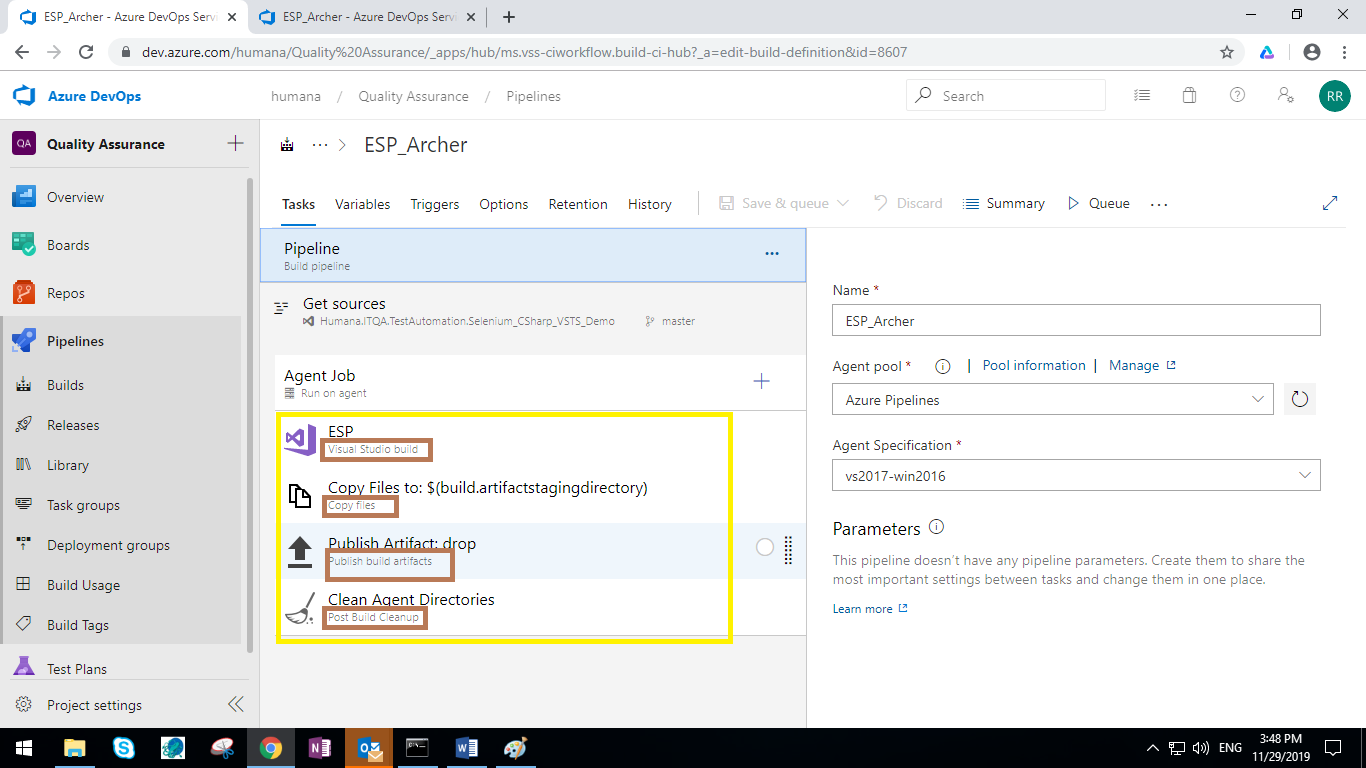


* 1. Click on “Agent Job 1” and do the below configuration

Select your respective “Agent pool” as mentione din the below screenshot

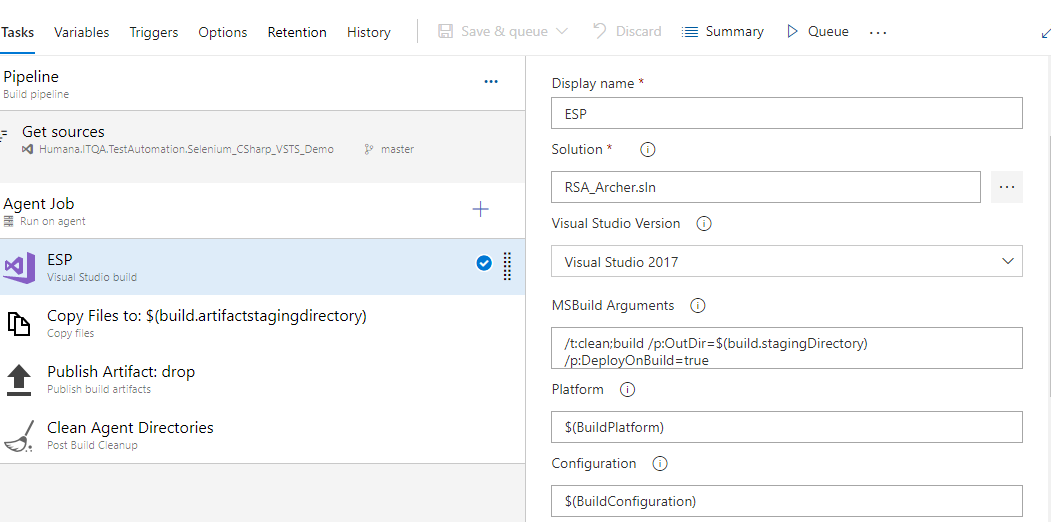


* 1. Click “+” icon in the Agent Job 1 and in the search box in the right Search for below tasks and addto the pipeline
     1. Visual Studio Build
     2. Copy Files
     3. Pubish Build Artifacts
     4. Post build clean up

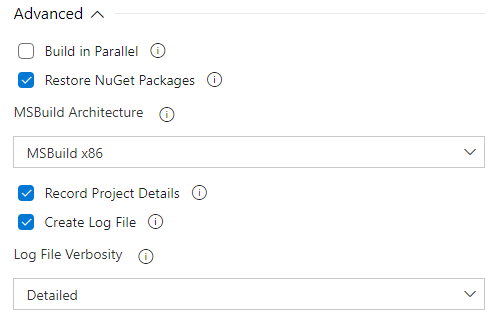


1. Click on “Visual Studio Build” Task and do the below configuration

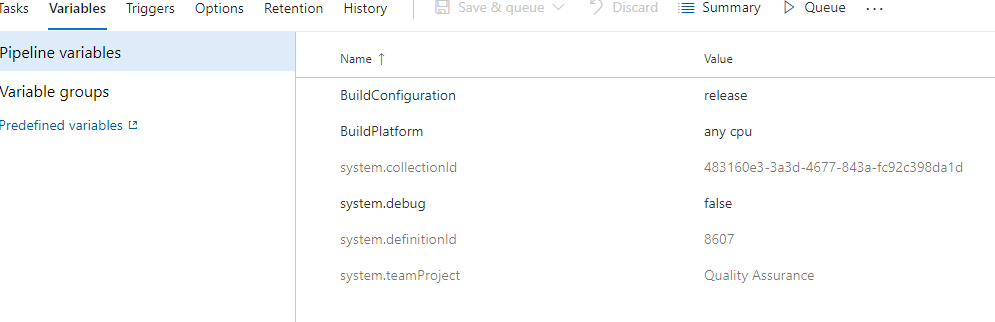
* Display Name: User Defined Name
* Solution: Clock the … and navigate to the solution and select the .sln file
* Visual Studio Version: Select the Appropriate Version in the agent machine
* MSBuild Argument: /t:clean;build /p:OutDir=$(build.stagingDirectory) /p:DeployOnBuild=true
* Platform: $(BuildPlatform)
* Configuration: $(BuildConfiguration)



Under Advanced section

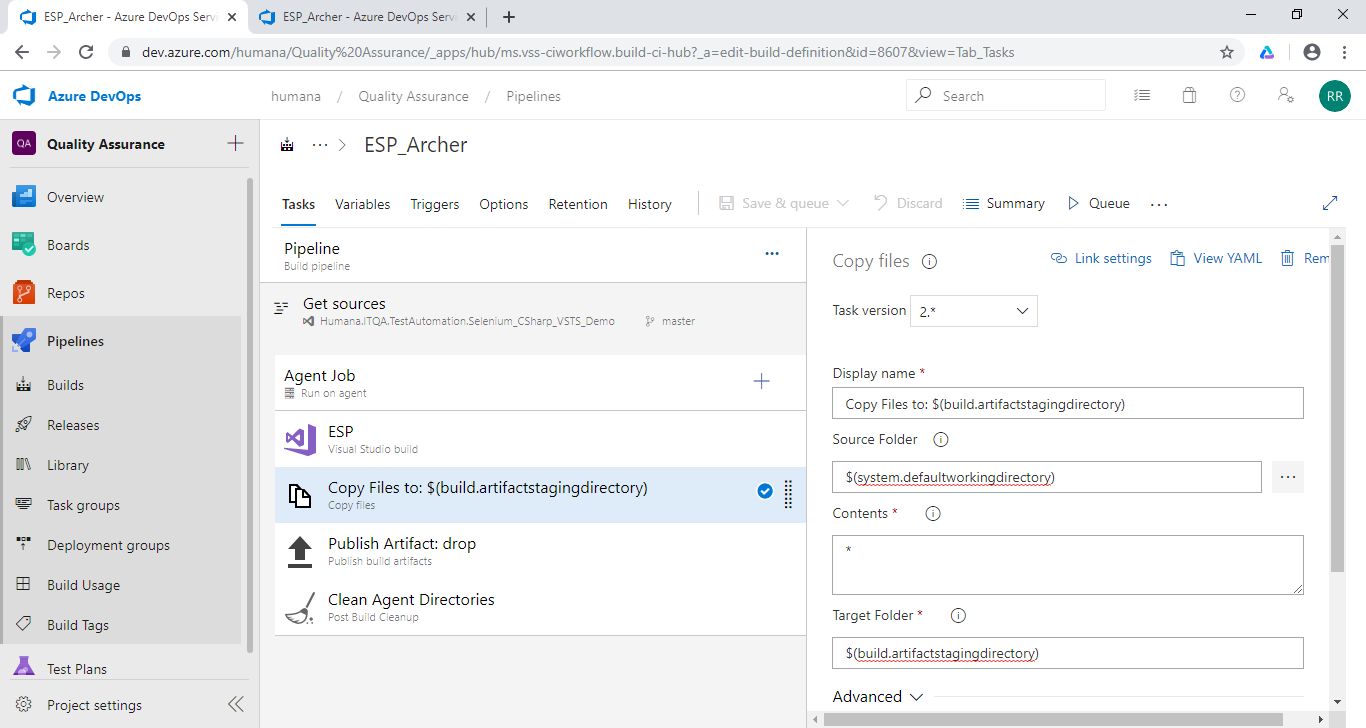


In the Variable Tab do add Buildplatform and Buildconfiguration variables as shown below



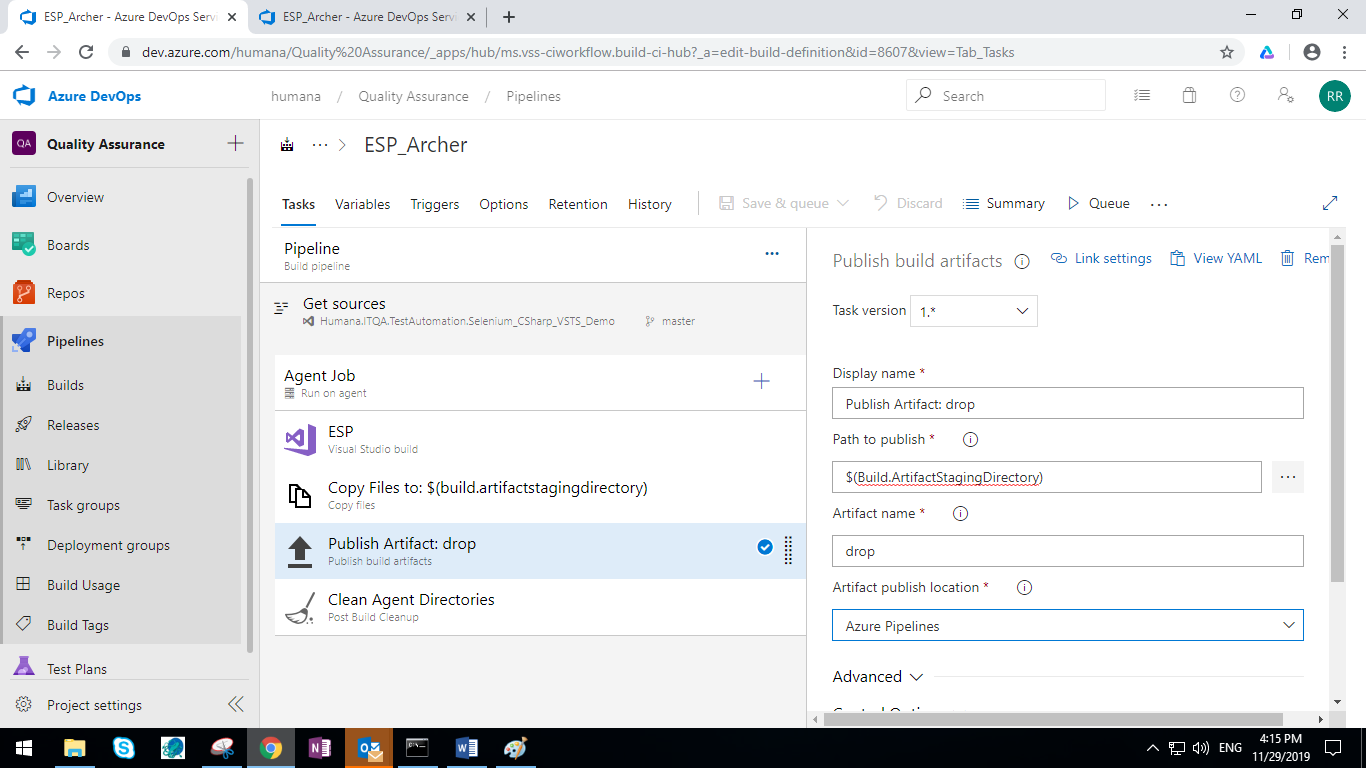
1. Click “Copy Files” task and update the configuration as shown below.

* Display Name: User defined Name
* Source Folder: $(system.defaultworkingdirectory)
* Contains: \*
* Target Folder: $(build.artifactstagingdirectory)



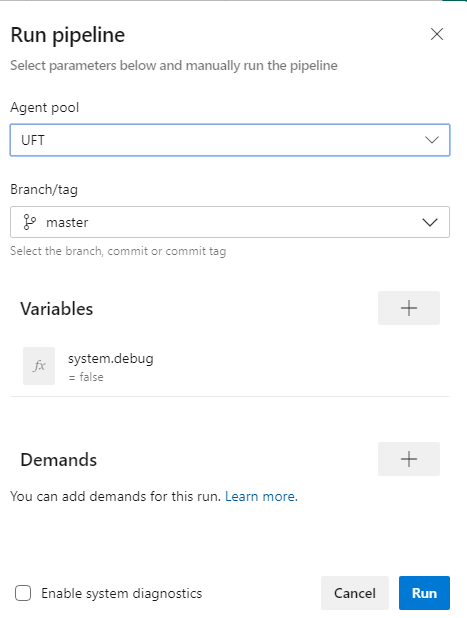
1. Select the **Publish Artifact: Test DLLs (Publish Build Artifacts)** task and follow the configurations shown below.

* Display Name: User Defined Name
* Path to Publish: $(Build.ArtifactStagingDirectory)
* Artifact Name: user defined Name
* Artifact Publish Location: Azure Pipeline

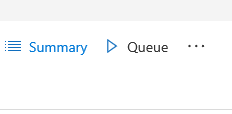


1. No configuration updates in “Clean Agent Directories” Task
2. Click **‘Save and queue’** Button, Select “**Agent Pool”**  and click **Run** Button

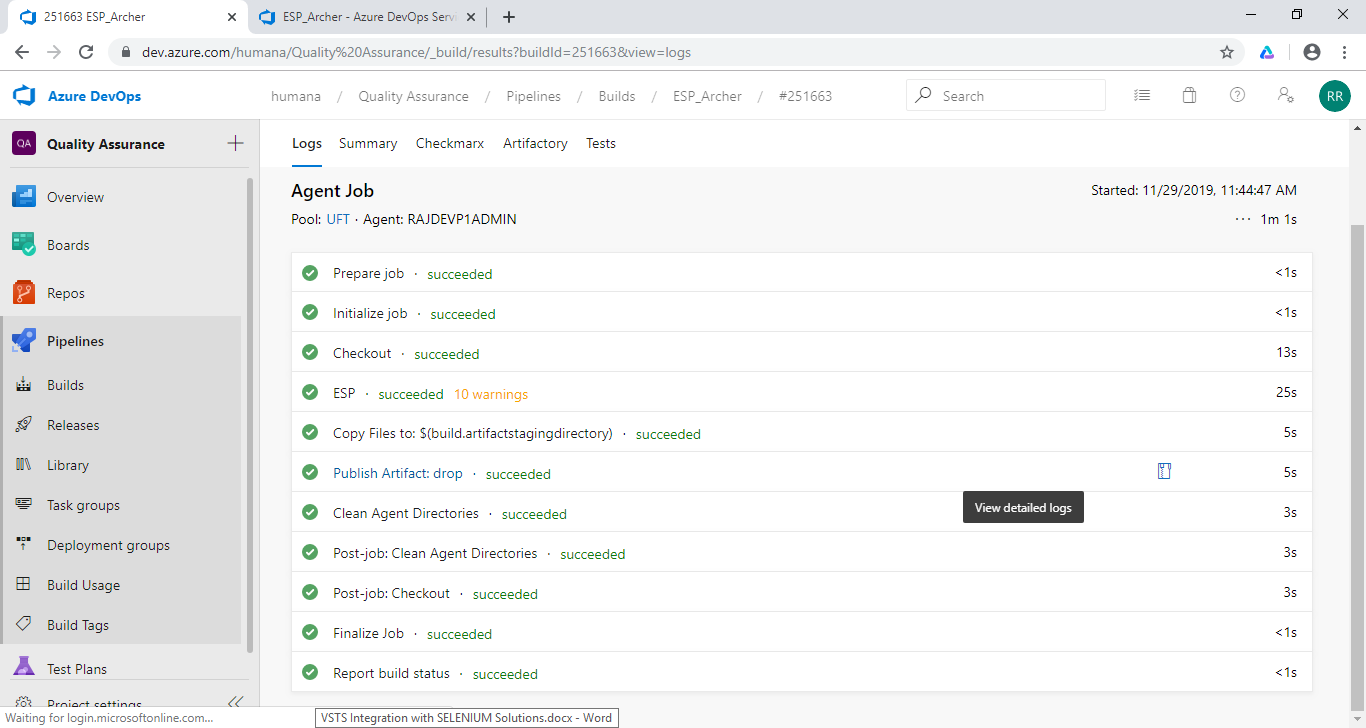




1. Or Queue the build by click on queue if the build is already saved as shown below.



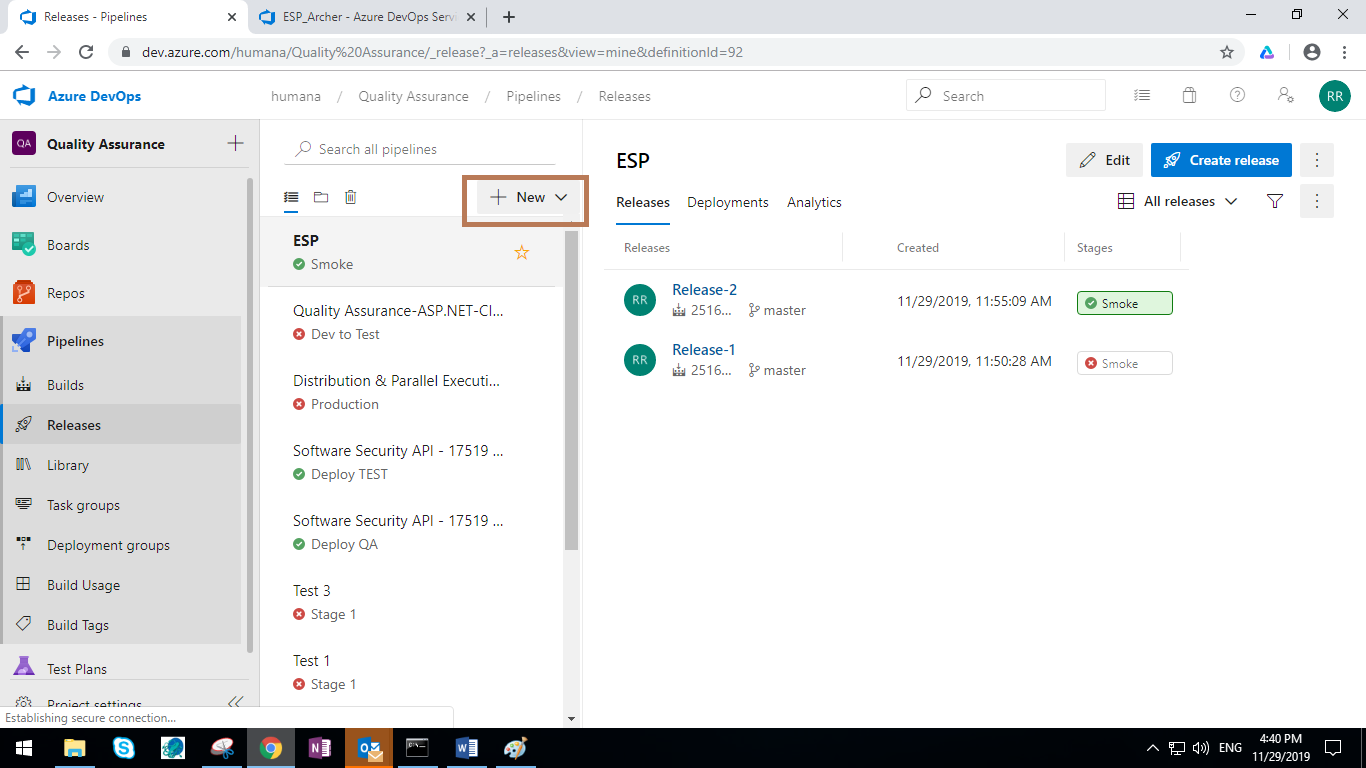
1. To view the Log Details click the log tab

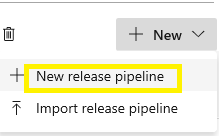


1. Once build is completed, we will receive a build notification to our email with the status of either succeeded (or) failed.

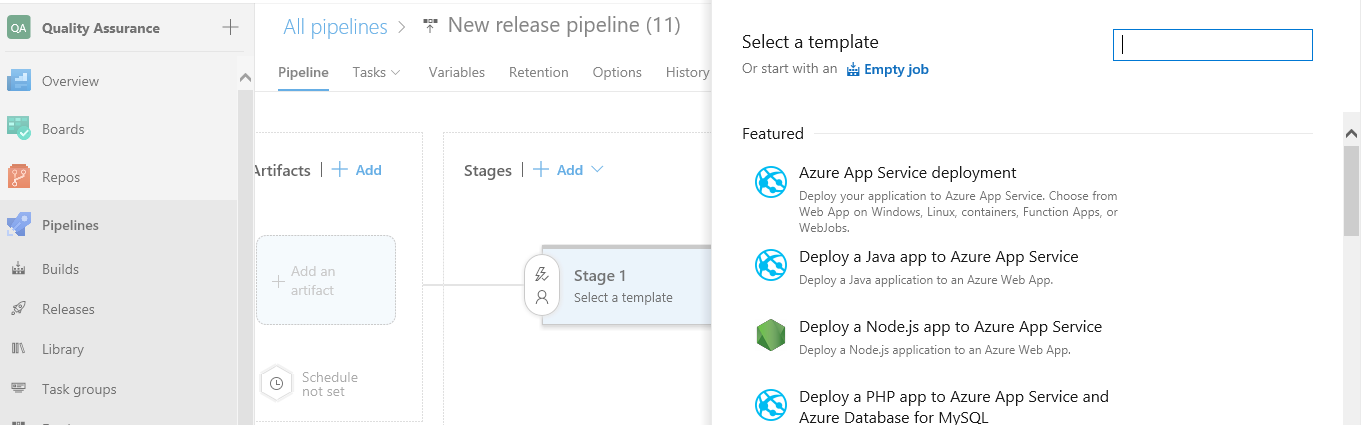
# **Release Pipeline Creation**

1. Once the build is succeeded, we have to create a new release pipeline.
2. For that, go to the releases, click on ‘+’ symbol and select the create release definition.

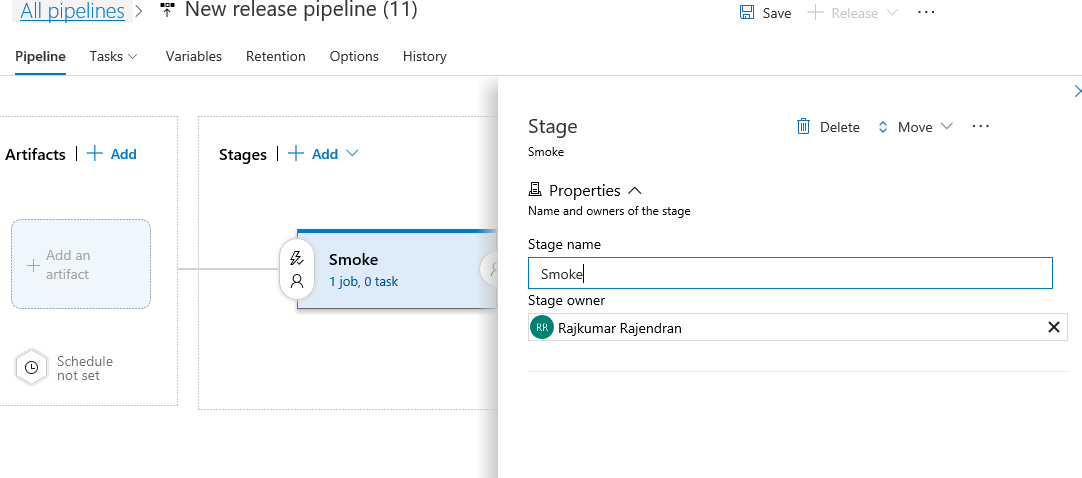




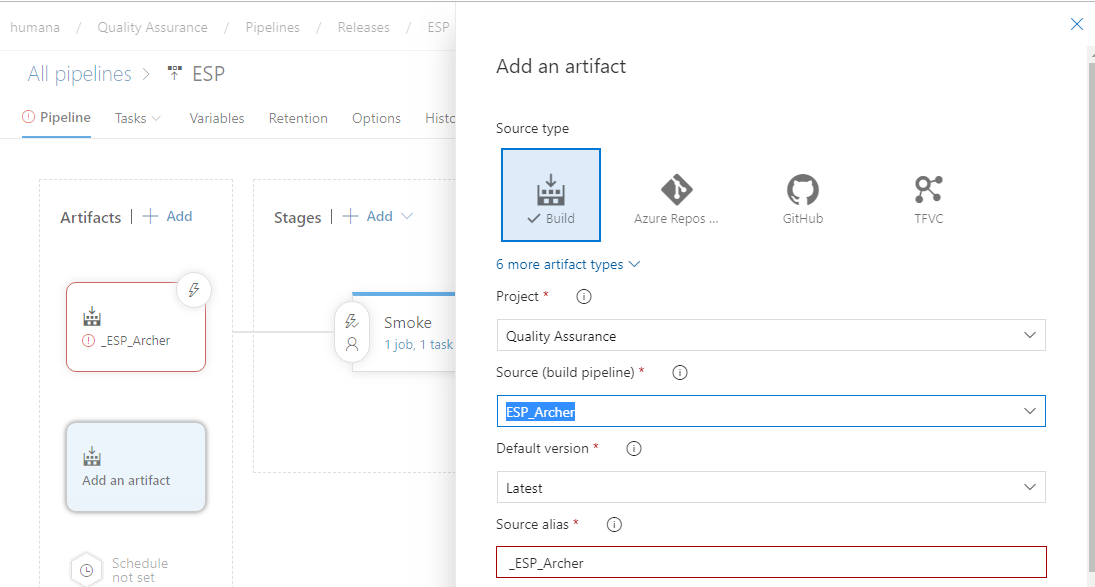
1. Select the template as an empty process.

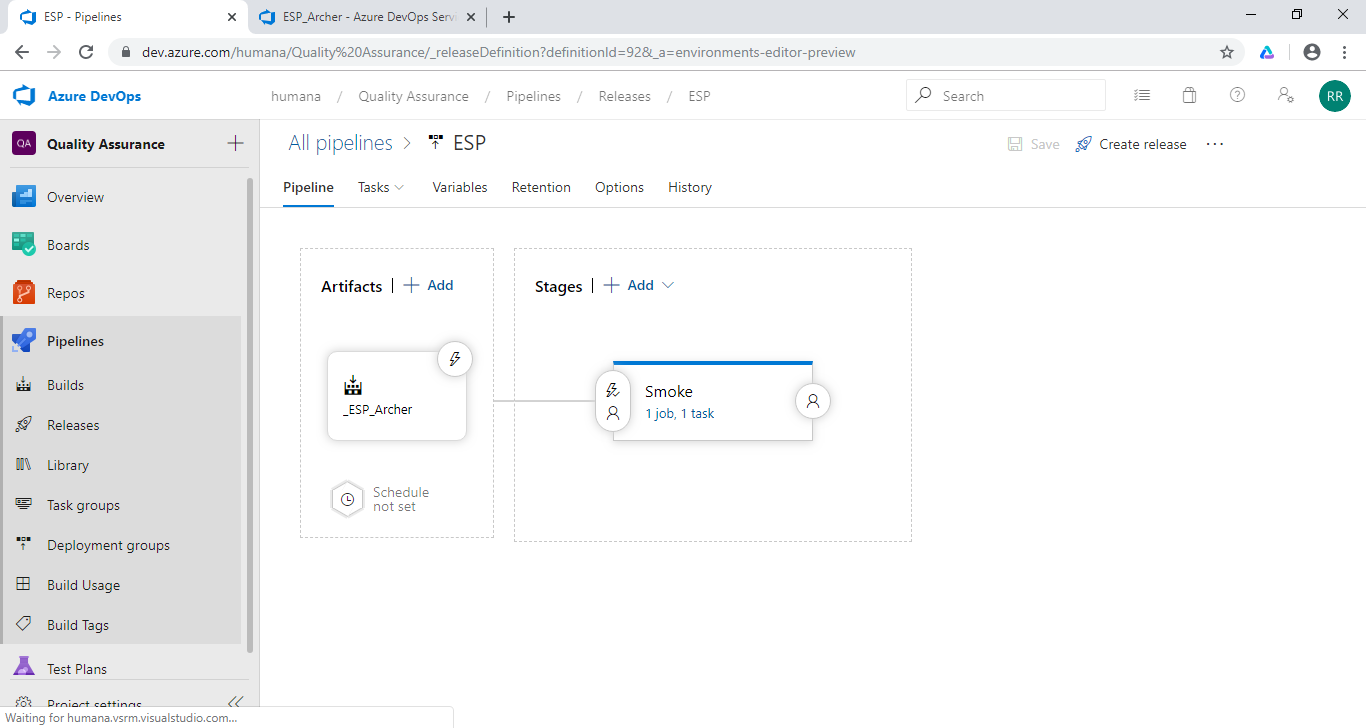


1. In the next step, we have to provide Stage name based on the requirement.

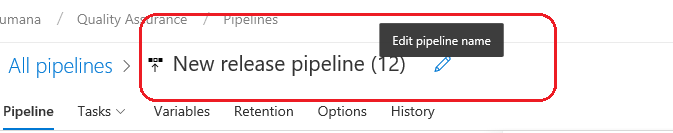


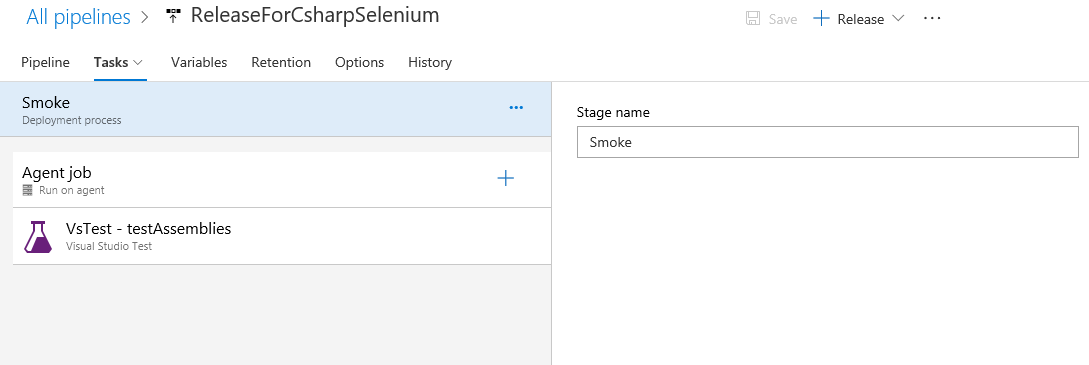
1. Click on the add artifact and choose the source field. Make sure to provide the source with the same name as given in the process of build.



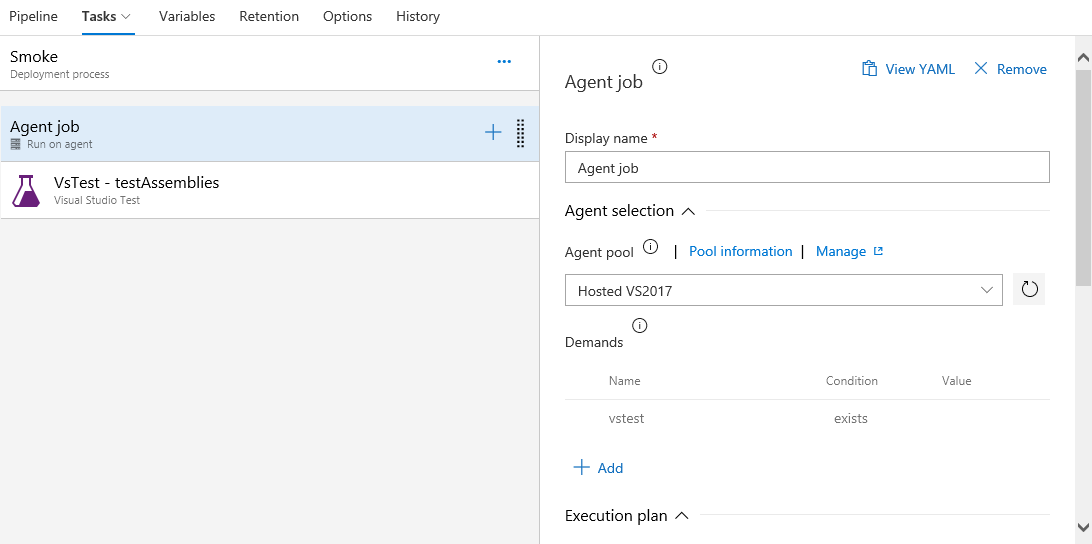


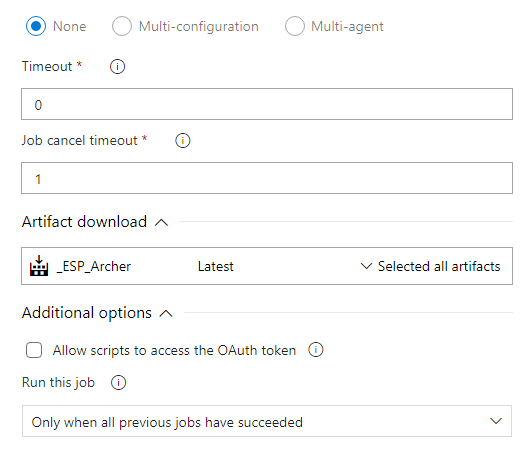
1. Using the Edit option change the Release pipeline name.





1. Go to the tasks by selecting the tasks option, choose the agent job and follow the configurations shown below.

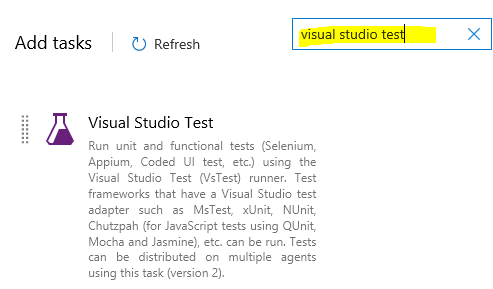




1. Click on ‘+’ symbol at the agent phase to add the task

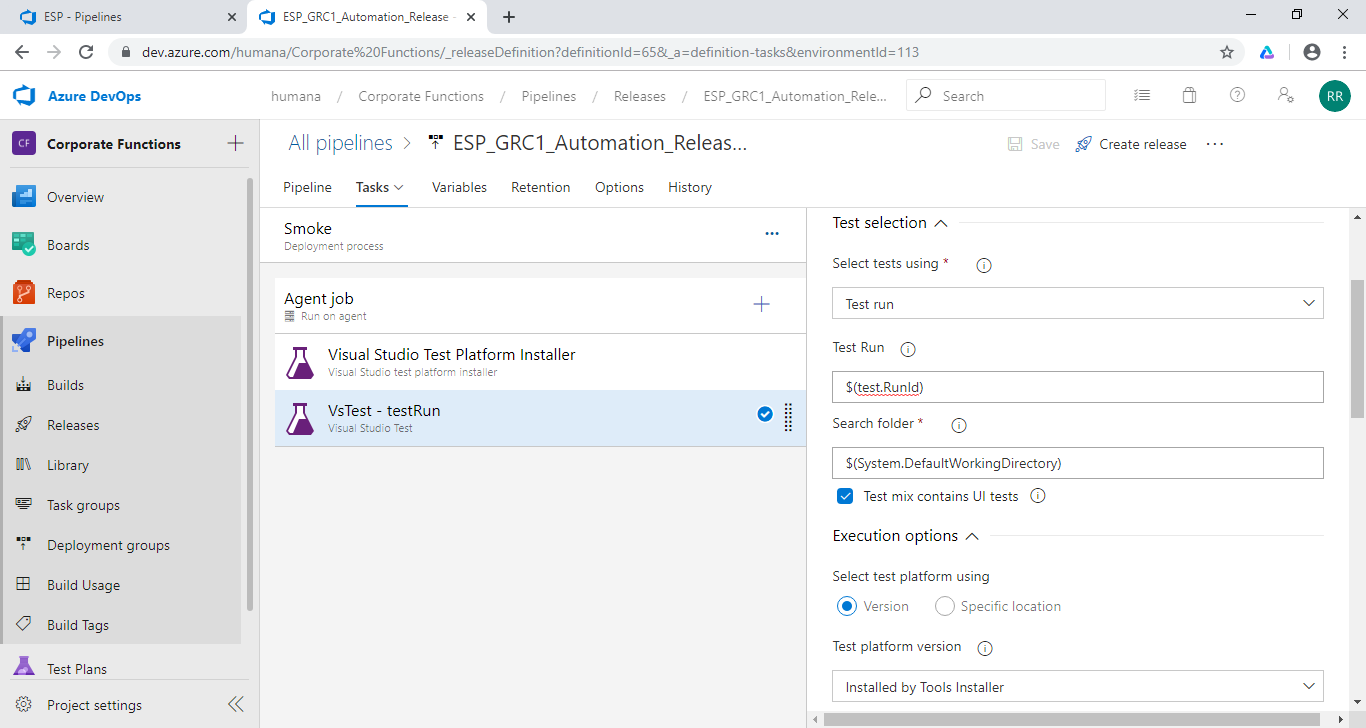


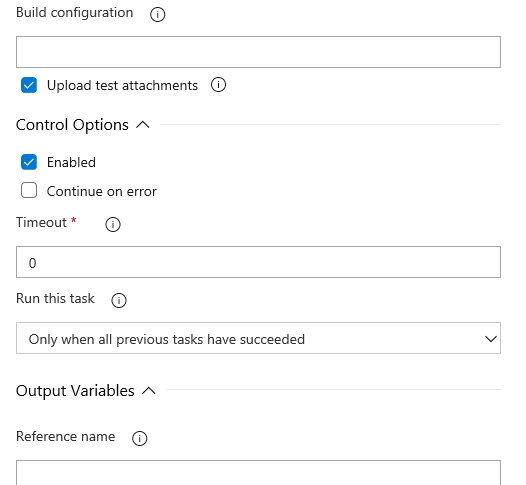
1. Select the **Visual Studio Test Platform installer** as a task and click on add button to add it to the tasks. Don’t make any changes in the task
2. Select the **Visual Studio Test** as a task and click on add button to add it to the tasks.



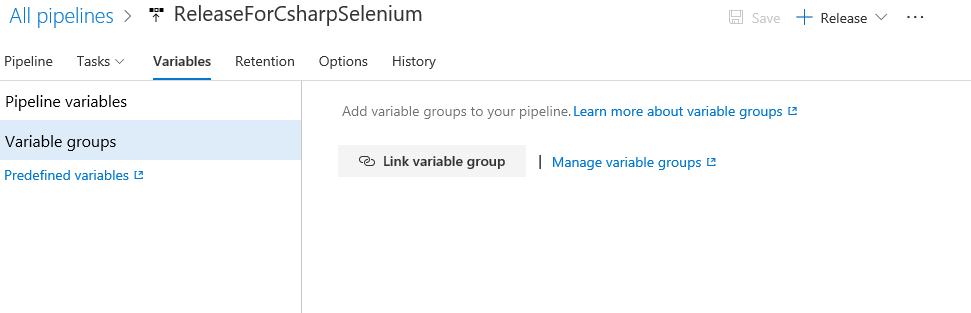
1. Select the **VsTest – testAssemblies (Visual Studio Test)** task and follow the configurations shown below.

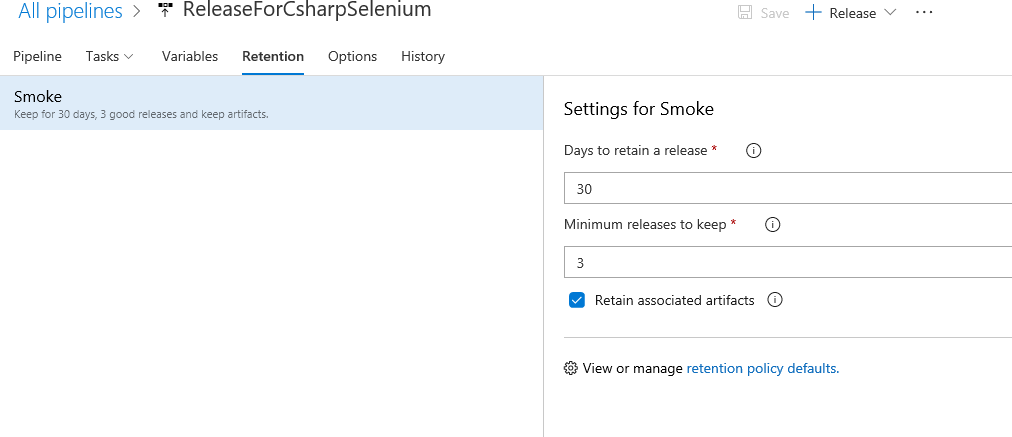
* Select Test Using: Test run
* Test Run: $(test.RunId)
* Search Folder: $(System.DefaultWorkingDirectory
* Select Test mix contains UI tests
* Test Platform Version: Select “Installed by Tools Installer”

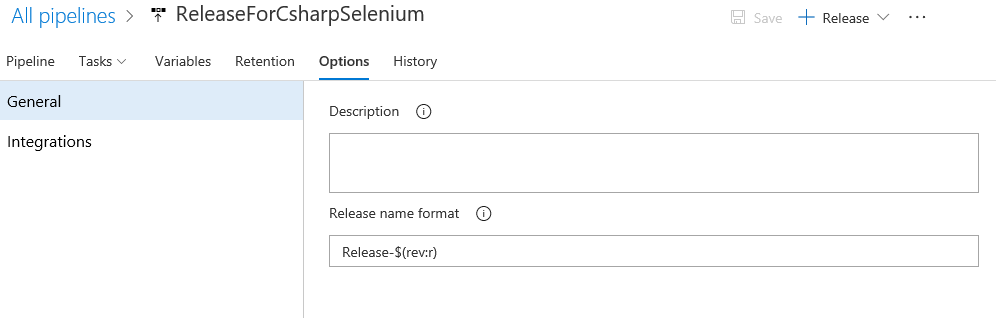




1. Navigate to all the remaining tabs and follow the configurations shown below.





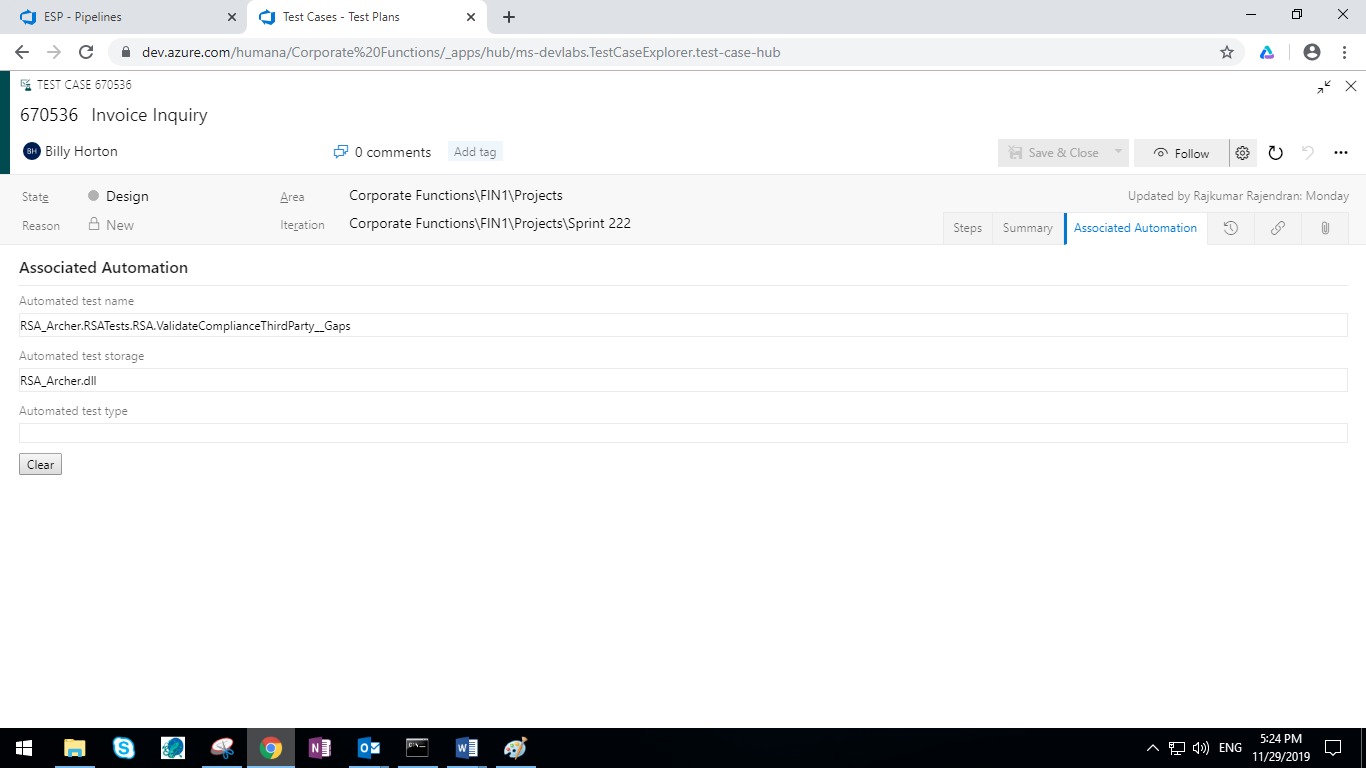


1. Click Save

**Execution From Test Plan:**

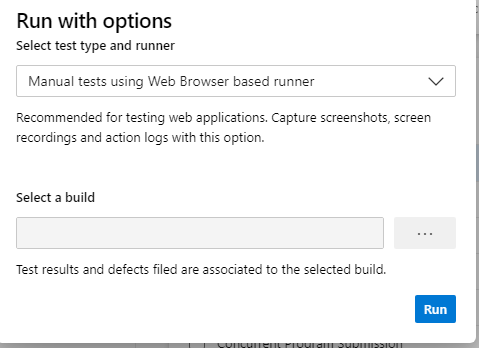
Pre Requisite: Automation test script should be mapped to manual test case using Visual Studio 2017.

1. Checkout solution from Azure devops Repos branch and open in Visual Studio 2017
2. Seelct the Test from Test explorer pane in left side
3. Right click on theTest “Associate Test” option
4. Provide the Test cases ID in the popup, Search and Save.
5. To confirm the Associate please navigate to the Test cases in Azure, Goto Associated Automation Tab. Verify the updated details as shown below
6. Make sure Agent Machine is up and Running
7. If executing from Grid please bring Selenium node up and running

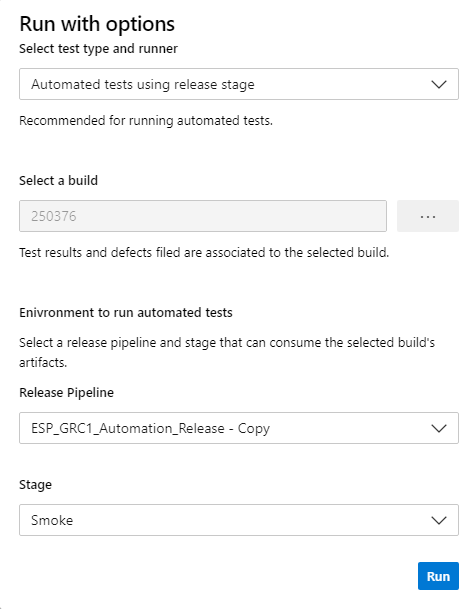


**Execution Section:**

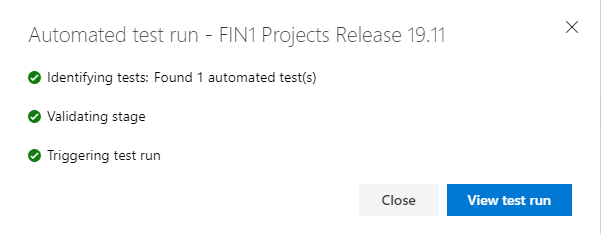
1. Go to Test plan
2. Navigate to the Test case which you need to Execute
3. Selct the Checkbox of the corresponding test cases
4. Click the three Vertical dots and Mouse hover to and Click Run options, The below popup will come



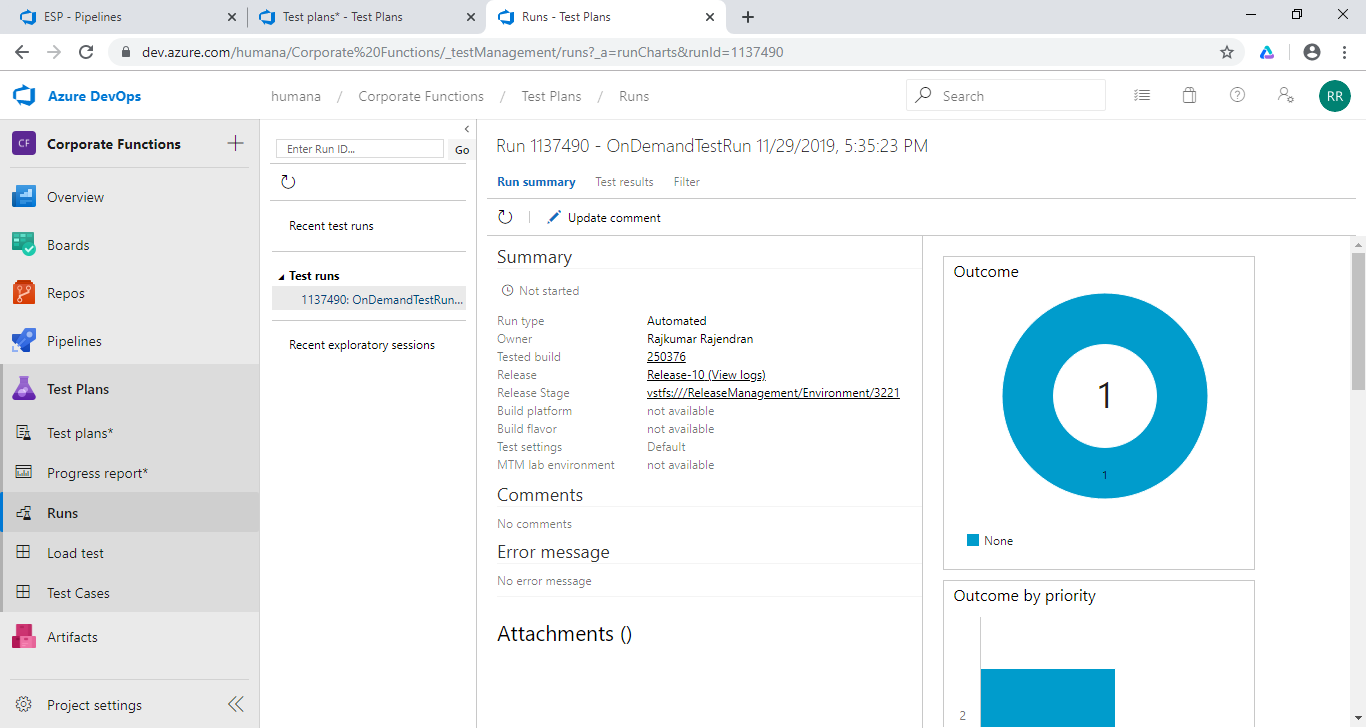
1. Select as mention below
   1. Selct Test Type and Runner: Automated Tests using Relase Stage
   2. Select your Build which crated in the above section
   3. Select Releae Pipleline annd Stage and click Run. Refer below creenshot for reference



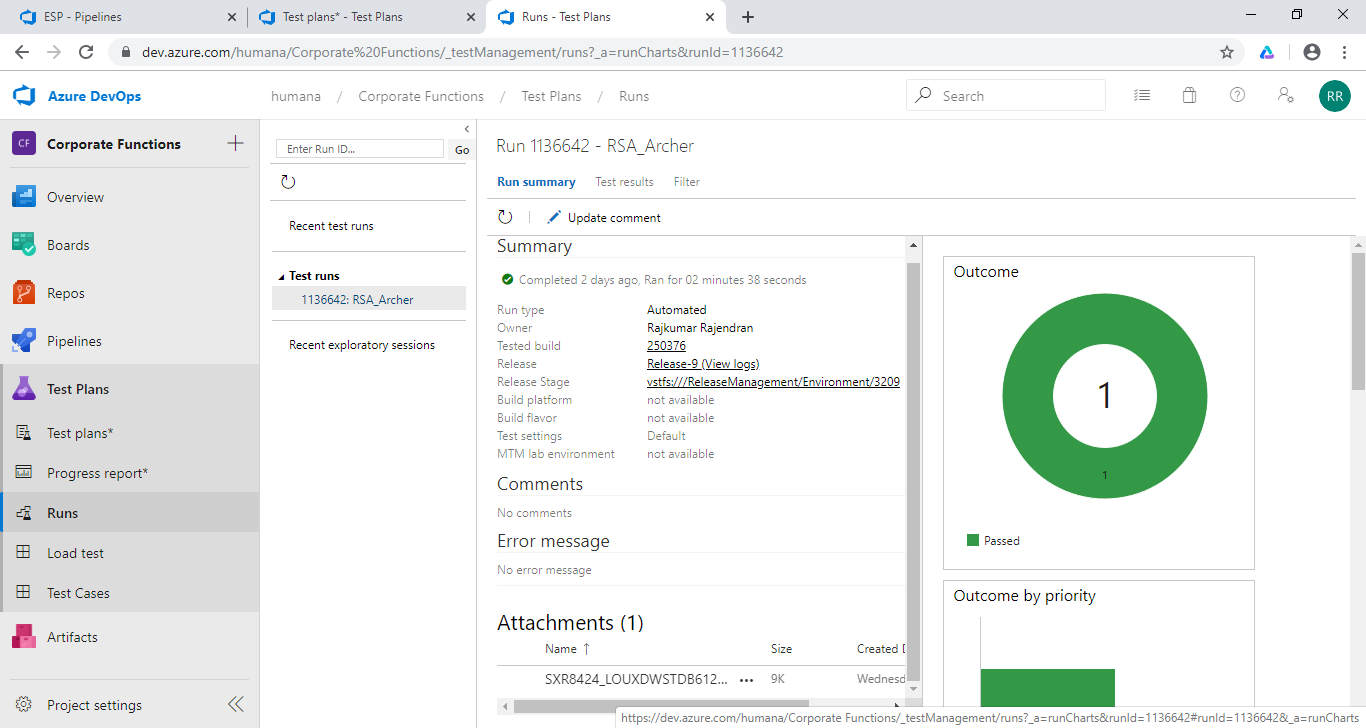
1. Click Run



1. Click View Test Run



1. Post Successful execution the Run deails will display like below



1. Click View Logs

