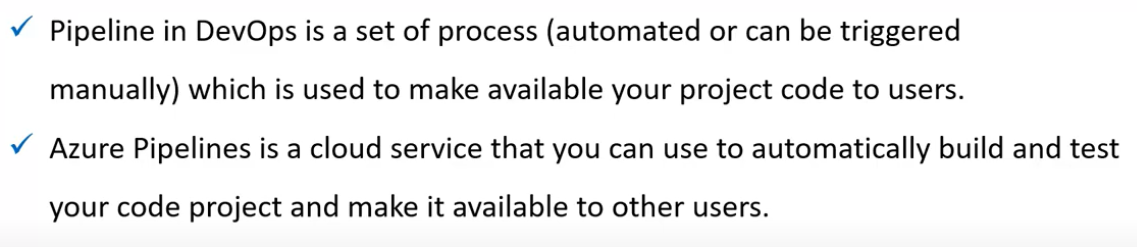
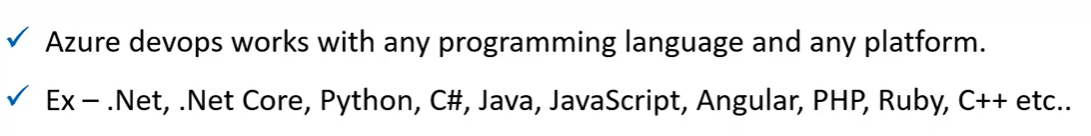
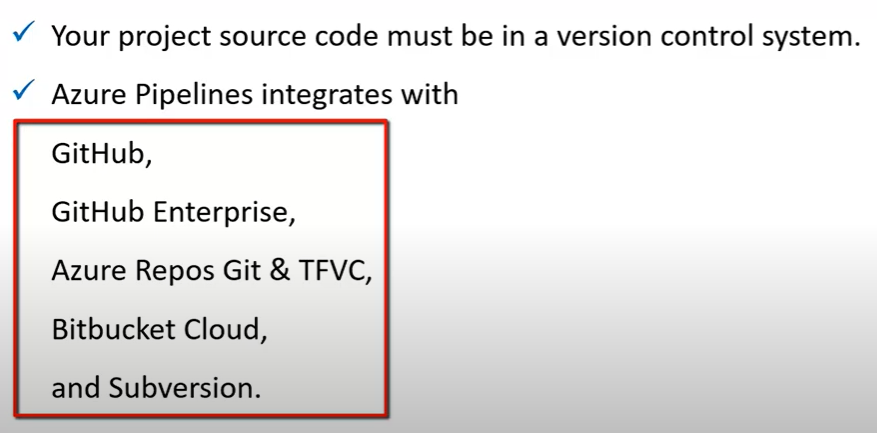
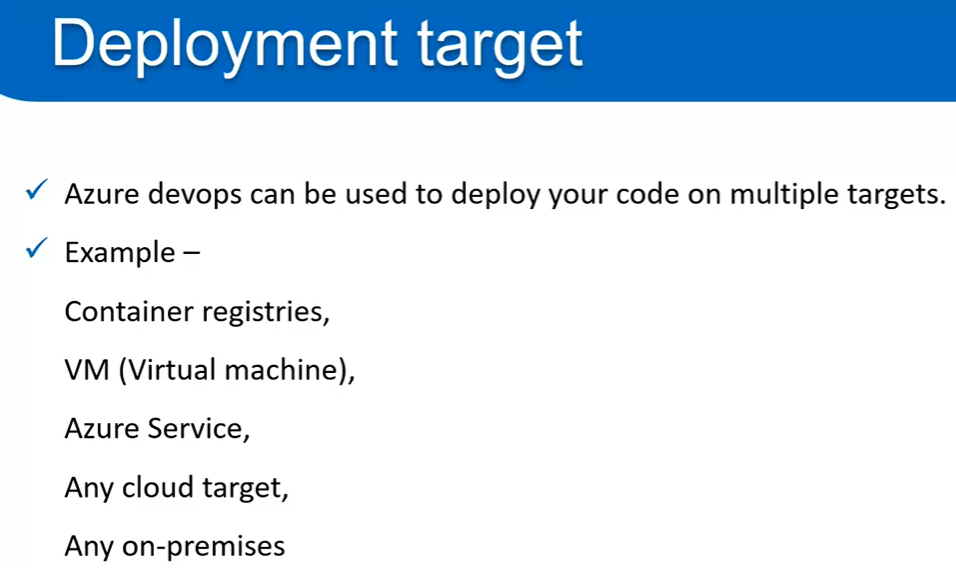
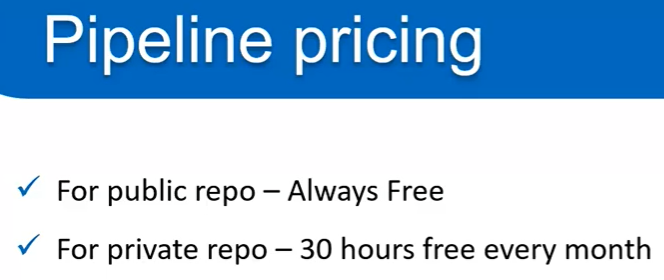
**Pipelines:**



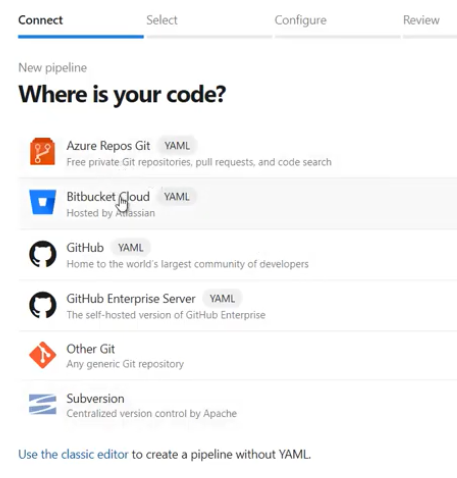




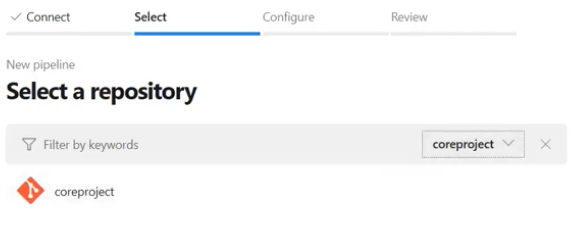




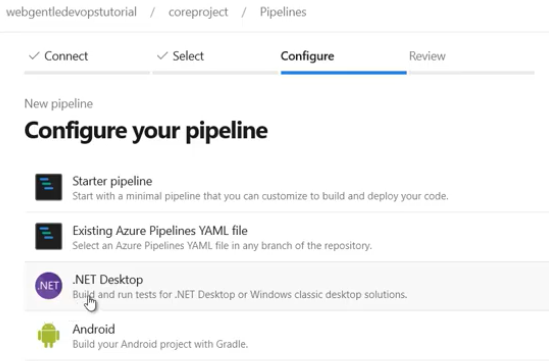
**Creating a pipeline yaml method:**



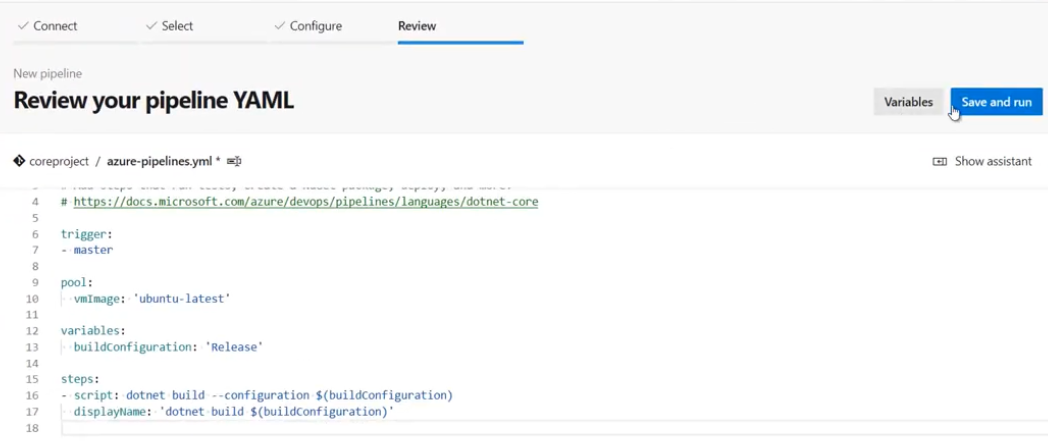
* First step, we need to select is which code repo we are using as above.



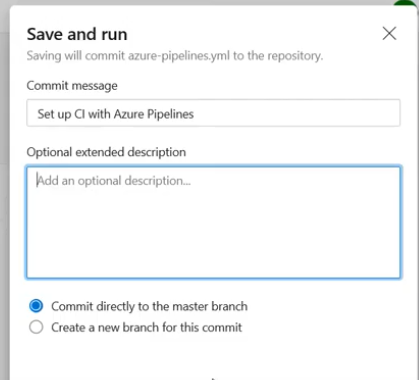
* Then select organisation and project as above.



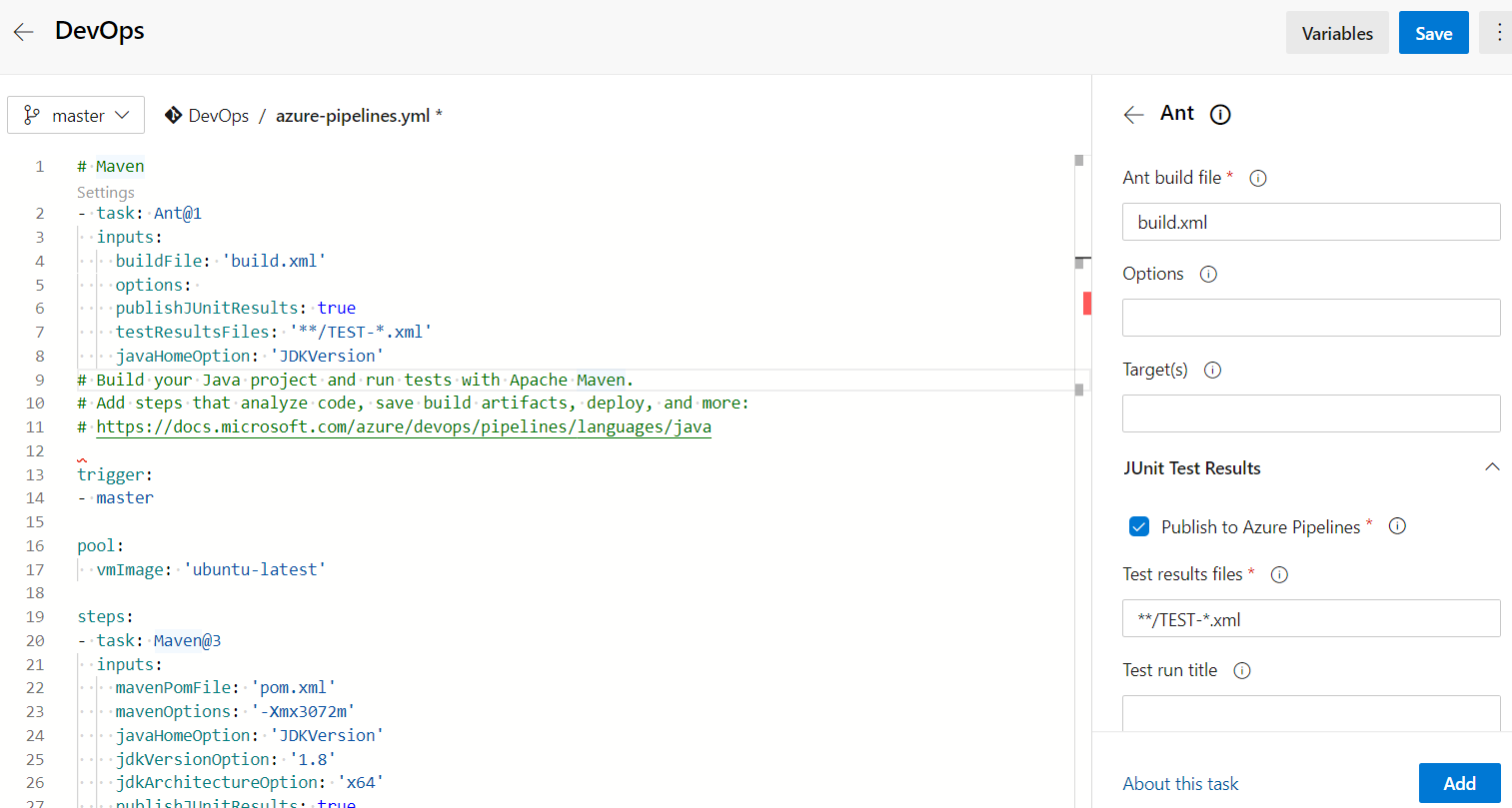
* After that, we need to select which code are we going to build. So that the template will be created for the respective code.



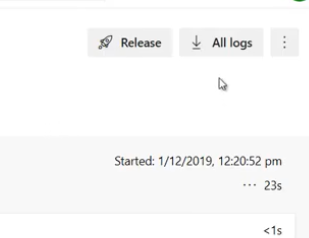
* Finally, in review section we can see the yaml file got created as above with the name of **“azure-pipelines.yaml”** in **“coreproject”.**
* We can rename it if we want.



* Now, we can save and run the pipeline as above and make a commit to master branch or create new branch for the commit.
* We can see in repo that **“azure-pipelines.yaml”** file will be created after this.
* If we want to add any more script or steps to the pipeline. We can add it to the yaml file.
* Or we can edit the pipeline and add the step-in right side of the screen as below. So that the yaml file will be autogenerated by adding the steps.

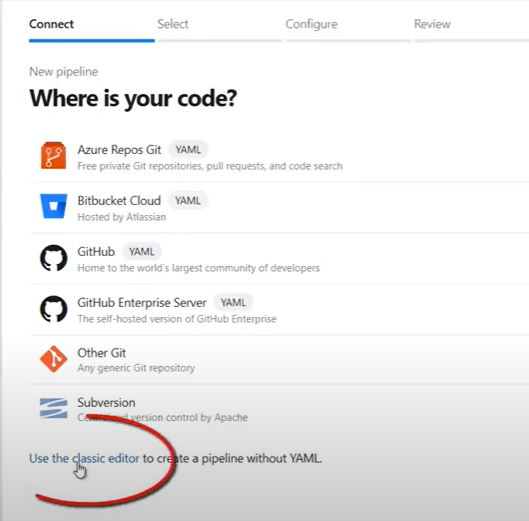


* We can also download all build pipeline logs with below option after opening the build.

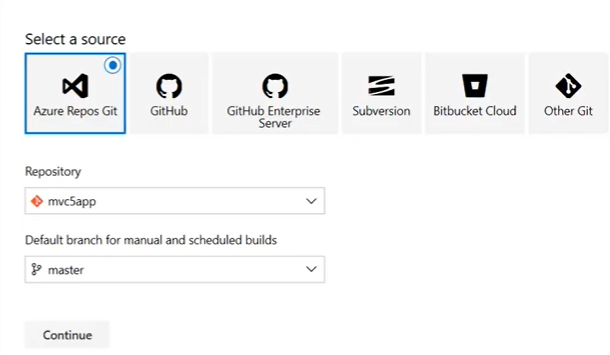


* We can set the notifications on build failure or success in organisation/project settings 🡪notifications option.

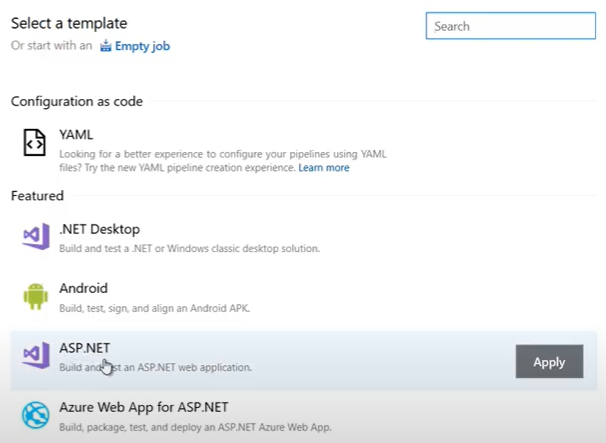
**Creating build pipeline with classic editor:**



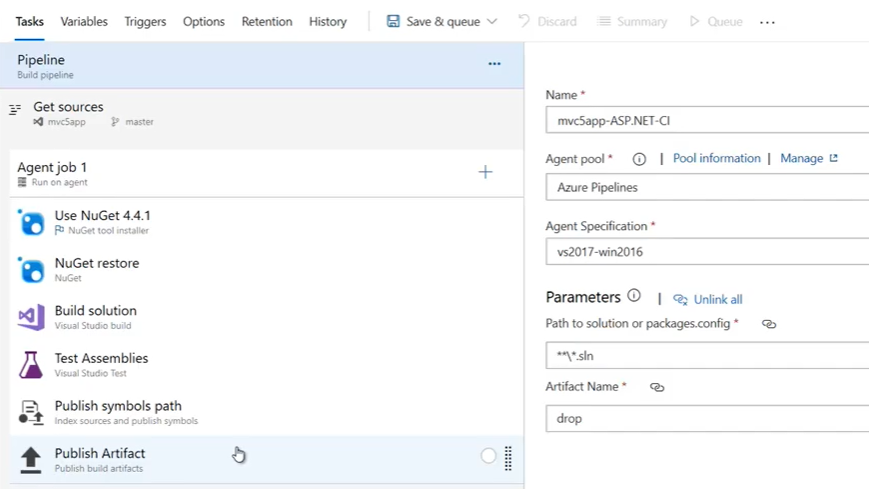
* While creating the pipeline, we need to select **“use the classic editor”** top to create build pipeline with UI not with yaml file.



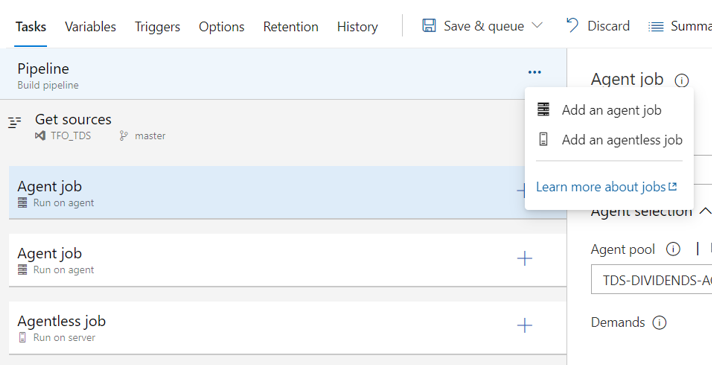
* In next step, we will have to select code repo type, repository and the default branch as above. Then click on the continue button.



* Now, we need to select the template depending on the type of the application we are going to build.

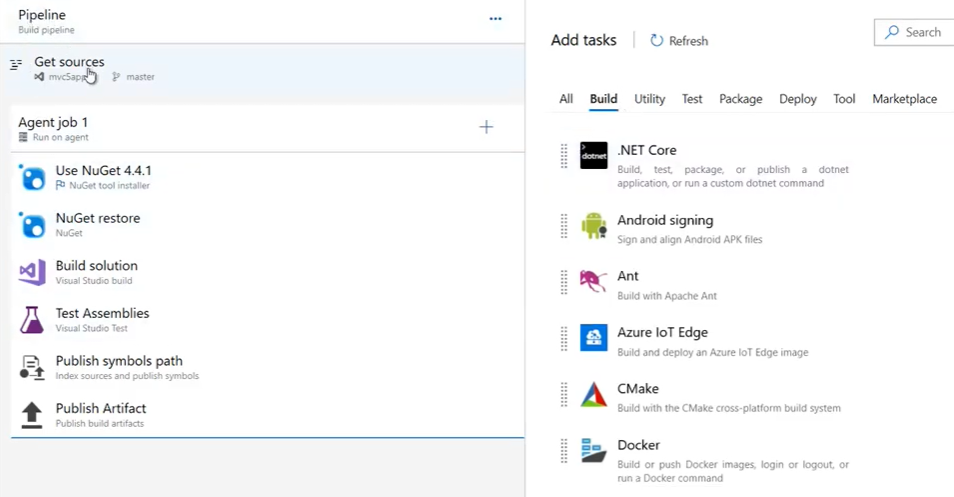


* We can also select any step and view the yaml file for the particular step.



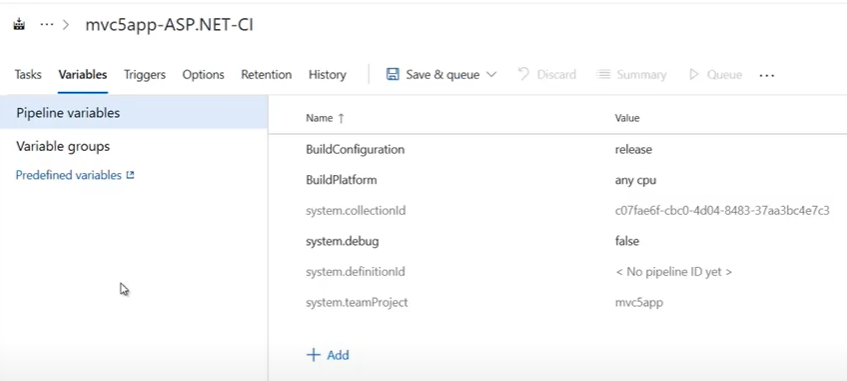
* We can add agent job or agentless jobs in pipeline as above.
* Under that job. We will have tasks or steps to build and process the code. And each job we can run in different agent pools.

**Tasks:**



* Under the job, we can add tasks as above with the templates. We can also go to marketplace and download the required extension and use it.

**Variables:**

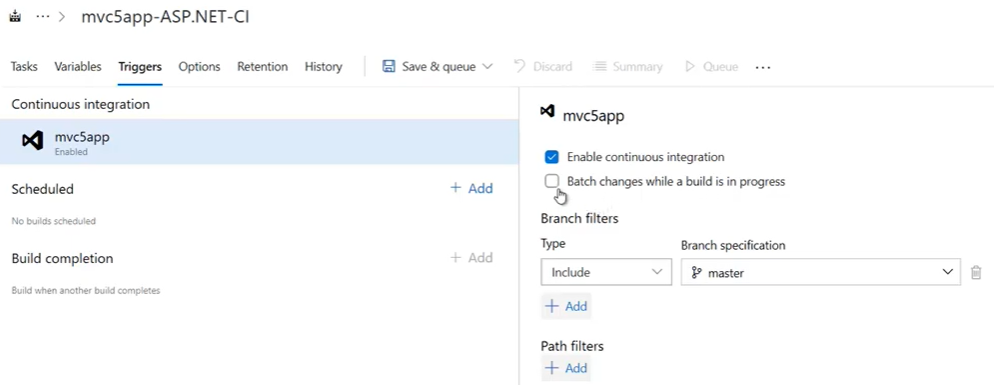


* We can add any variables as above if we want and add those variables in task as **$(variable).**

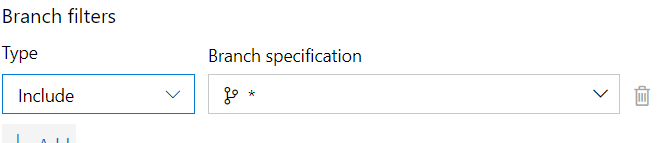
**Triggers:**

* Triggers are depending on the type of repository we are using.

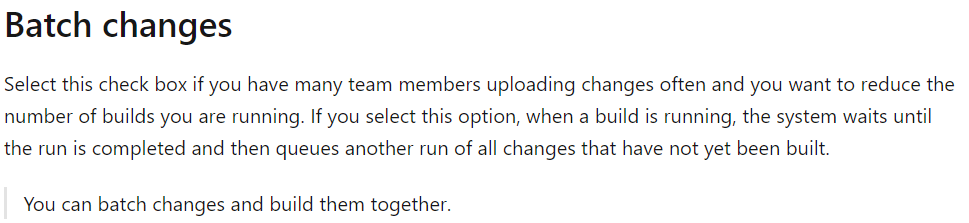
**CI triggers in azure repos git:**

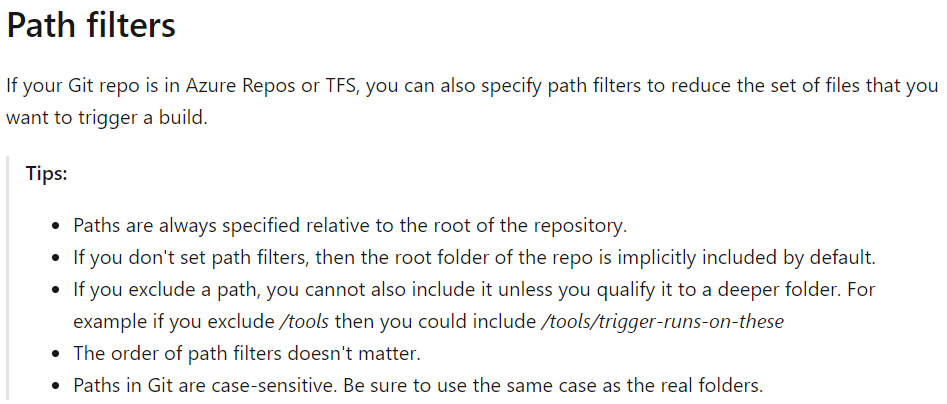


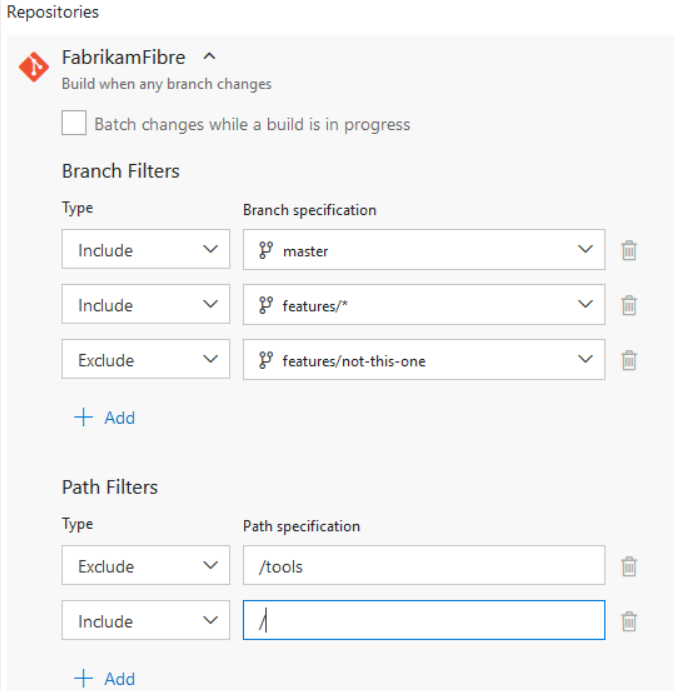
* Here in triggers, we can enable **“continuous integration”** option so that the build will be triggered on every check-in. we can include or exclude the branches along with that.



* We can put \* for the continuous integration on all the branches

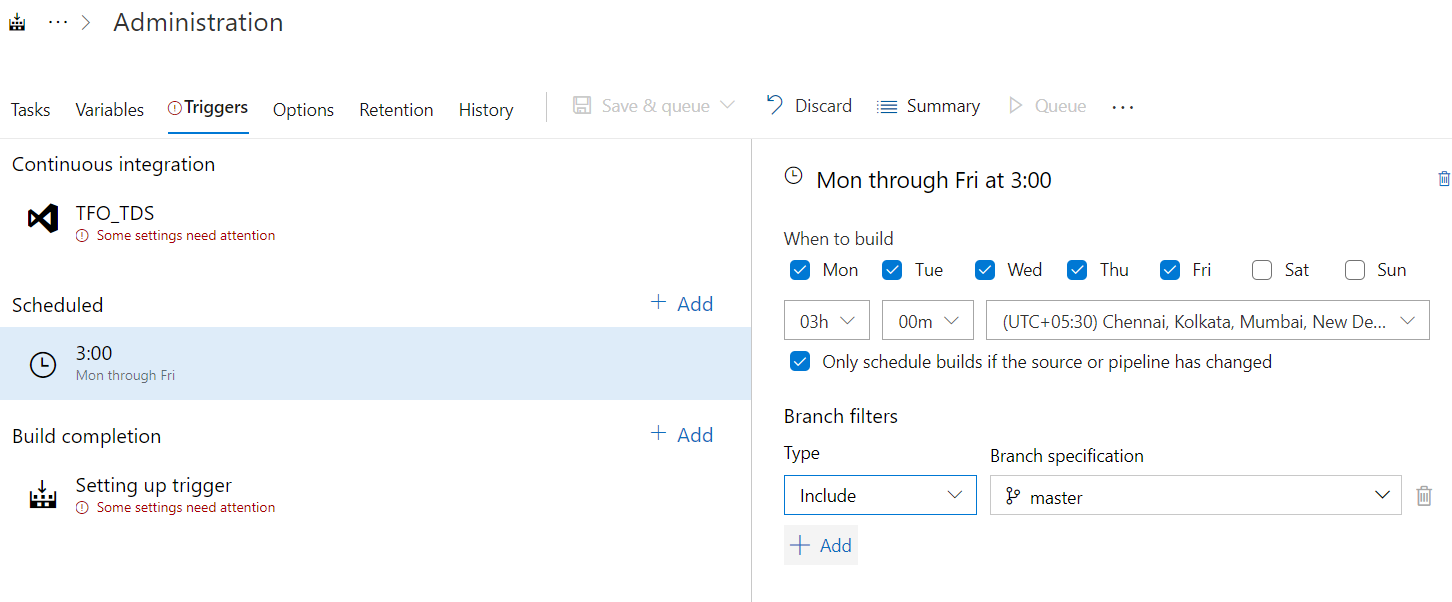






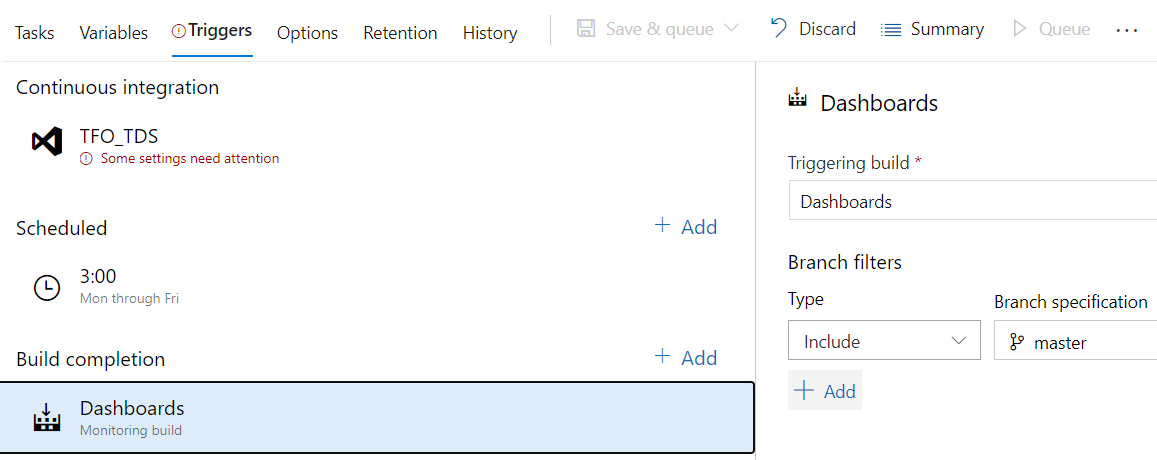
**Scheduled triggers:**

* We can also have scheduled triggers on pipeline as below.



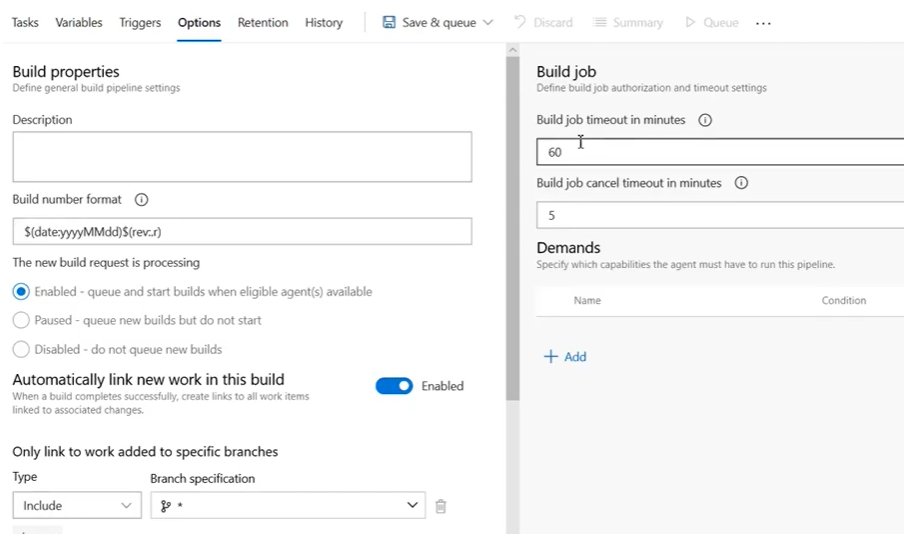
* We can schedule time, day and time zone as above.
* We can also select the option to trigger the pipeline only if there is any change in source code or the pipeline as per the scheduled time.
* We have branch filters to include/exclude here as well.

**Build completion:**

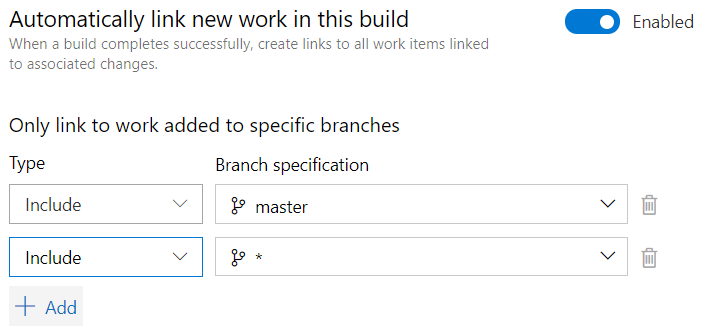


* We can add another build job to trigger once the current one is completed as above.
* We can also use branch filters here.

**Options:**

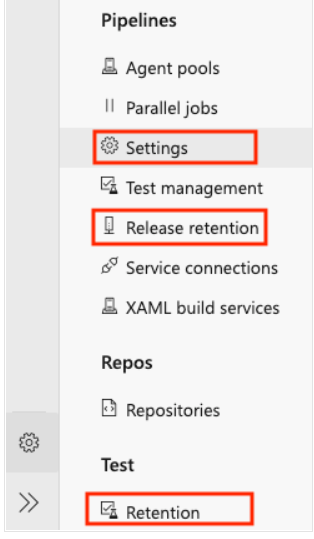


* Here, we can modify **build number format**. Enable to keep build in queue and trigger whenever the agent is free, or we can pause the build to be in queue but do not start even when the agent is free, or we can disable the job to not queue new build
* We can also set the timeout on the right side.

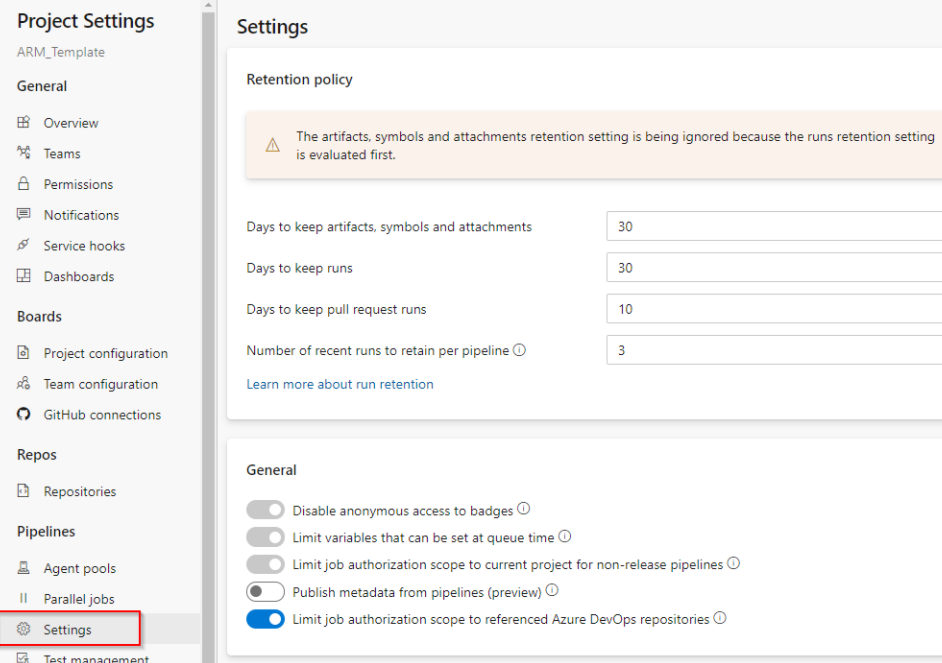


* In the downside, we can also include/exclude branches as per our requirement. Only included branches will be displayed at the time of triggering pipeline.
* We can keep \* for all the branches

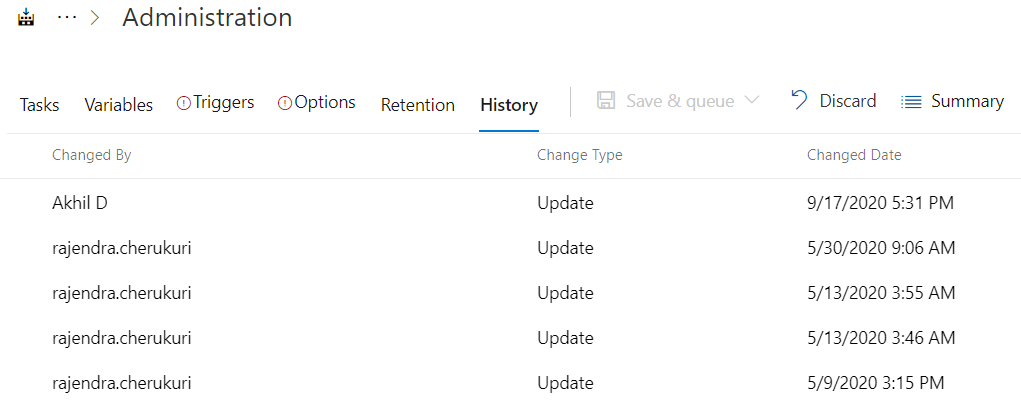
**Retention:**



* In project settings, we need to go to **settings** to enabled retentions for pipelines. **“release retention”** option for releases and **“retention”** option under test for test plans
* Below are the retentions that we can enable for the pipelines.



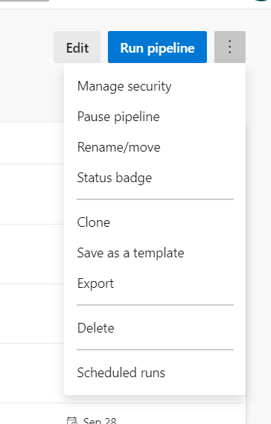
**History:**



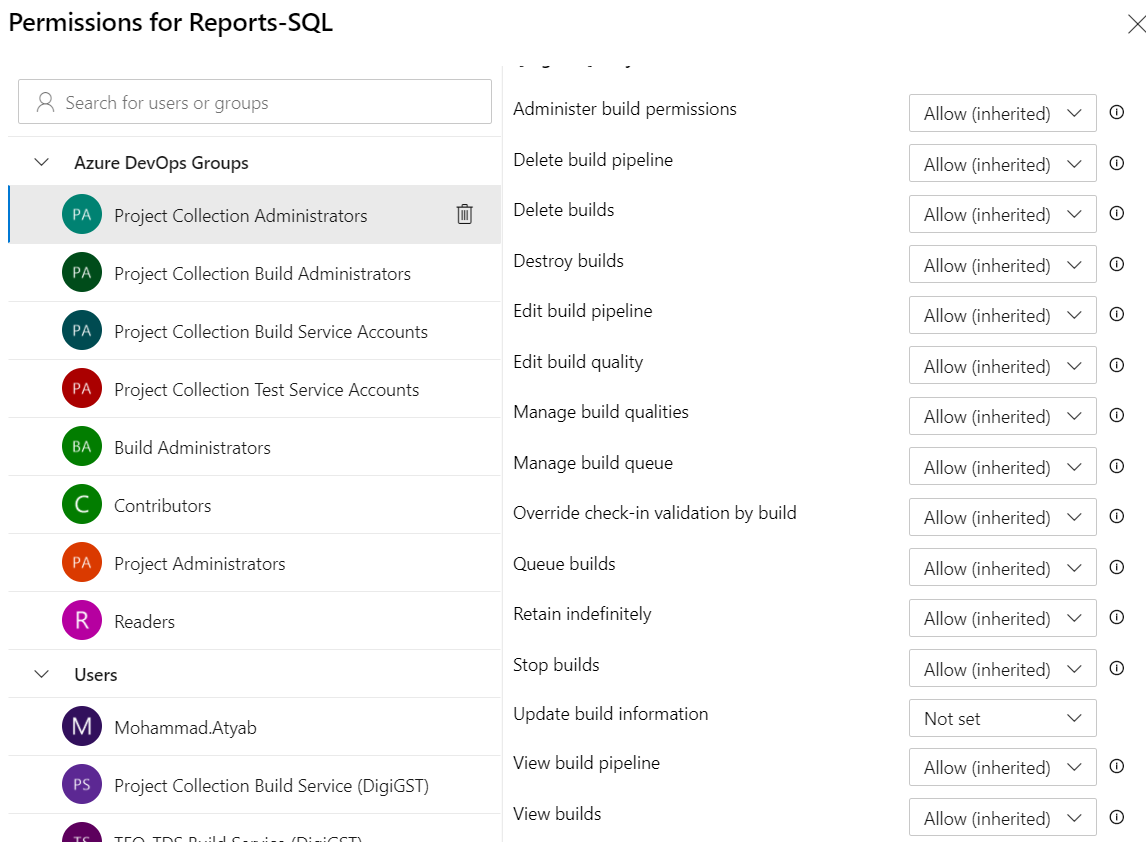
* Here, we can see the history of pipeline like who modified and when.
* We can also compare the difference between changes and revert the pipeline to any specific change.

**More options:**

* We can use below options on pipelines.



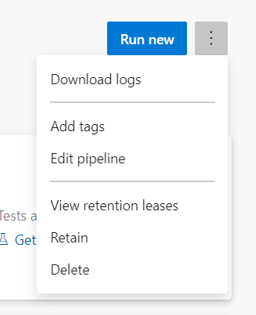
* We can assign below permission on pipeline to users or groups.



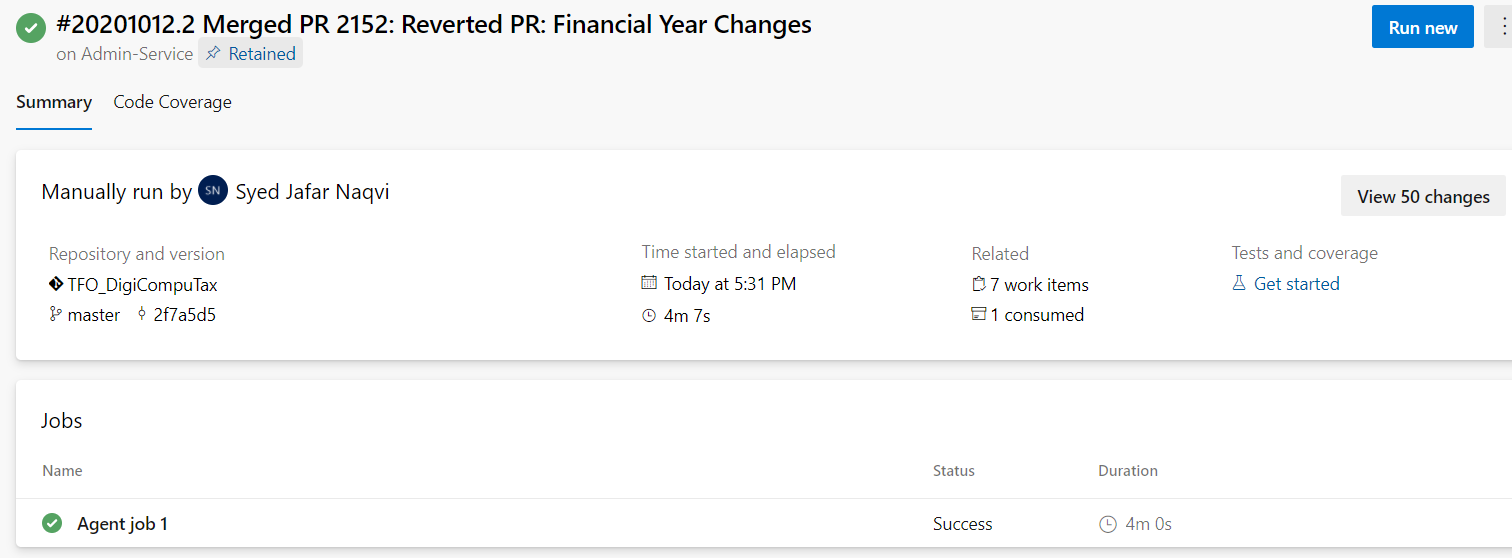
* If we pause the pipeline, we can’t trigger it until we resume it.
* Then we can rename/move to any other folder.
* Also, we can clone it and modify as a separate pipeline.
* Save as a template and use it later for any new pipeline.
* We can also export the pipeline to a json file and import to newly creating job instead of adding all these steps again freshly.

**Pipeline builds options:**

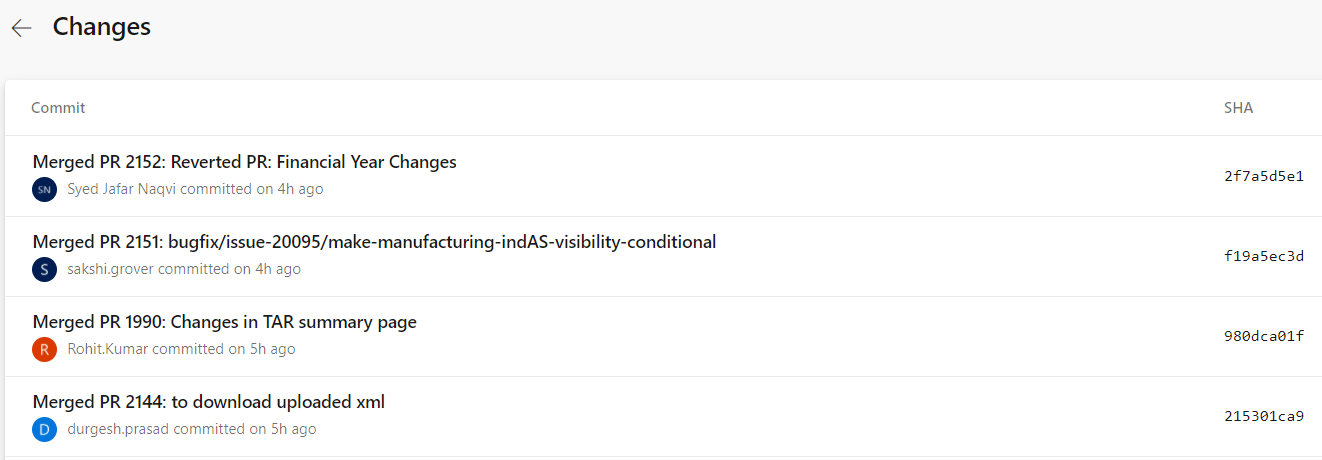
* If we open any build of a pipeline, we can see the below options.



* We can run the pipeline newly again.
* We can download the build logs.
* We can add tags etc.

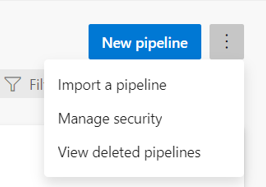


* We can also see the build details like who triggered, what time, how much time build took, code repo, branch and the changes from the old build as below.

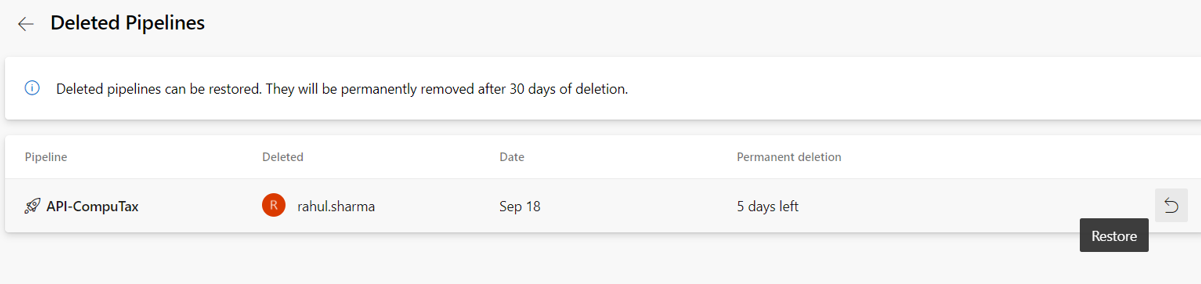


**All build pipelines options:**

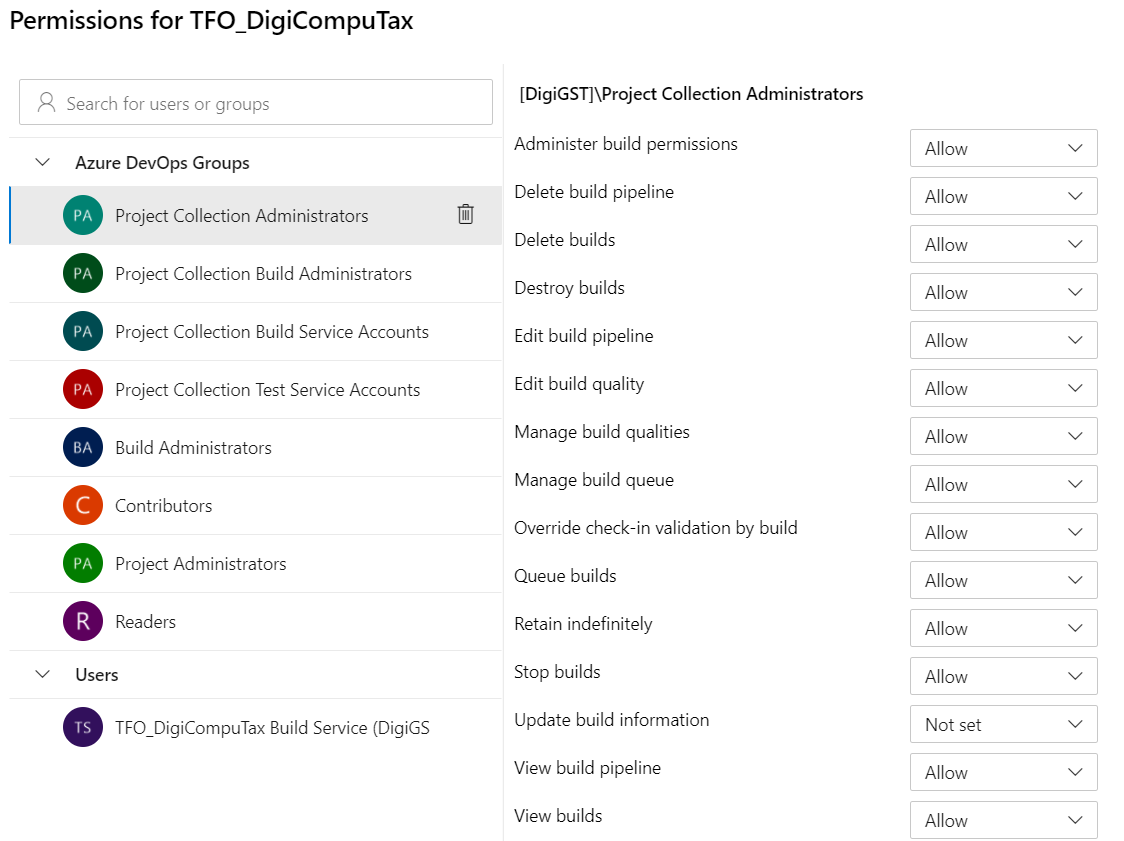
* If we just go to pipelines option but not into any particular pipeline. We can see the below options.



* Here, we can import a pipeline if we have exported pipeline’s json file.
* We can view the deleted pipelines and restore as below.

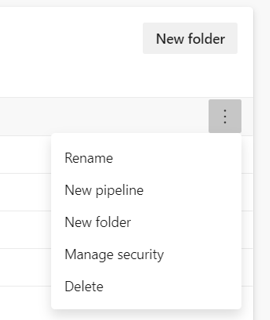


* The pipelines will be deleted automatically after 30 days
* Also, we can add the below permissions to all the pipelines at a time under “manage security” option.

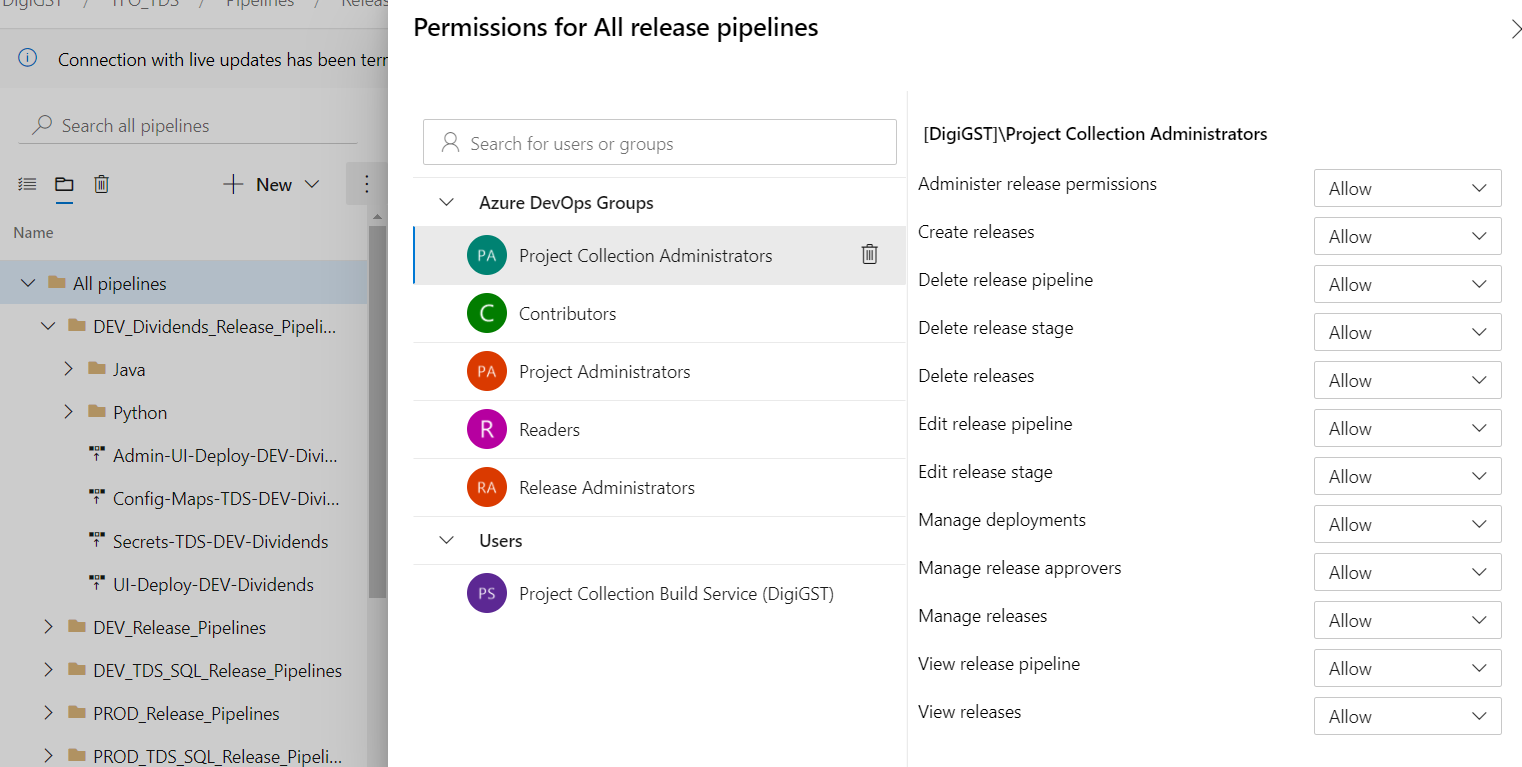


**Folder options in pipelines:**

* We have below options in for folders in pipelines.



* Here, we can add new pipeline, rename folder, add sub folder or delete the folder
* We can also give the below permission in folder level under **“manage security”** option.



* We can simply add the security to all the pipelines instead of going to each and every one.