Azure Artifacts

9. Azure Artifacts

Azure Artifacts enable developers to consume and publish different types of packages to Artifacts feeds and public registries such as NuGet.org and npmjs.com. You can use Azure Artifacts in conjunction with Azure Pipelines to deploy packages, publish build artifacts, or integrate files between your pipeline stages to build, test, or deploy your application.

Publish and Download Pipeline Artifacts

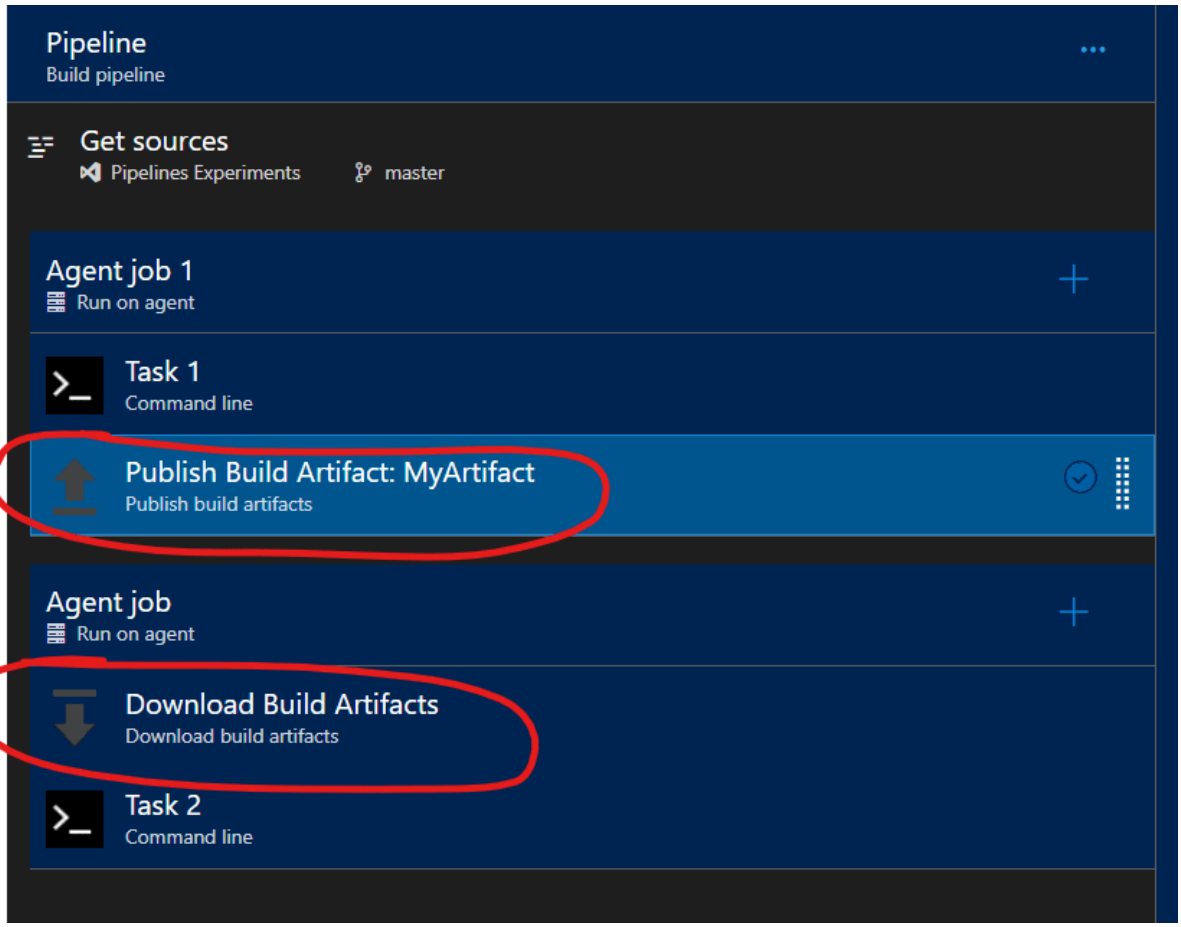
Azure DevOps Services

Using Azure Pipelines, you can download artifacts from earlier stages in your pipeline or from another pipeline. You can also publish your artifact to a file share or make it available as a pipeline artifact.

Build Artifacts

Build Artifacts have been in Azure DevOps for a long time and are the **built-in artifact storage mechanism** for Azure Pipelines.

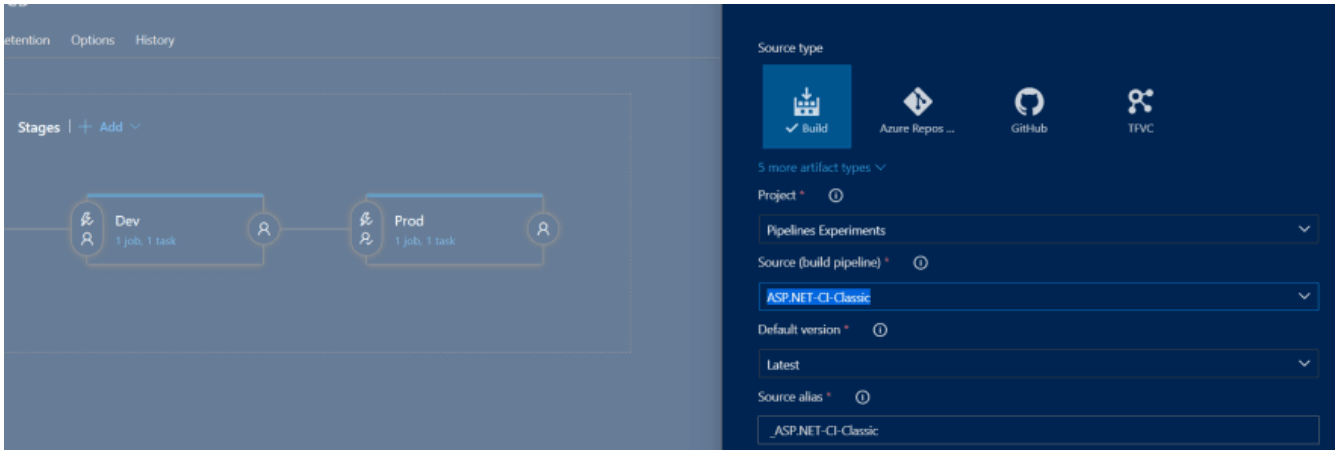
They can be used in both the Classic Build Pipelines, the one created using the UI, as well as the newer YAML Pipelines.



TechData-Infinity-Devops with MultiCloud

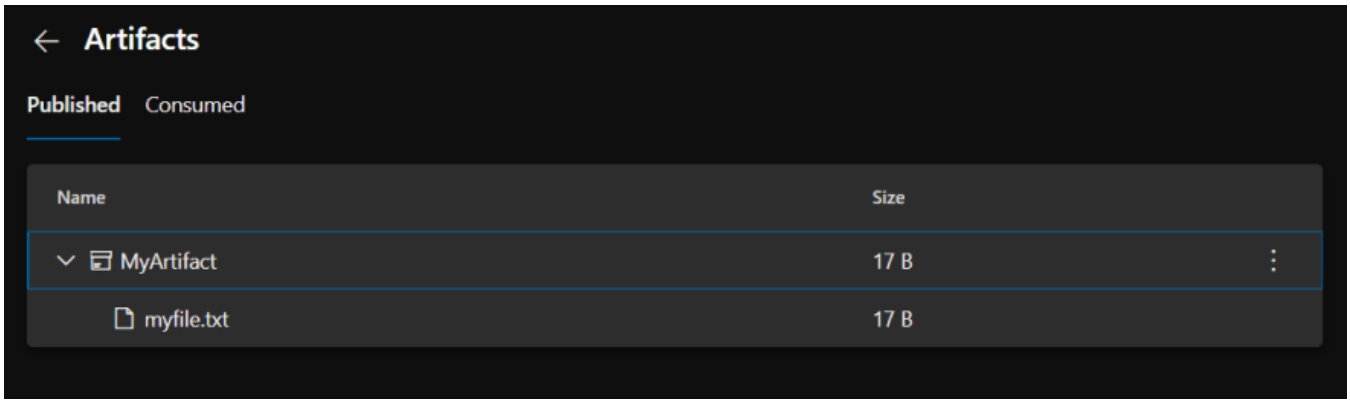
Build Artifacts are published via the Publish Build Artifacts task and can be downloaded with the Download Build Artifact task. And when you publish them, you can instruct the task to either push the content up to the Azure DevOps cloud or serve, or to copy the files to a local file share instead.

Build Artifacts can be consumed from other jobs in the same Pipeline, and from other Pipelines.



Additionally, Build Pipelines can be used if you want to consume your artifact from a Release Pipeline triggered by the build completion.

And you can always download your artifacts from the Build run status page.



And as you can see in the image above, you can explore the content of your artifact directly in the UI.

Pipeline Artifacts

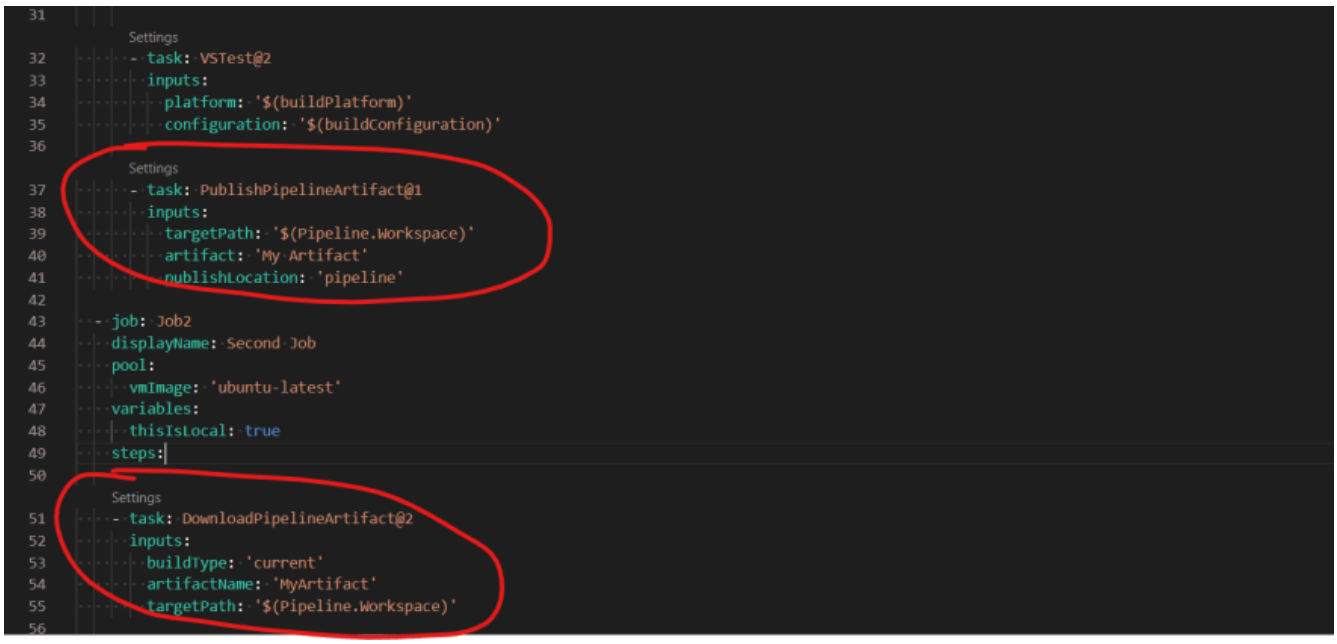
Alright, let's talk about **Pipelines Artifacts** now.

These are the **newer version**, if you will, of Build Artifacts, and as such they can be used only from within the YAML Pipelines.

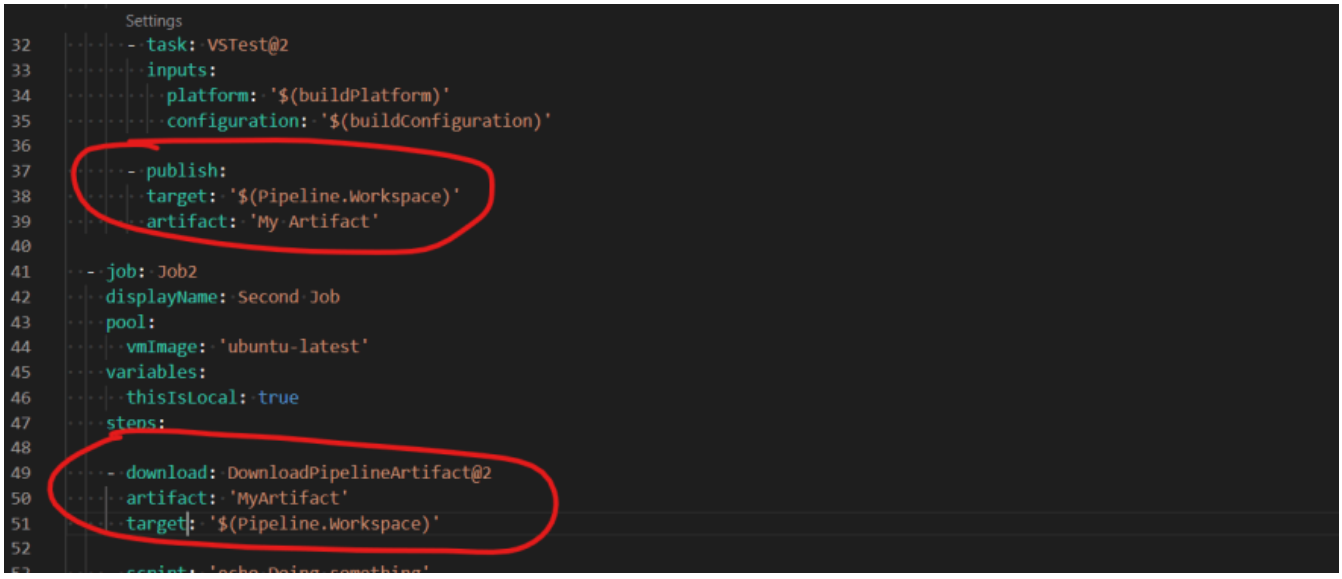
One of the main benefits of Pipeline Artifacts is that they can **dramatically reduce the time** it takes to upload and download the artifacts because of the way the files are both uploaded and stored. And this is especially true for large artifacts.

Until fairly recently, Pipelines Artifacts couldn't be used in Classic Release Pipelines, or from other Pipelines, but now that limitation is gone so their usage is very similar to Build Artifacts.

TechData-Infinity-Devops with MultiCloud



To publish the Pipelines Artifacts you can use the Publish Pipeline Artifact task and you can download them using the Download Pipeline Artifact task.

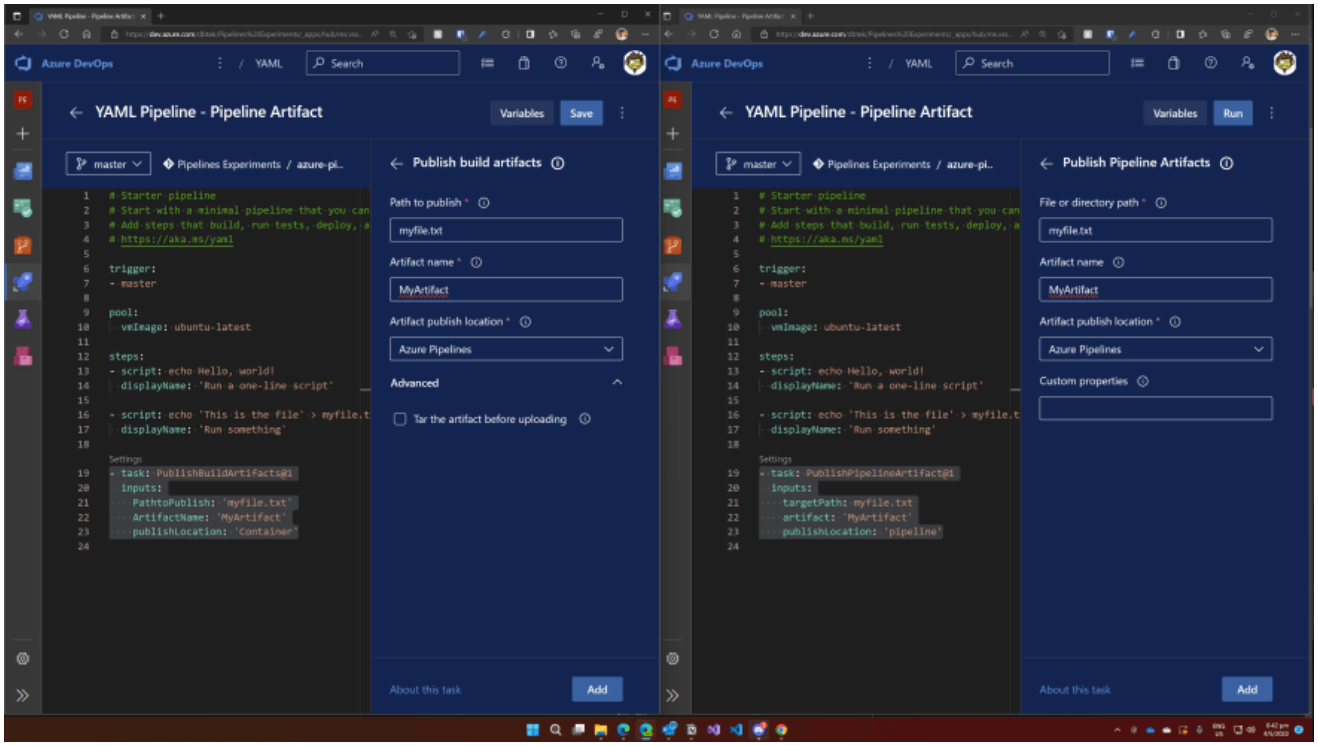


Alternatively, since this feature is only available in the YAML Pipelines, you can use the publish keyword and the download keyword, which are just the abbreviation for the whole tasks.

And if you publish a Pipeline Artifacts, and you want to use it in a deployment job in the same pipeline, you don't even have to add the download task because Azure Pipelines will download them automatically.

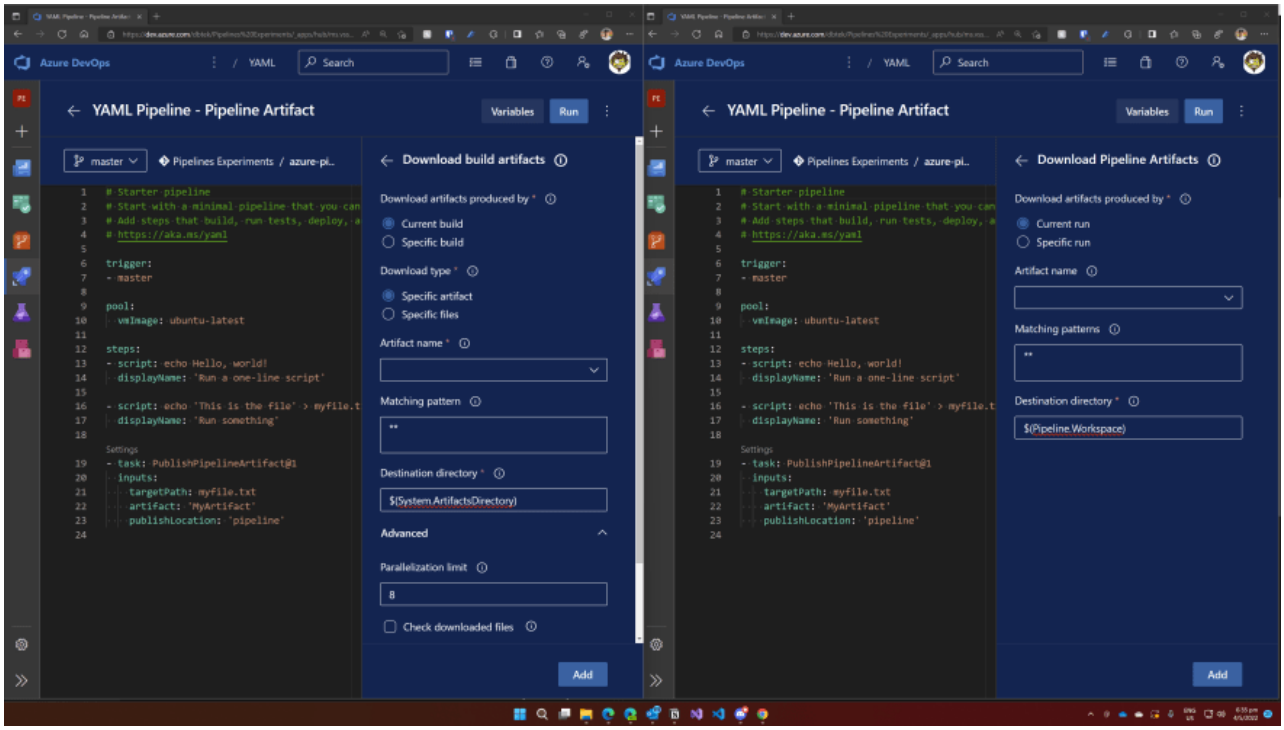
Build vs Pipeline Artifacts

There are few more differences in publishing and downloading the artifacts between Build and Pipeline Artifacts.



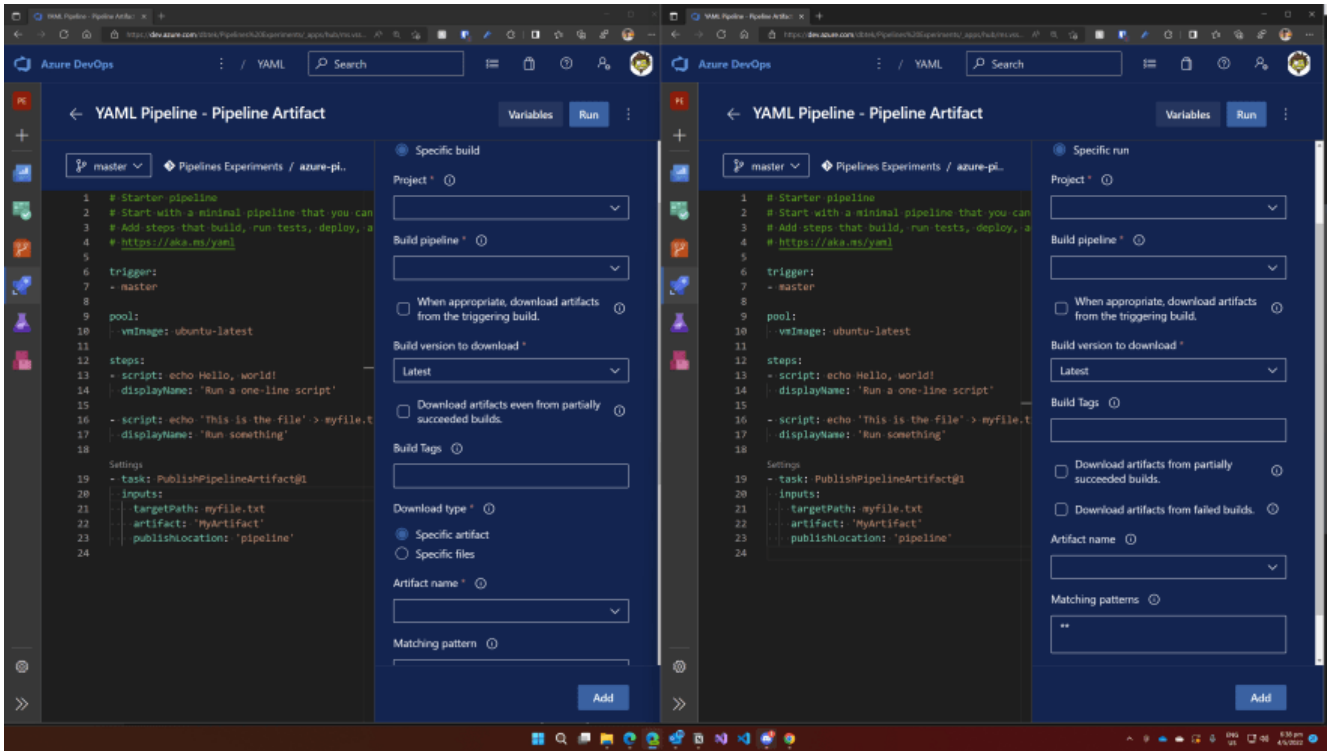
The publish tasks are virtually identical, with the only differences being that in the Publish Build Artifact task here on the left you can optionally choose to further include your artifact in a Tar file, while this is not present on the right on the Publish Pipeline Artifact Task. This one, instead, allows you to add some custom properties to the artifact. They must be in JSON format, all keys having the prefix user-

It gets more interesting if we look at the Download tasks



As you can see in this side by side view, when you download a a Build Artifact (on the left) you can choose if you want to download the whole thing, or just some specific file from the artifact. You can also set some parallelization settings and other parameters. When downloading a Pipeline Artifact, instead, you don't have that option, as you can see here on the right hand side of the image.

And if we switch the task to download from a different pipeline or run, instead of from the current one, we have one more difference.



Aside from a different positioning of the fields, you can see that when you download a Pipeline Artifact you can choose to do so even if the pipeline run you are targeting has failed

And that basically covers everything there is to say about Build and Pipeline Artifacts. If you have noticed, I kept comparing the two… but I haven't mentioned Azure Artifacts yet. Why? Well, because it is a **completely different thing**.

What About Azure Artifacts?

So Azure Artifacts… as I was saying it is pretty different from Build and Pipeline Artifacts. Despite the similar name, it's a **different service which serves a different purpose**.

Both Build and Pipeline Artifacts are very generic, you can save whatever you want in them, and what Azure DevOps does is just packaging the files in a zip archive and saving it somewhere. Azure Artifacts, instead, is a **typed package repository**.

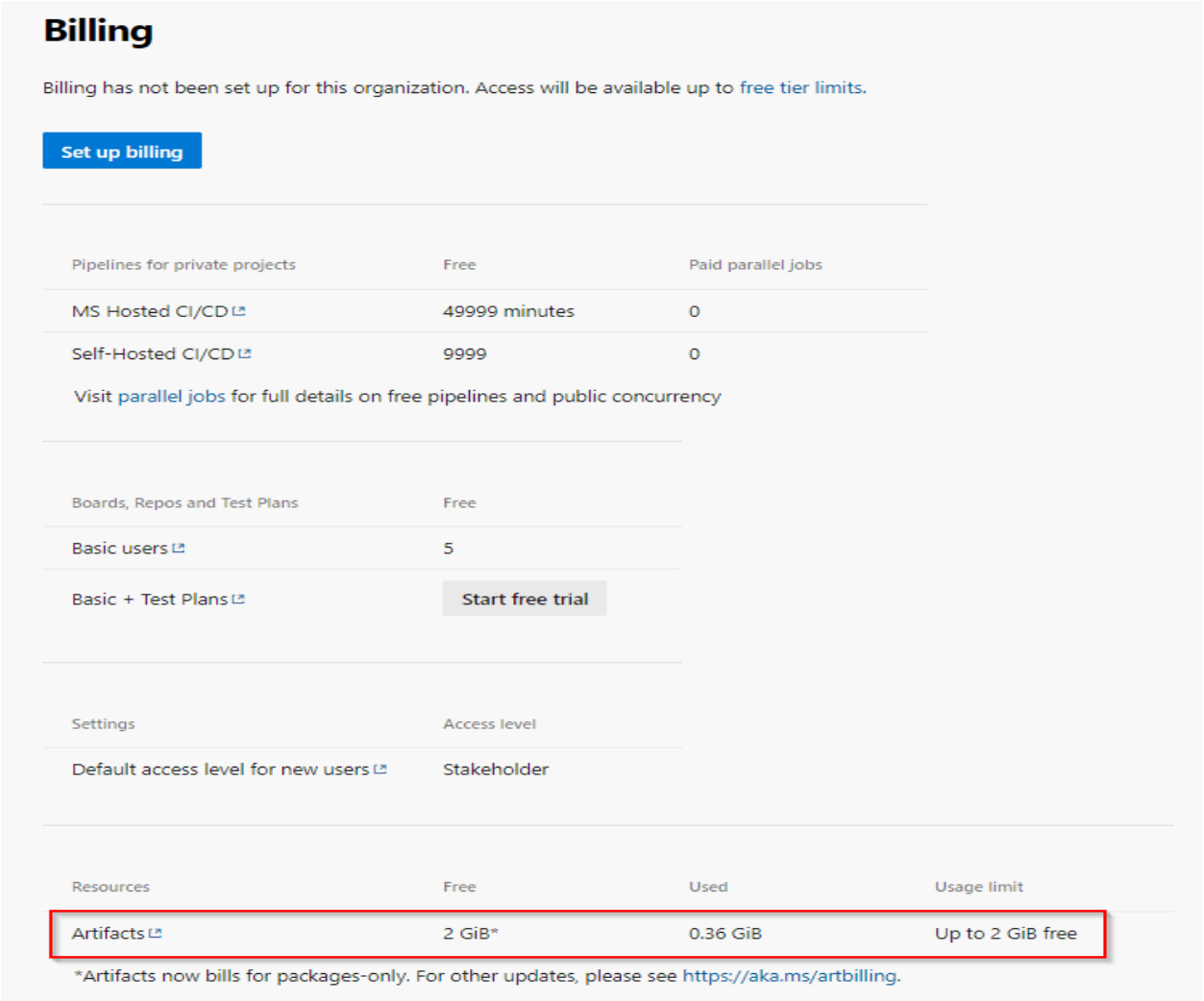
Package type	Azure DevOps Services	Azure DevOps Server	TFS
NuGet packages	Yes	Yes	TFS 2017 and TFS 2018
npm packages	Yes	Yes	TFS 2017 and TFS 2018
Maven packages	Yes	Yes	TFS 2018
Python packages	Yes	Server 2019 Update 1 and newer, Server 2020	TFS 2018
Universal Packages	Yes	No	No

It supports multiple package types such as NuGet, npm, Python, Maven, and Universal Packages… you can basically see it as an alternative to Artifactory, Nexus, GitHub Packages, and services like that.

You may have seen that Azure Artifacts also supports Universal Packages, and although that is somewhat similar to the other types of artifact we have seen before, it is conceptually different.

You would use Universal Packages when you want to create an artifact with a lifetime independent of the pipeline that created it. In fact both Build and Pipeline Artifacts are always tied to the Pipelines that created them. As we have seen, you can download Pipeline Artifacts after a pipeline has completed via the artifacts UI – but if you want something that really exists independent of pipeline you would go for Universal Packages.

Another big difference is about the pricing. Whether you use Build Artifacts or Pipeline Artifacts, you will not have to pay a single cent for them, no matter how many files you store or how big they are. Azure Artifacts, on the other hand, is billed by size.



You have a free grant of 2 Gb for each organization, but once you reach the maximum storage limit, you can no longer upload new artifacts and will need to either delete some of your existing artifacts, or set up billing to increase your storage limit.