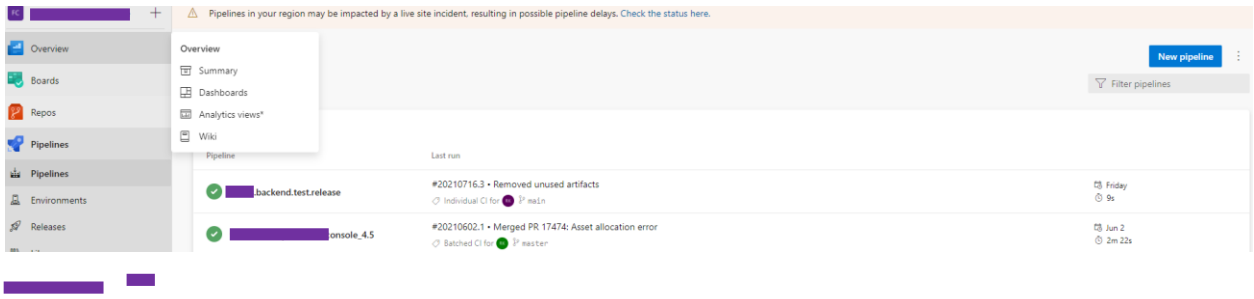
Setup Azure Build Pipeline, Release Pipeline, and Stages

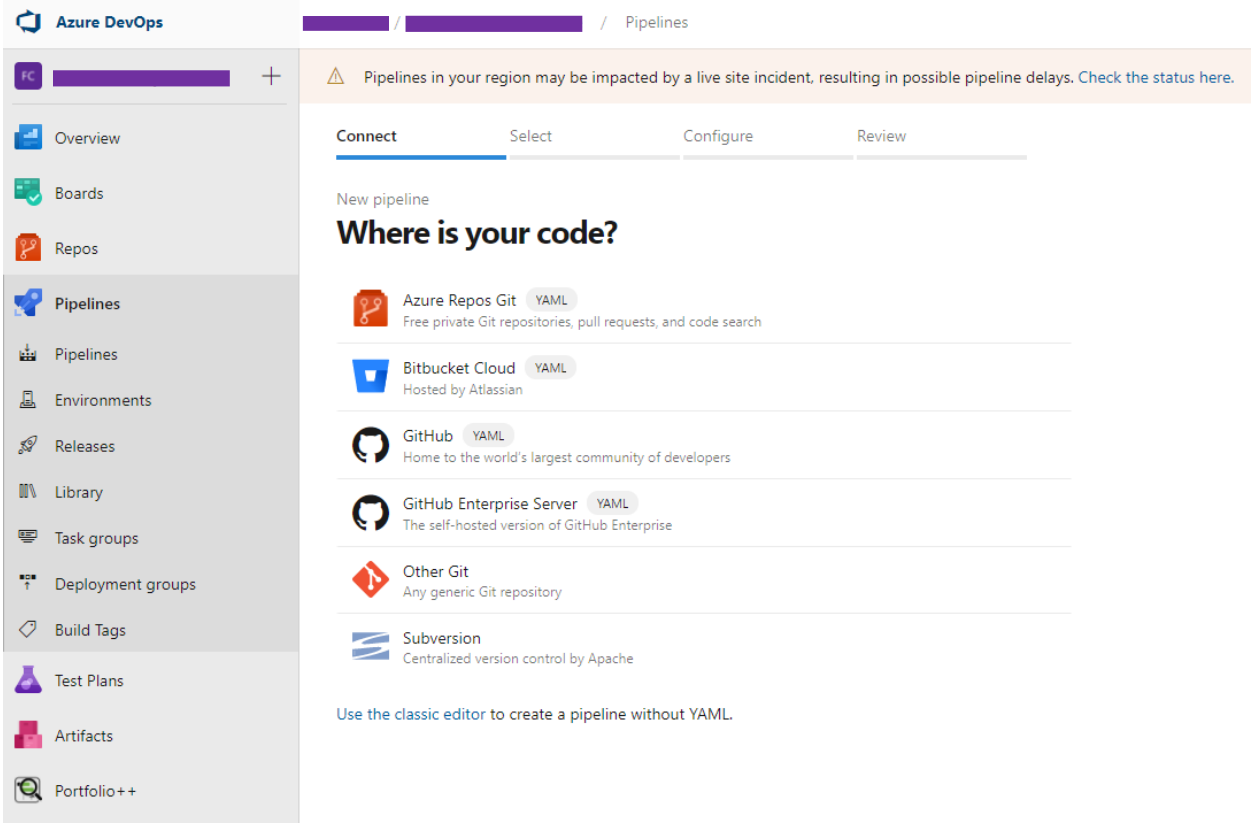
Step 1: **Create Build Pipeline**.

Click on the Pipelines tab in order to create a new pipeline, Once, you invoke Pipelines, you can see a button **New pipeline** to create a new pipeline on Azure DevOps.

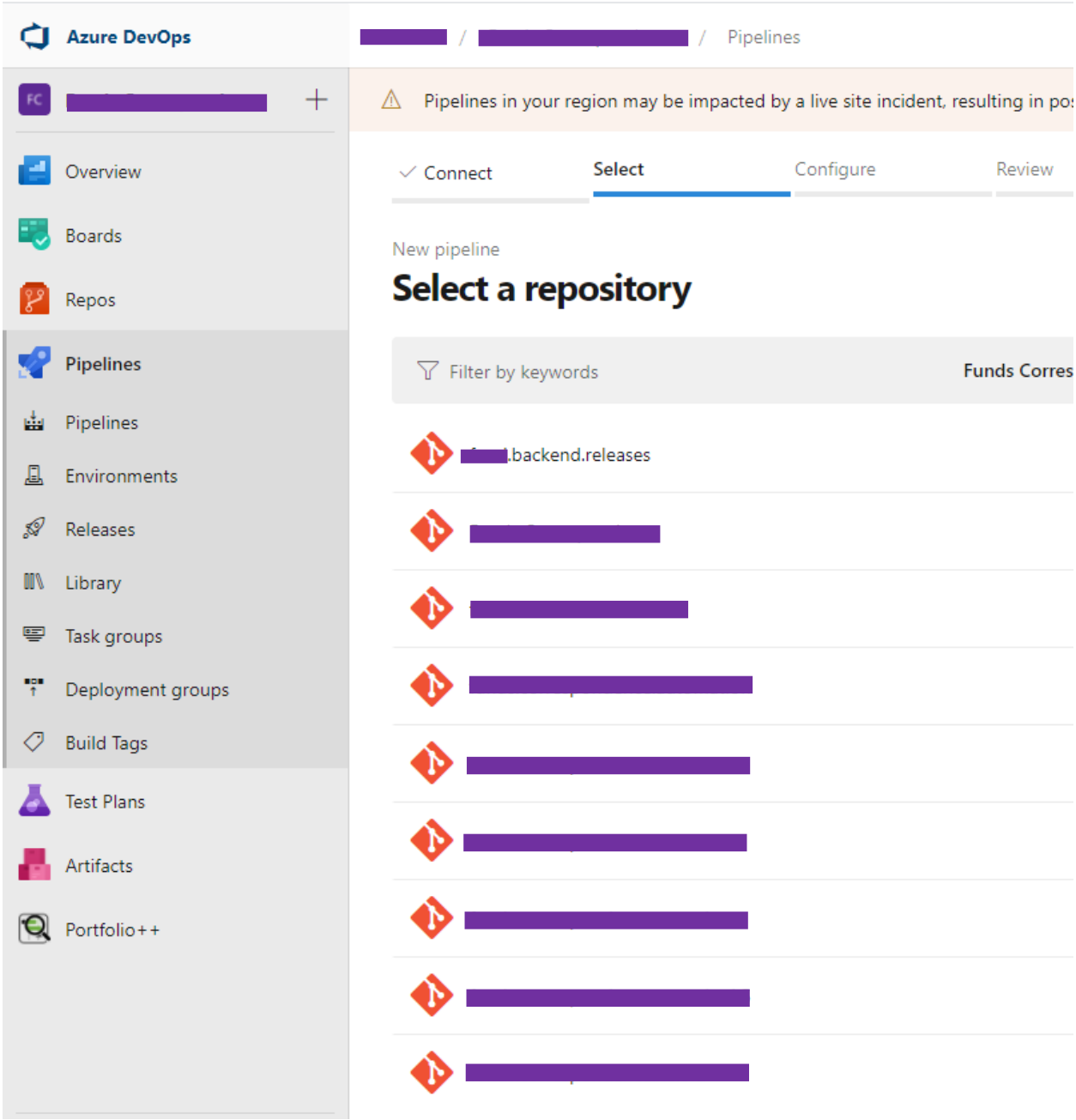


Configure a newly created **Pipeline**.

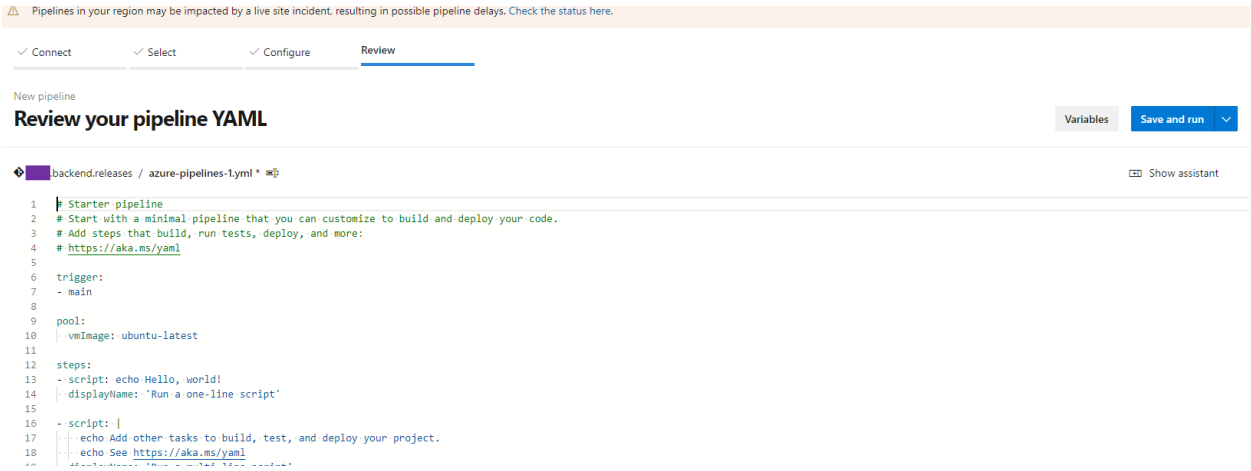
a. Select repository source e.g. Azure Repos Git.



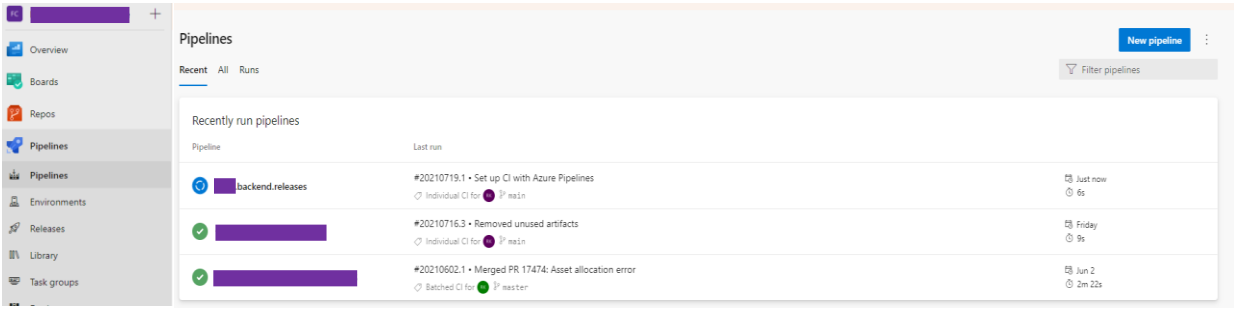
b. Select the repository name for the pipeline.

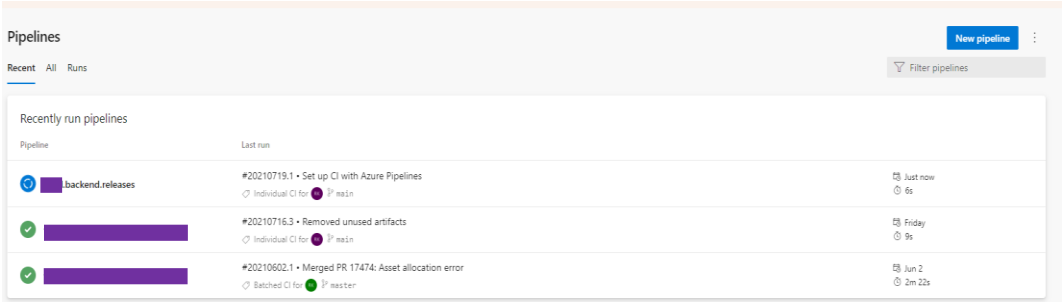


c. Run/Save and run the Pipeline in order to make it active.



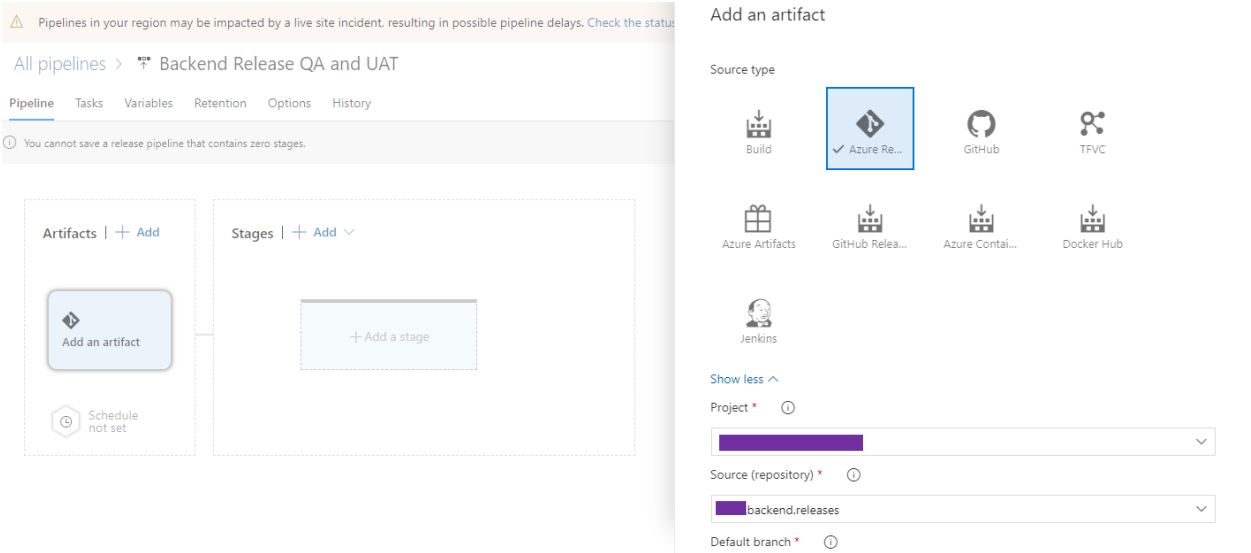
Once the pipeline is created you can view it under the Pipelines tab. The below screen shows newly created pipeline is still executing.





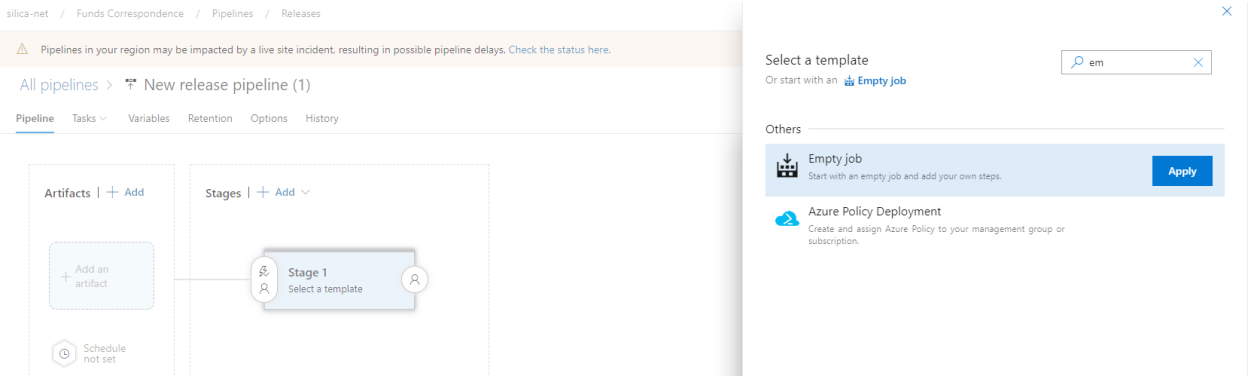
Step 2: **Create** a **Release pipeline and Add an artefact.**

Click on the + Add button under Artifacts. Select Azure Repos Git, the select project as repository **backend.releases**.



Step 3: Add stage

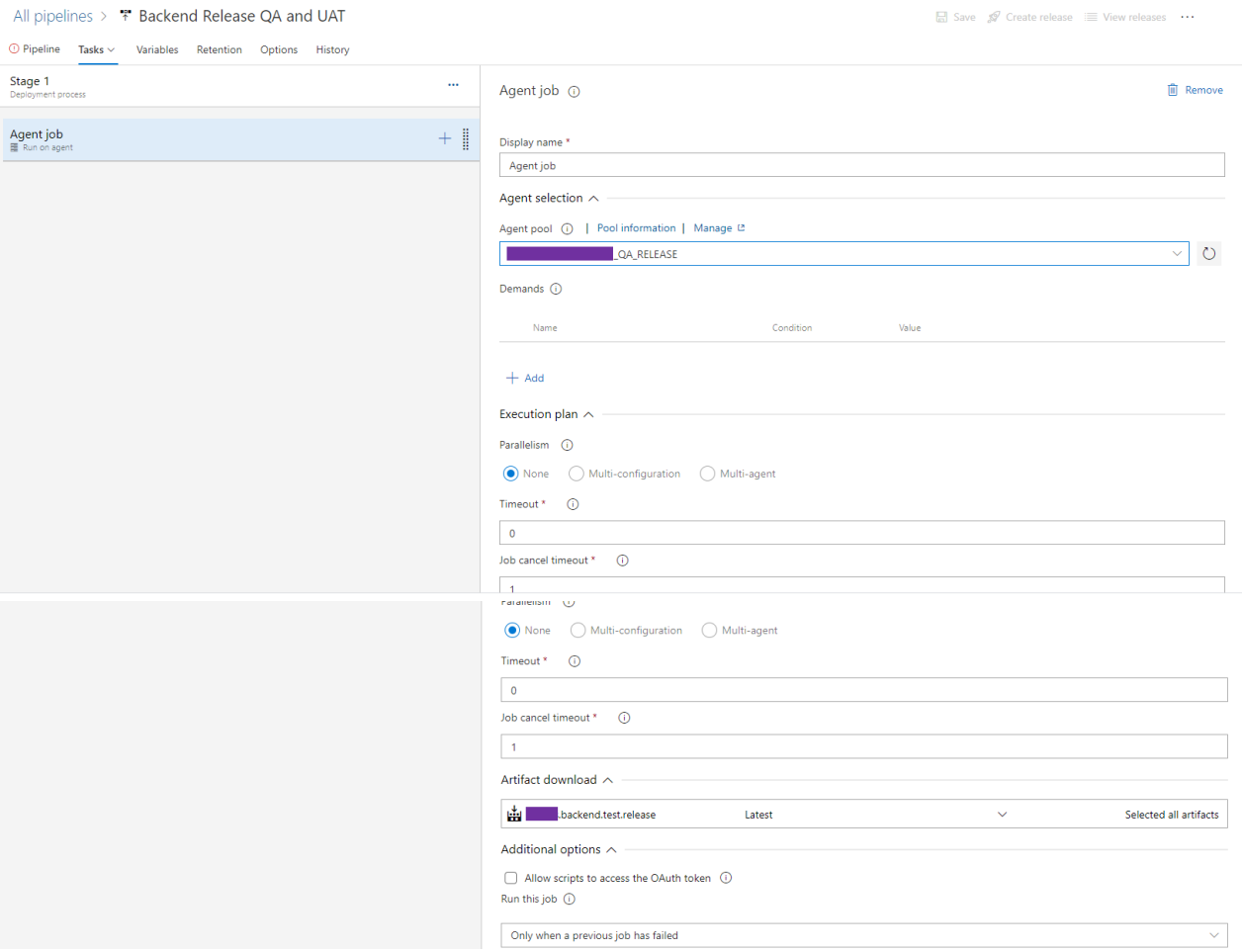
Click on the + Add button under the Stages section. Once you click on Add button under the Stages section it asks you to select a template, and search Empty job, once you find it click on Apply button.



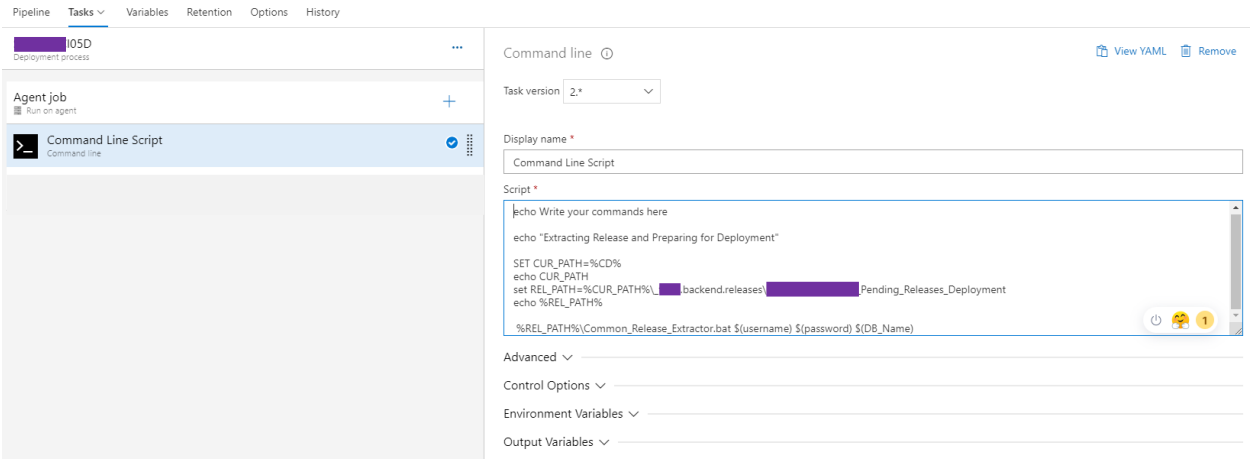
Step 3: Add a task for a stage.

Once you added a Stage, you can see a task link on the Stage box. Click on the task button. Once, you click on the task button, you can see the Agent job.

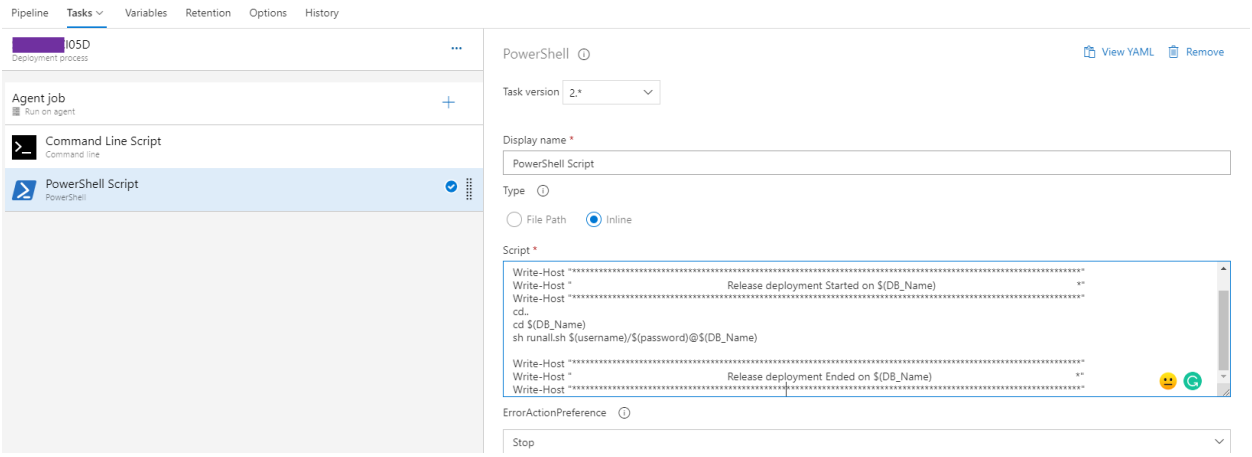
a. Configure Agent Job.
   Click on + to add an agent, you can see the below screen to configure a specific agent pool for the Agent job.



b. Add first tasks to a Stage.
Once you added the agent job, you can create a + button to add a task, and search the command line, once you find the command line, hover the mouse on it, then get add button, and click on add button.
Add the following command to prepare the deployment script.



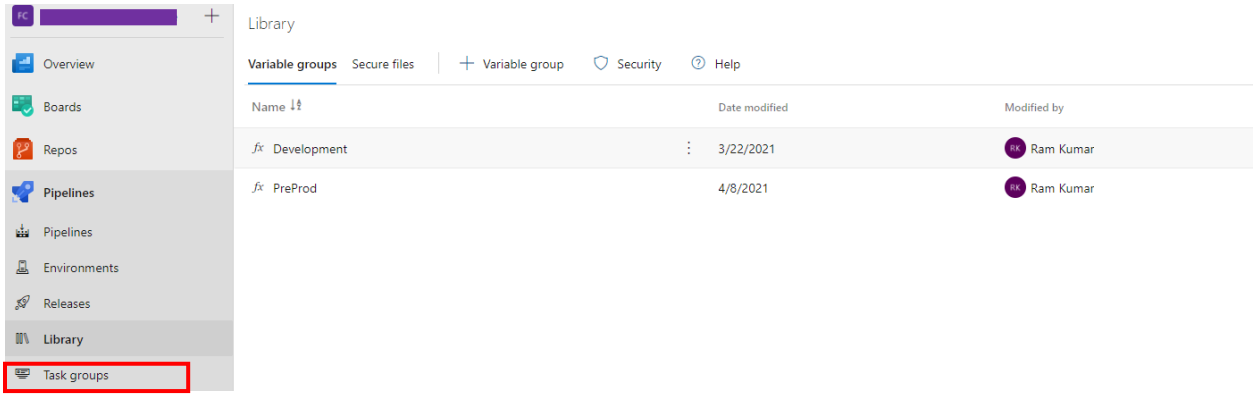
c. Add the second task, like the first task, click on the + button to add the second task, and search PowerShell, once you find it add it.
Add the below command in a script to execute the shell command.



Step 4: - Add variable group.

a. Create a Variable group.

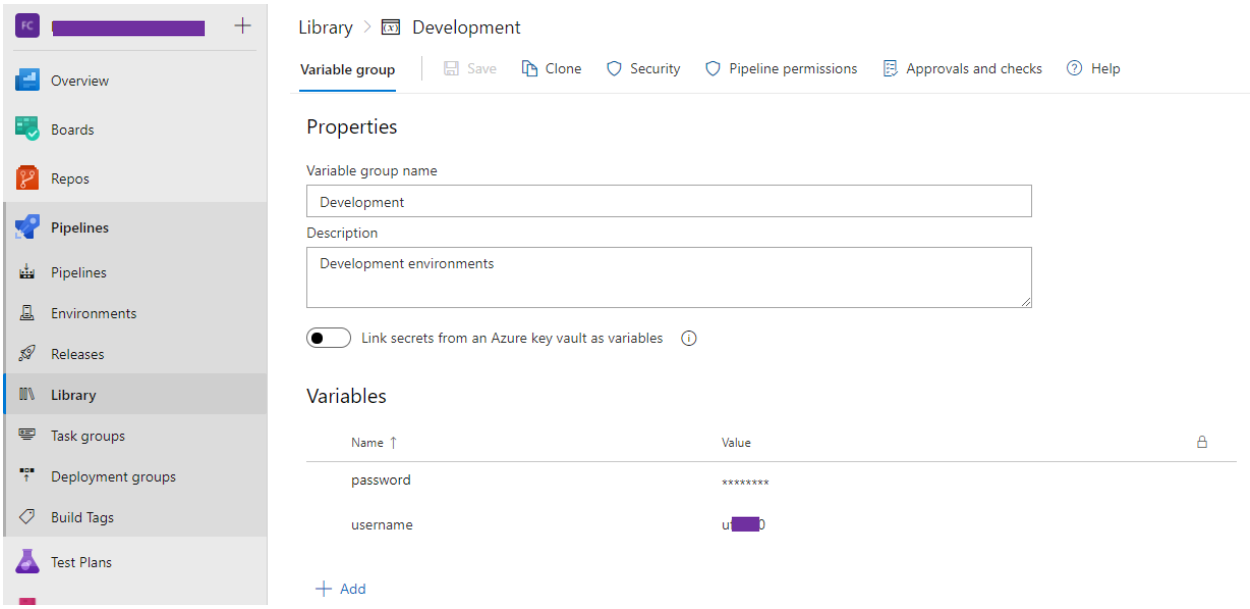
Click on Library (Shown in the red box in the below image) under Azure DevOps. Once you click on the Library tab you can see + Variable group, click on it in order to create a new group. The below screen shows that two variable groups **Development** and **Pre prod** are created.



b. Add variables in a variable group.
Once you added a variable group, click on the variable group name. Once you click on the variable group name you can see another tab to add a variable. To add a variable click on the + button under the Variable tab.

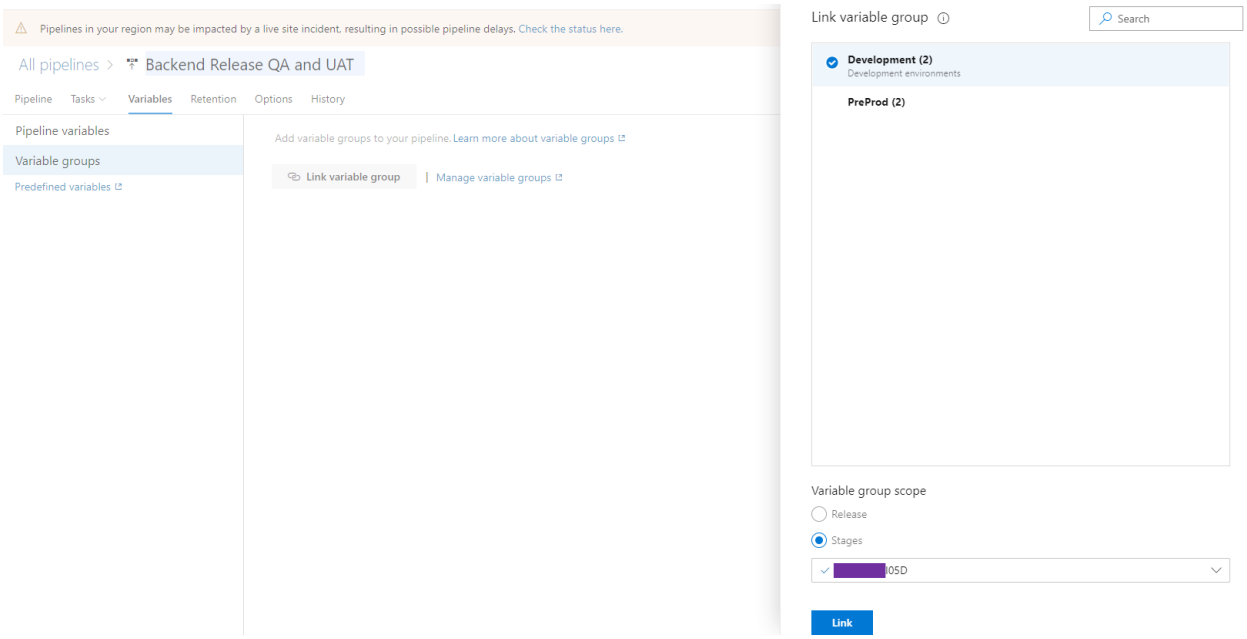
The below screenshot shows that username and password variables are added to Development variable group.

Note: for the password when you hover on value, you can get the lock option, if you click on the lock option password will be not visible. Once you have locked it you can't open it.



c. Link a variable group to a Stage.

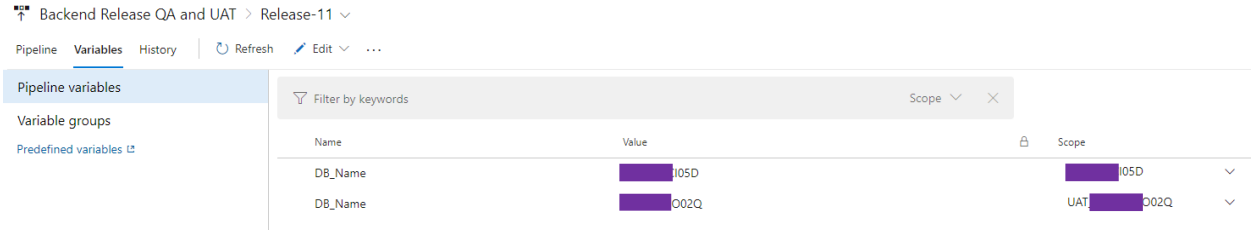
Click on Variable tab of Pipeline, you can find an option to link a variable group to the Stage.



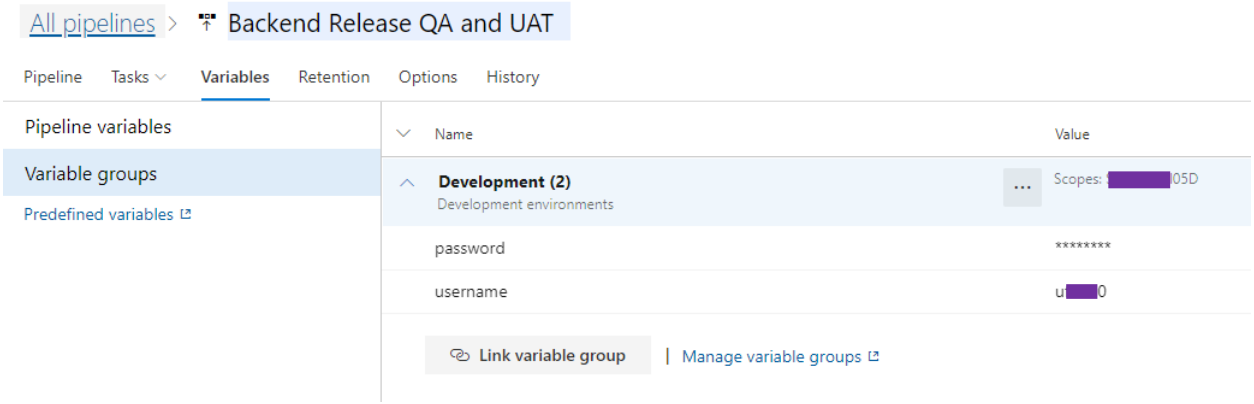
d. Pipeline Variable.

Pipeline variable can be created under Variables of the specific pipeline. Below screen shows that two pipeline variables created with same for with different scope. Scope represents

different Stages.

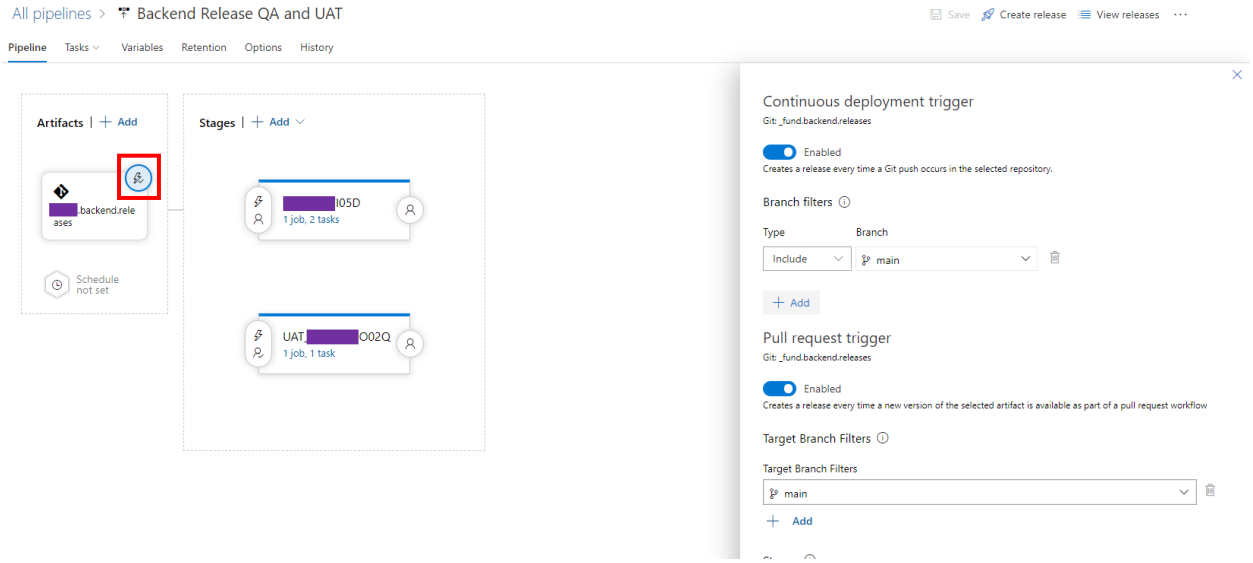


e. Once the Variable group is created, it can be seen under Pipeline's variables section.



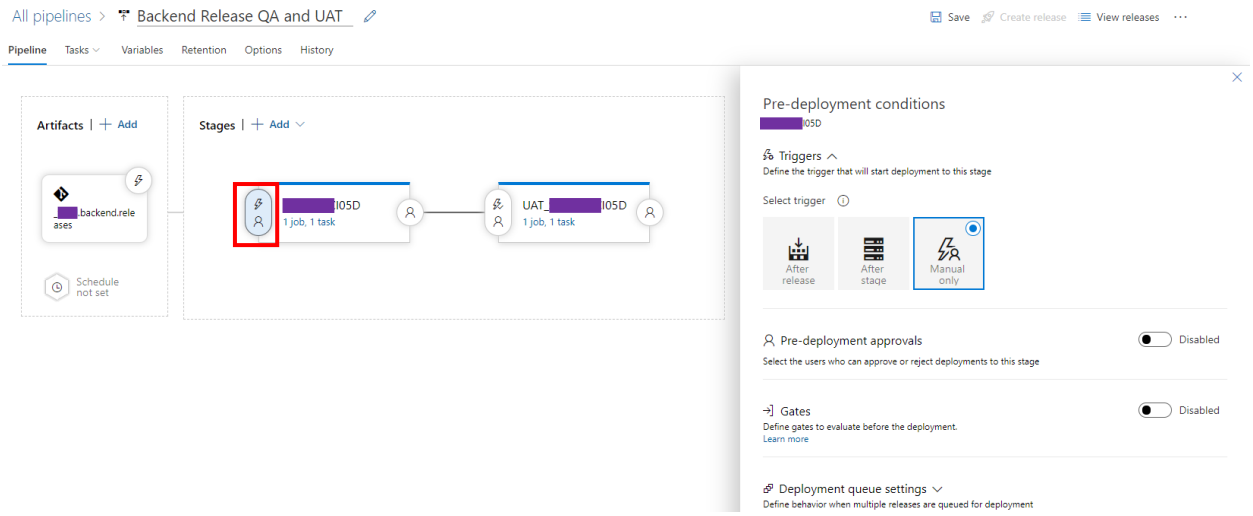
Step 5: - Auto **Azure DevOps release (Note. it is not backend release)** creation and manual deployment (It can include backend release) setting.

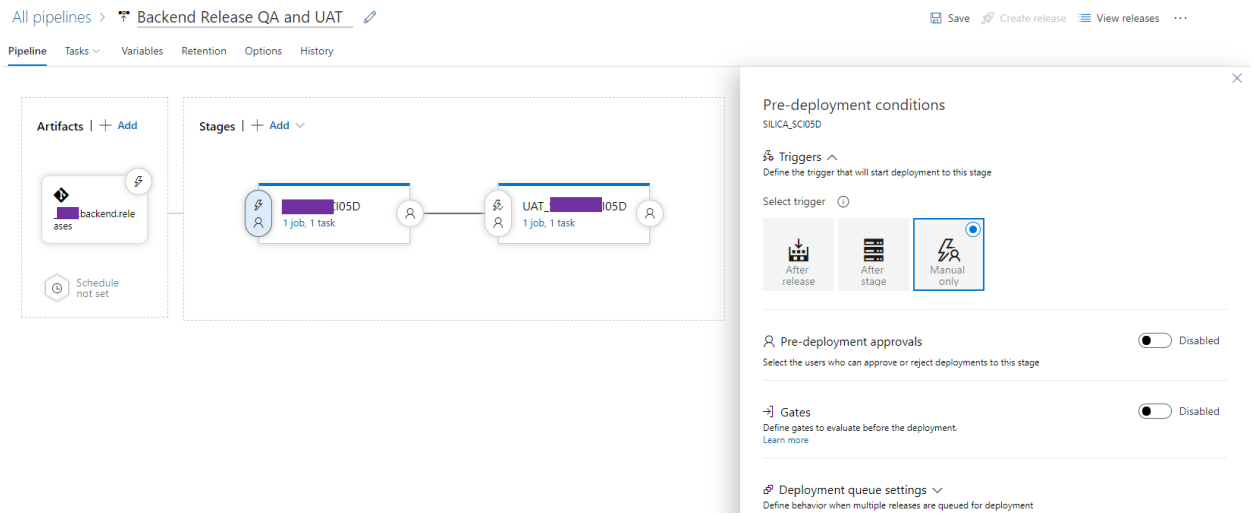
a. Below setting can be done by clicking on the Continuous deployment trigger button (shown in the red box). Enable the option for Continuous deployment trigger. Enable the option for the Pull request trigger.



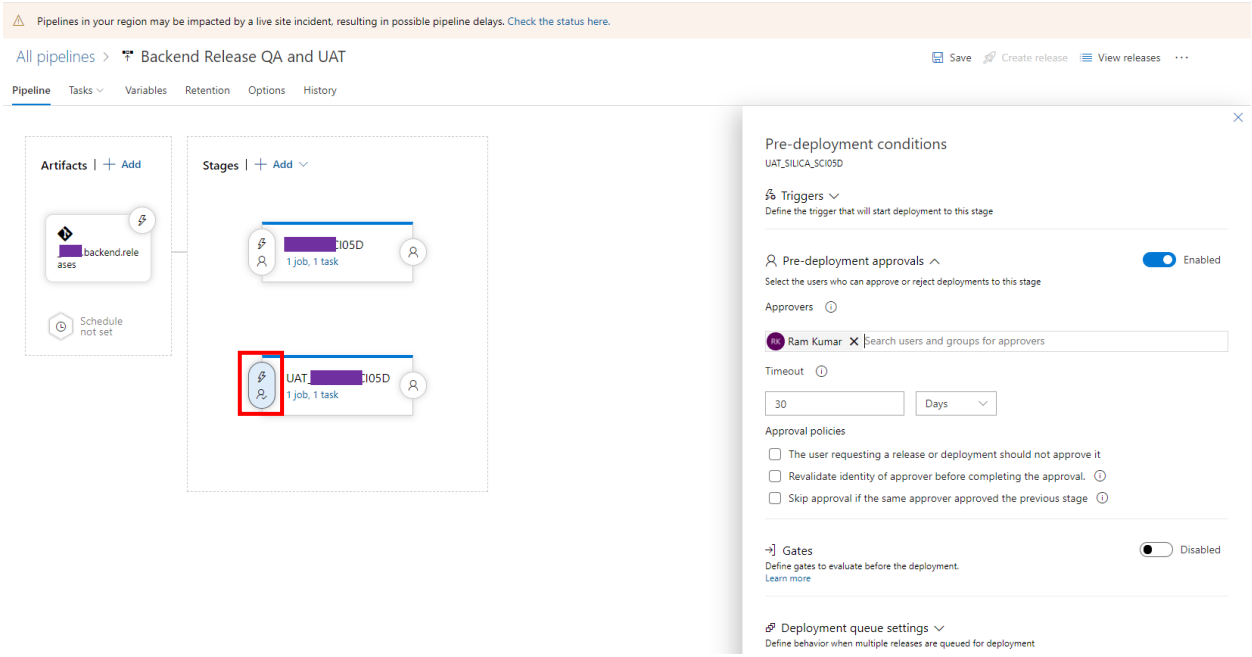
b. Below setting can be done by clicking on Pre-deployment conditions (shown in red box).

Select the **Manual only** option under the Triggers section.

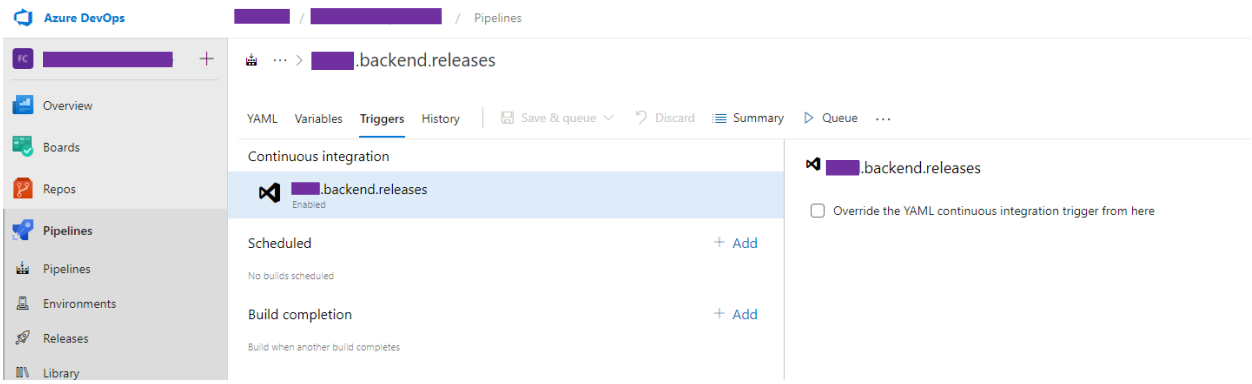


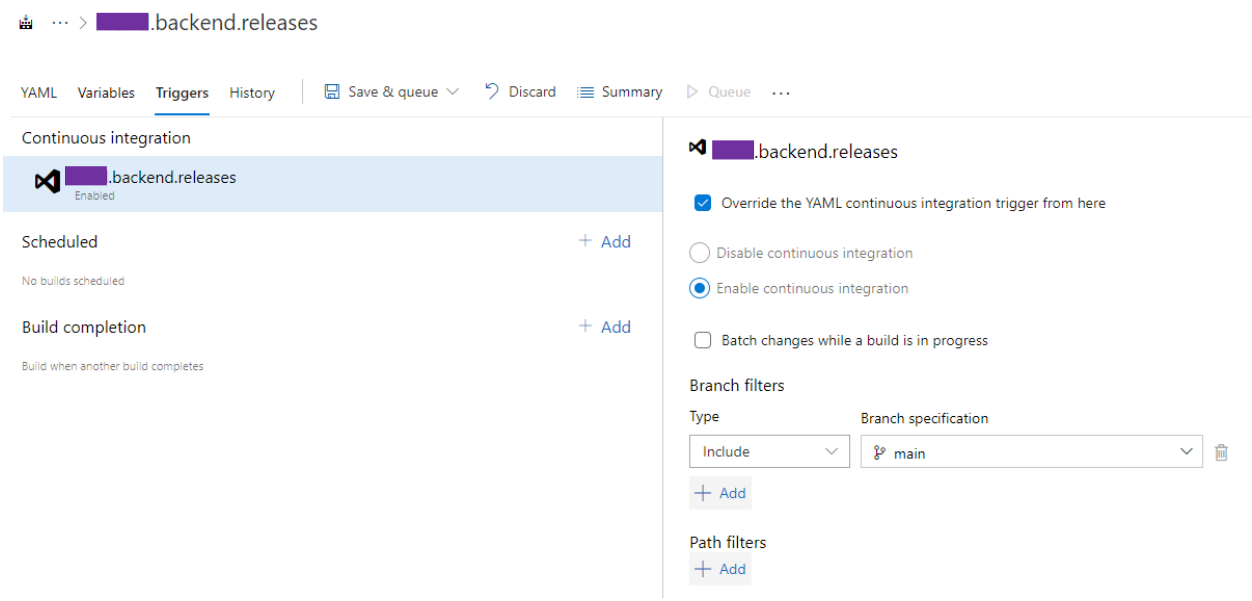


c. Below setting can be done by clicking on Pre-deployment conditions (shown in red box). Select the **Manual only** option under the Triggers section. For UAT and PROD environments select Pre-deployment approvals as **Enabled**.



Blow screenshot shows how to find trigger option of a build pipeline, in case of trigger override.


Azure DevOps
Pipelines
Overview
Boards
Repos
Pipelines
Pipelines
Environments
Releases
Library
.backend.releases
YAML
Variables
Triggers
History
Save & queue
Discard
Summary
Queue
Continuous integration
.backend.releases
Enabled
Scheduled
Add
No builds scheduled
Build completion
Add
Build when another build completes
.backend.releases
Override the YAML continuous integration trigger from here


.backend.releases
YAML
Variables
Triggers
History
Save & queue
Discard
Summary
Queue
Continuous integration
.backend.releases
Enabled
Scheduled
Add
No builds scheduled
Build completion
Add
Build when another build completes
.backend.releases
Override the YAML continuous integration trigger from here
Disable continuous integration
Enable continuous integration
Batch changes while a build is in progress
Branch filters
Type
Branch specification
Include
main
Add
Path filters
Add