# Continuous Integration and Continuous Delivery with Azure Data Factory and Azure DevOps

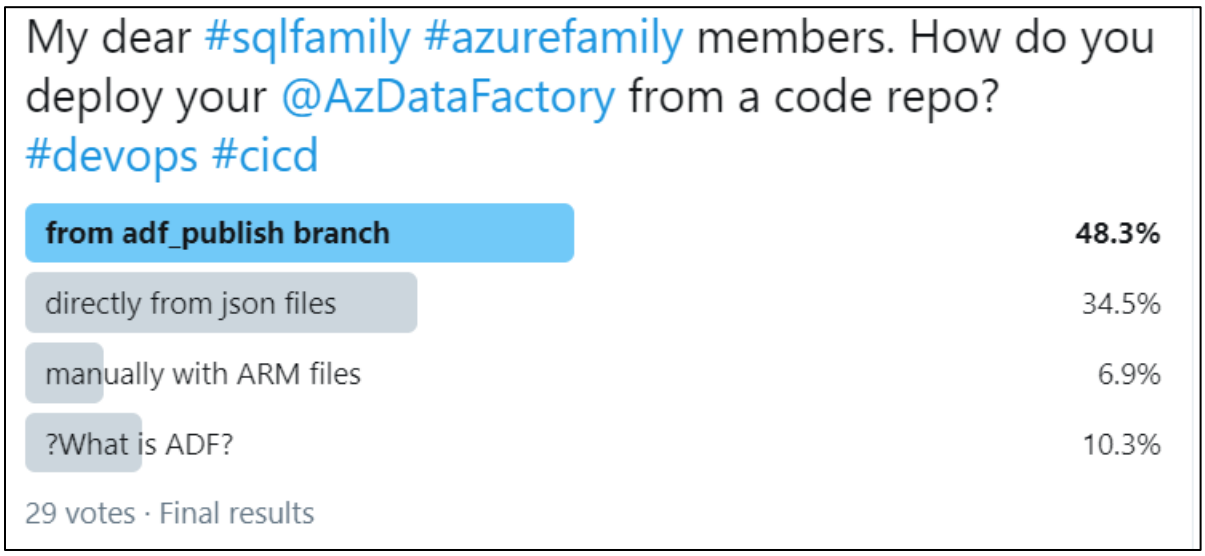
**Azure Labs by Roque Daudt (rdaudt@yahoo.com)**

## 07 – Deployment from JSON files

On March 25th, 2020, Kamil Nowinski ran a quick pool about how people are deploying ADF from a code repo. Turnout was not that great but the results were interesting. About 50 of the respondents are publishing from **adf_publish**, as recommended by Microsoft.

In a close second, though, came deployment straight from the json file.

https://twitter.com/NowinskiK/status/1242745169394442240




In the previous lab deployment was based on ARM templates in adf_publish, created out of the json files in master. This time we will bypass adf_publish altogether, using the json files in master as the source for deployment.

Additionally, we will use a tool created by Kamil to make things even easier. The tool is still in Preview, though. Therefore, we don't recommend its use in production at this time.

### Provision the UAT environment

We will first create a new environment, UAT, to be the target of the deployment in this lab. The process is almost the same as the one that we used to provision dev, test and production.

Start by editing file **C:\ADFDEVOPS\resources\createUATEnvironment.ps1**. At line 7, enter the same prefix that you used for the provisioning of the three previous environments.

```powershell
# script: createUATEnvironment.ps1
# functionality:
# - setup UAT azure environments in support of demo about
#   Azure Data Factory integration with Azure DevOps

# enter a different prefix. Ideally, use your initials as part of the prefix
$prefix = "rd2020"

# leave it as it is or enter the Azure location that best works for you
$loc = "West US 2"

# create the DEVELOPMENT environment
$env = "uat"
$rg = $prefix + "rg" + $env
# create resource group for uat environment services
New-AzResourceGroup -Name $rg -Location $loc
# create services for uat environment
New-AzResourceGroupDeployment -ResourceGroupName $rg -TemplateFile ".\azuredeploy.json" -prefix $prefix -env $env

```

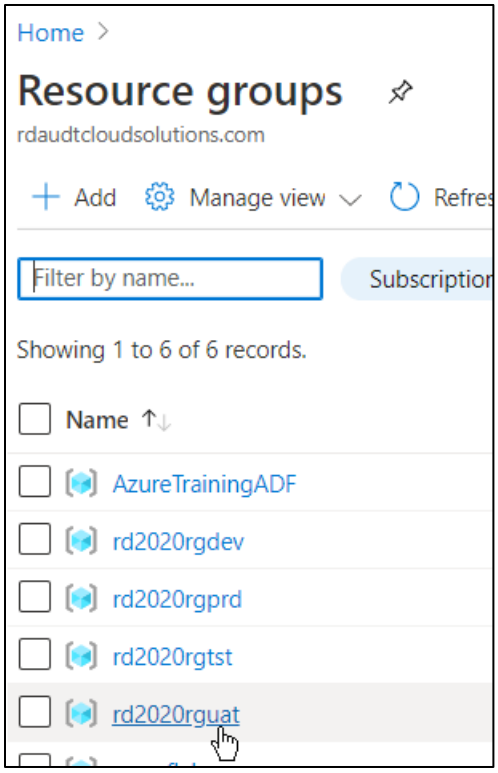
Save it, start PowerShell and connect to your Azure subscription with **Connect-AzAccount**.

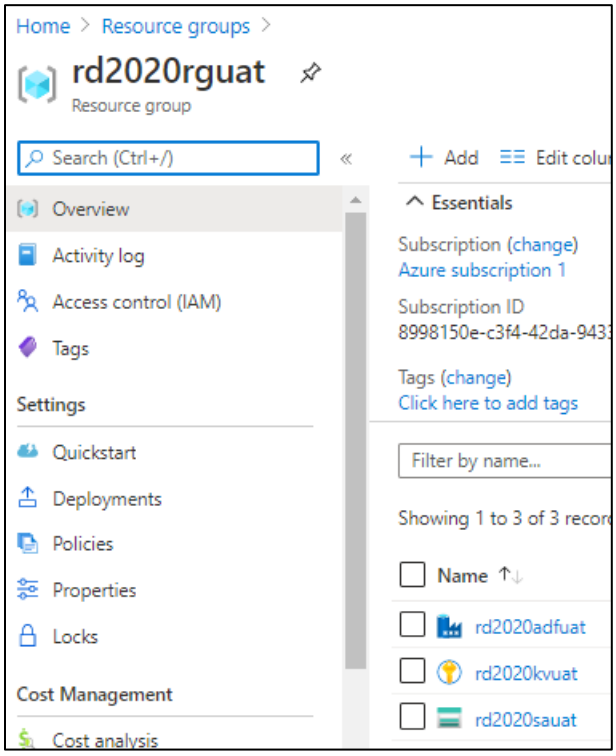
Configure the session so that you can run unsigned scripts. Run **Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass**

Then run **.\createUATEnvironment.ps1**

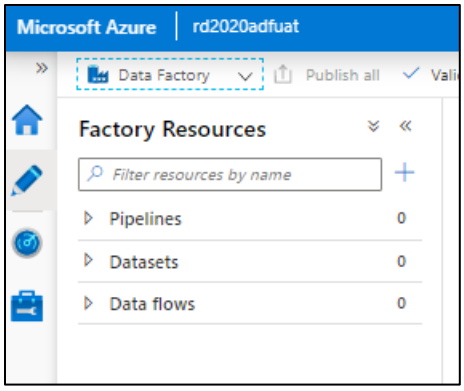
```
PS C:\adfdevops\resources> Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): Y
PS C:\adfdevops\resources> .\createUATEnvironment.ps1


ResourceGroupName : rd2020rguat
Location          : westus2
ProvisioningState : Succeeded
Tags              :
ResourceId        : /subscriptions/8998150e-c3f4-42da-9433-76eb32c20dea/resourceGroups/rd2020rguat
```

Once it finishes, browse Azure and verify that the new resource group and resources were created.




Open **ADF Uat** and verify that it is empty.

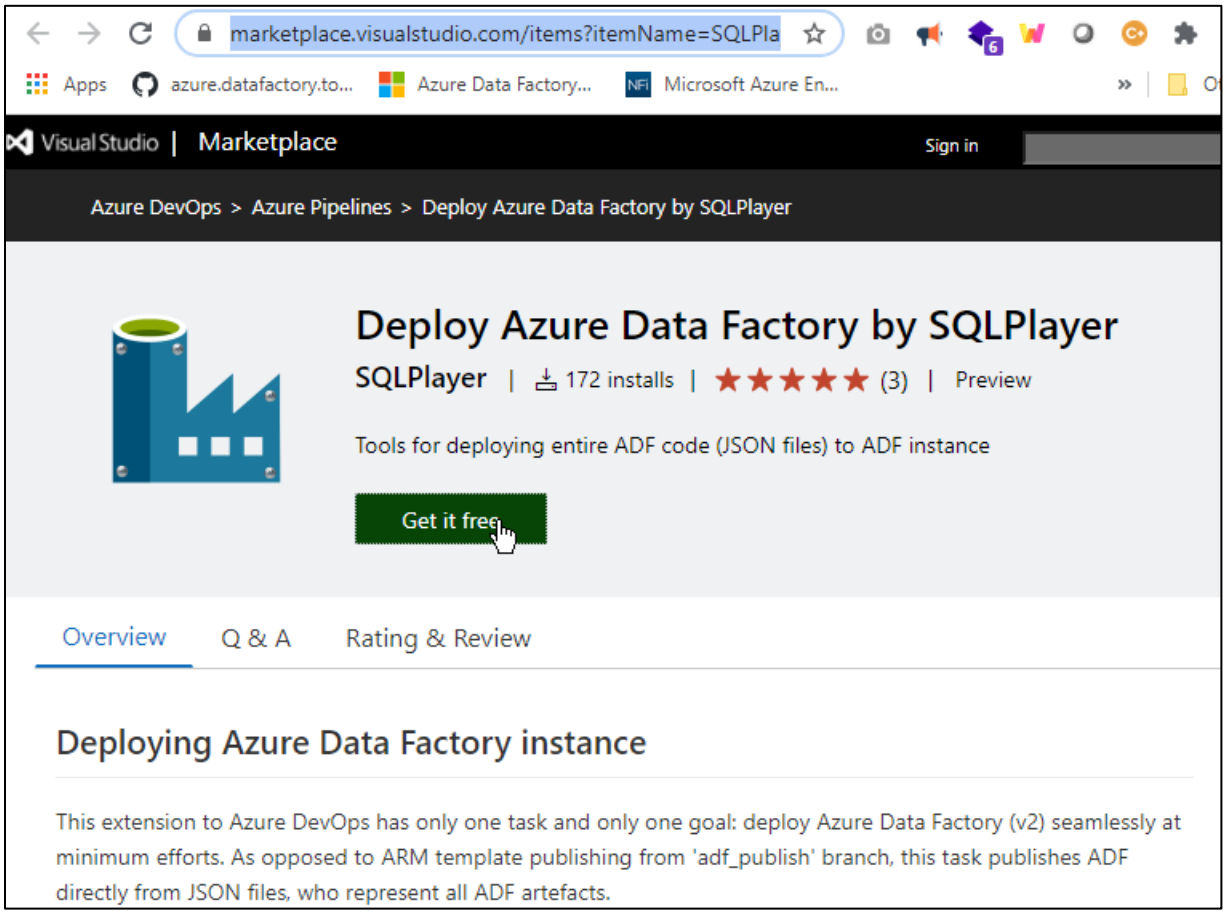



## Install Azure DevOps extension “Deploy Azure Data Factory” by SQL Player

This is an awesome tool, developed by Kamil Nowinski and made available for free. Kudos to Kamil.

In order to install it, make sure that you open a new tab in the same browser where you are connected already to the Azure Portal. In this new tab, navigate to url below.

https://marketplace.visualstudio.com/items?itemName=SQLPlayer.DataFactoryTools



Click **Get it Free** and follow the easy, simple installation steps. When prompted, make sure to select the Azure DevOps organization that you have been working with in these labs.

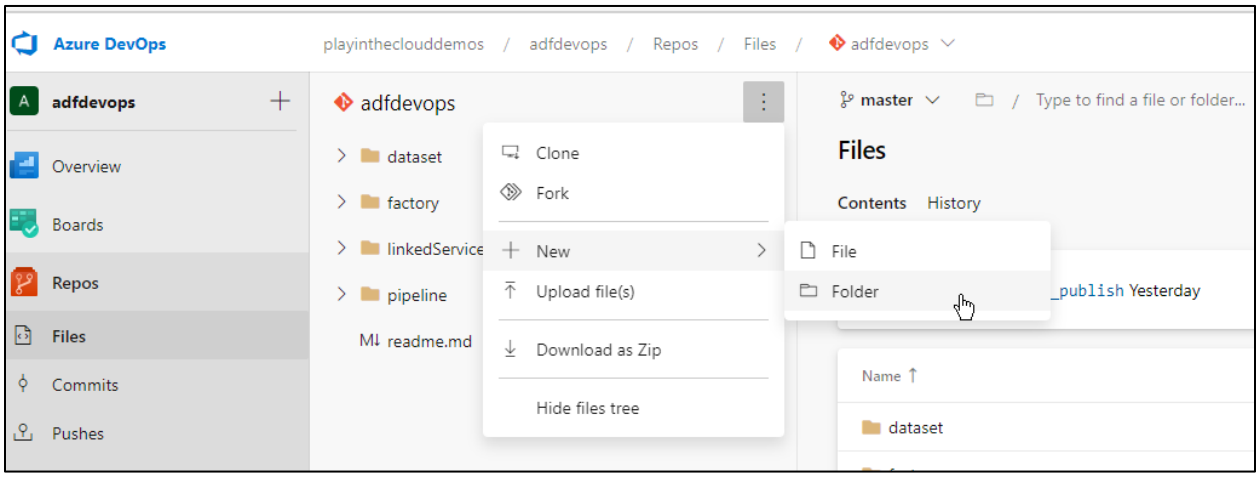
Full documentation about the tool is at https://github.com/SQLPlayer/azure.datafactory.tools.

## Create the environment's configuration file

You will remember that some configuration should change as we deploy ADF pipelines from environment to environment. With the previous method, the mechanism was ultimately based on the ARM Template Parameter file. Because, the new approach doesn't generate ARM templates, there is a need for another mechanism to implement similar functionality.

With the JSON files approach, a csv configuration file is used. Follow the steps below to implement it.

In Azure DevOps, in the **master** branch, create folder **\deployment** as a sibling folder of **\dataset**, **\factory**, etc.



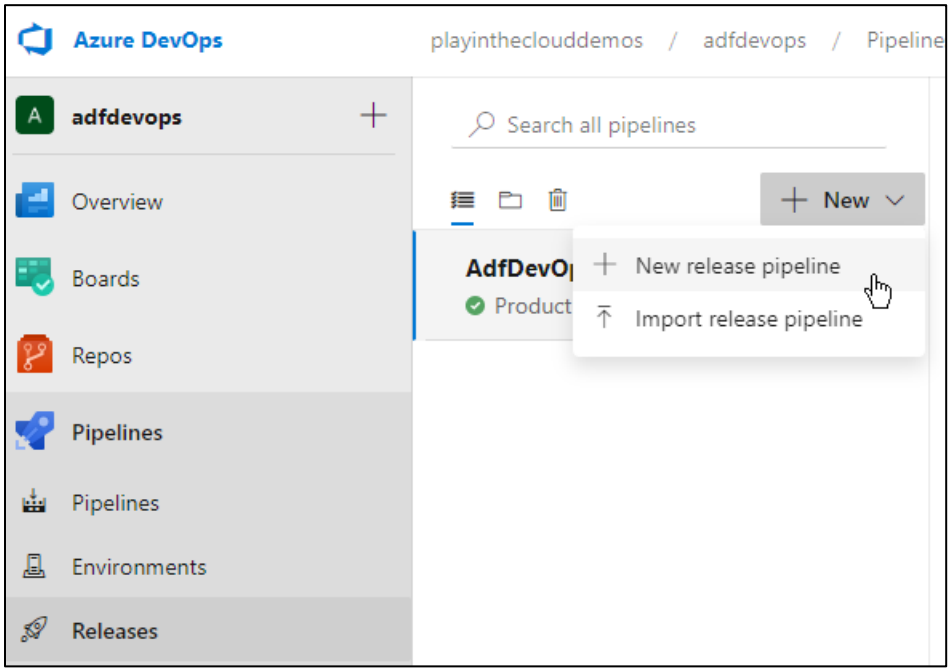

Name the file **config-uat.csv**, enter the content below but **make sure to replace "rd2020kvuat" by the name of the key vault in your Uat environment**.

*type,name,path,value*
*linkedService,AzureKeyVault1,typeProperties.baseUrl,"https://rd2020kvuat.vault.azure.net/"*

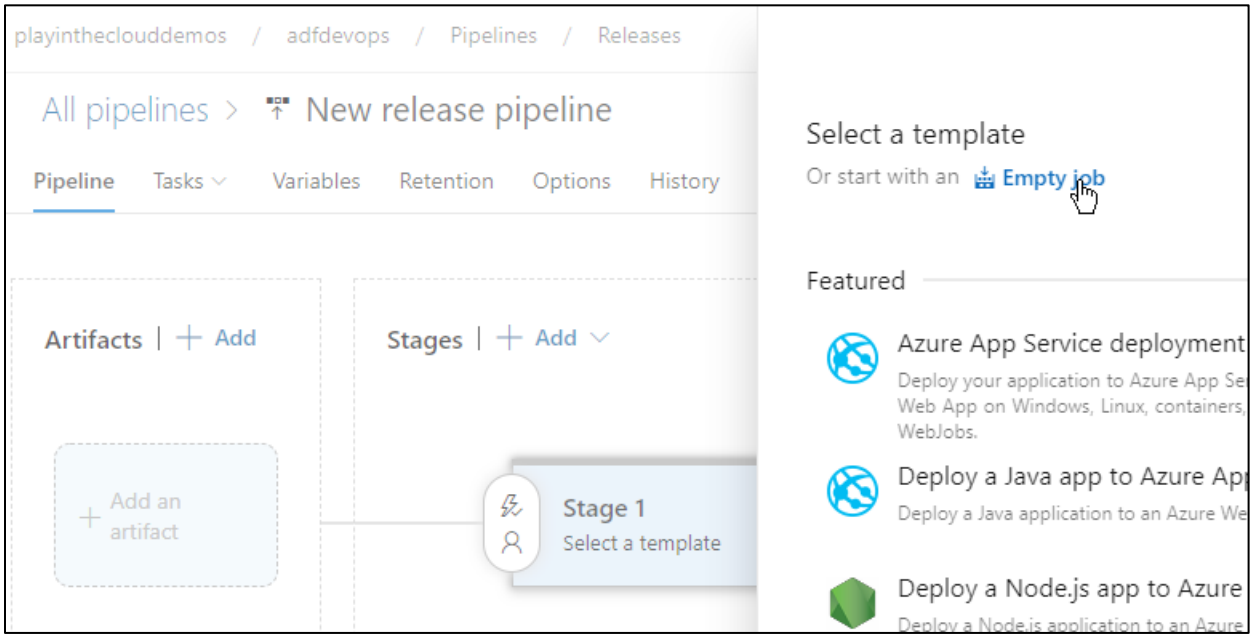
Click **Commit**.

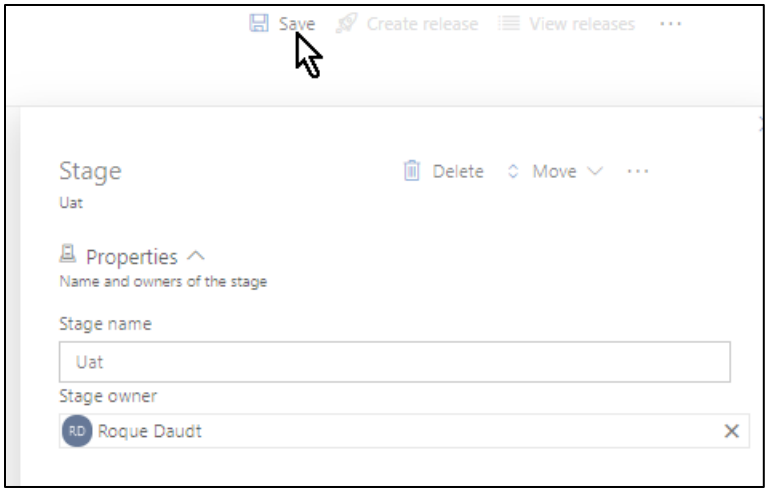
## Create a new release pipeline

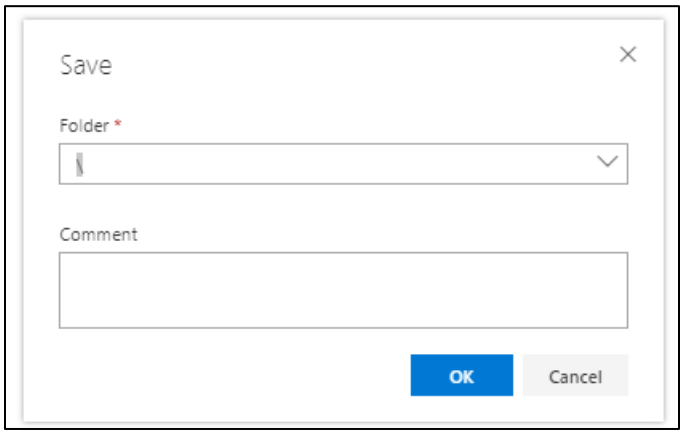
In Azure DevOps, create a new release pipeline.




As before, start by an **Empty job** and **Save**.



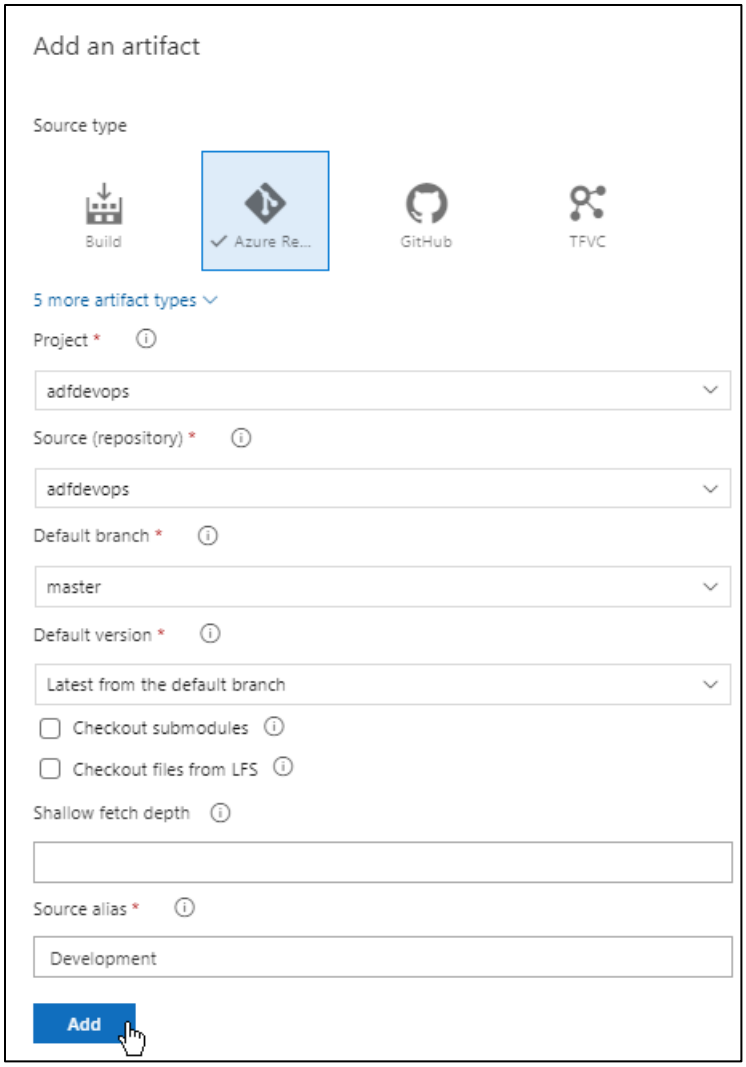


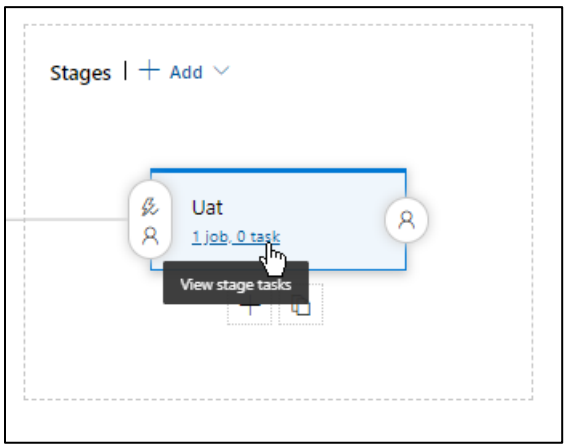

Click **OK**, close the **Stage** blade.

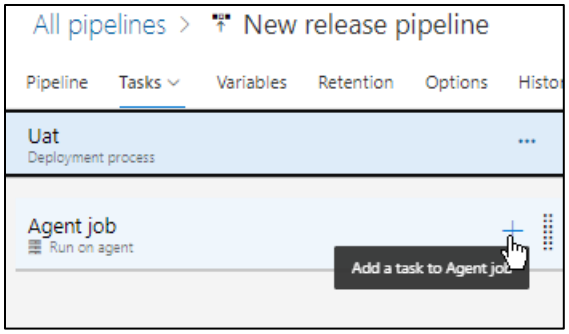
Add the source of the deployment, which is the set of json files in the master branch.


Select Azure Repo, enter the fields as below and click **Add**.

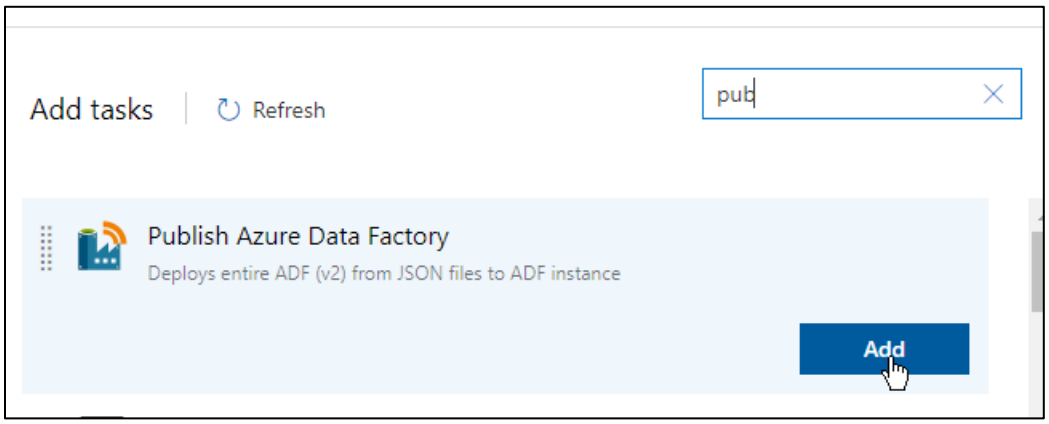



Let’s configure the deployment task now.

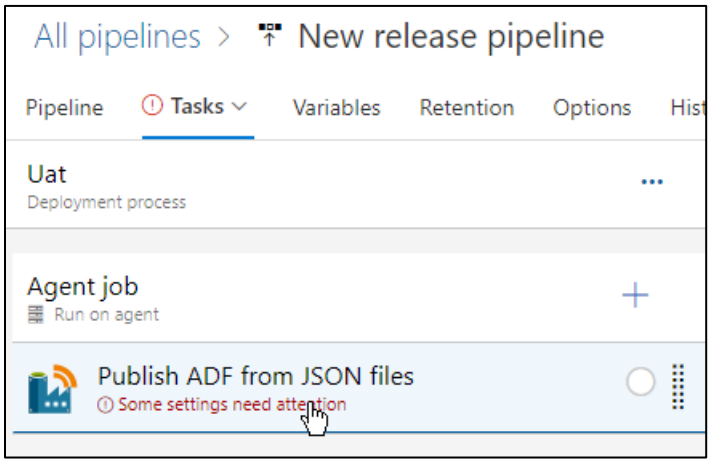




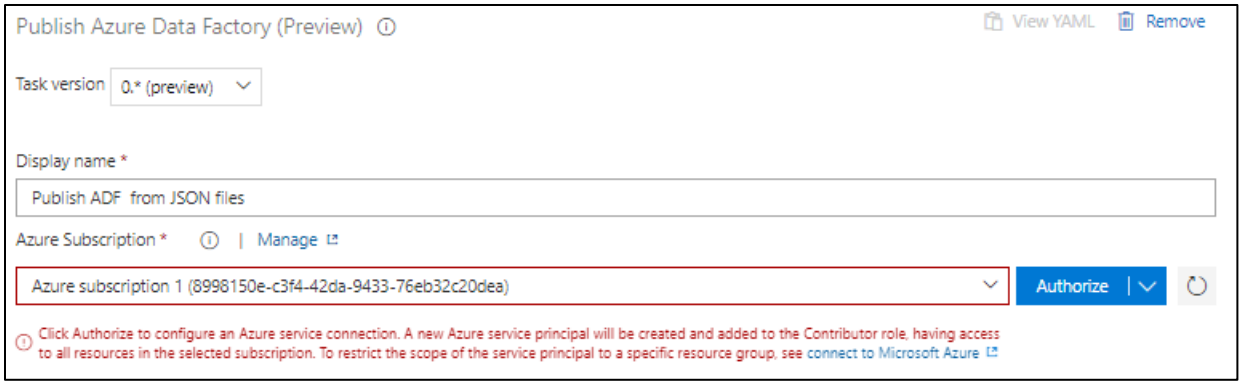

Look for the **Publish Azure Data Factory** task. This is the extension that was installed at the start of this lab.



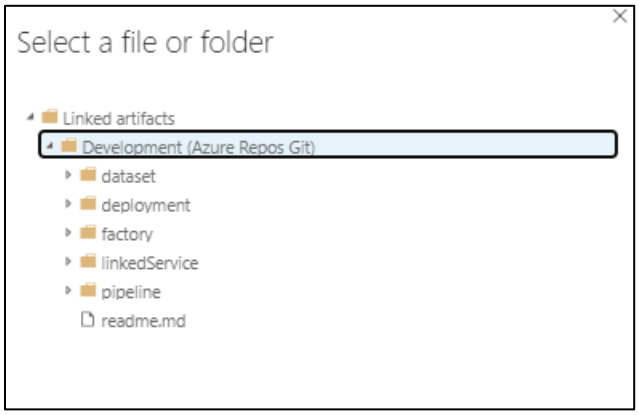

Configure the task.



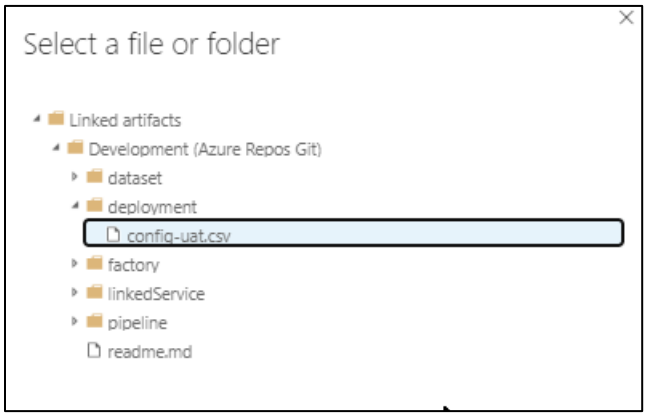

In the Azure Subscription field, select the subscription from the Service Connection group of options (you will see it as you use the pulldown control).



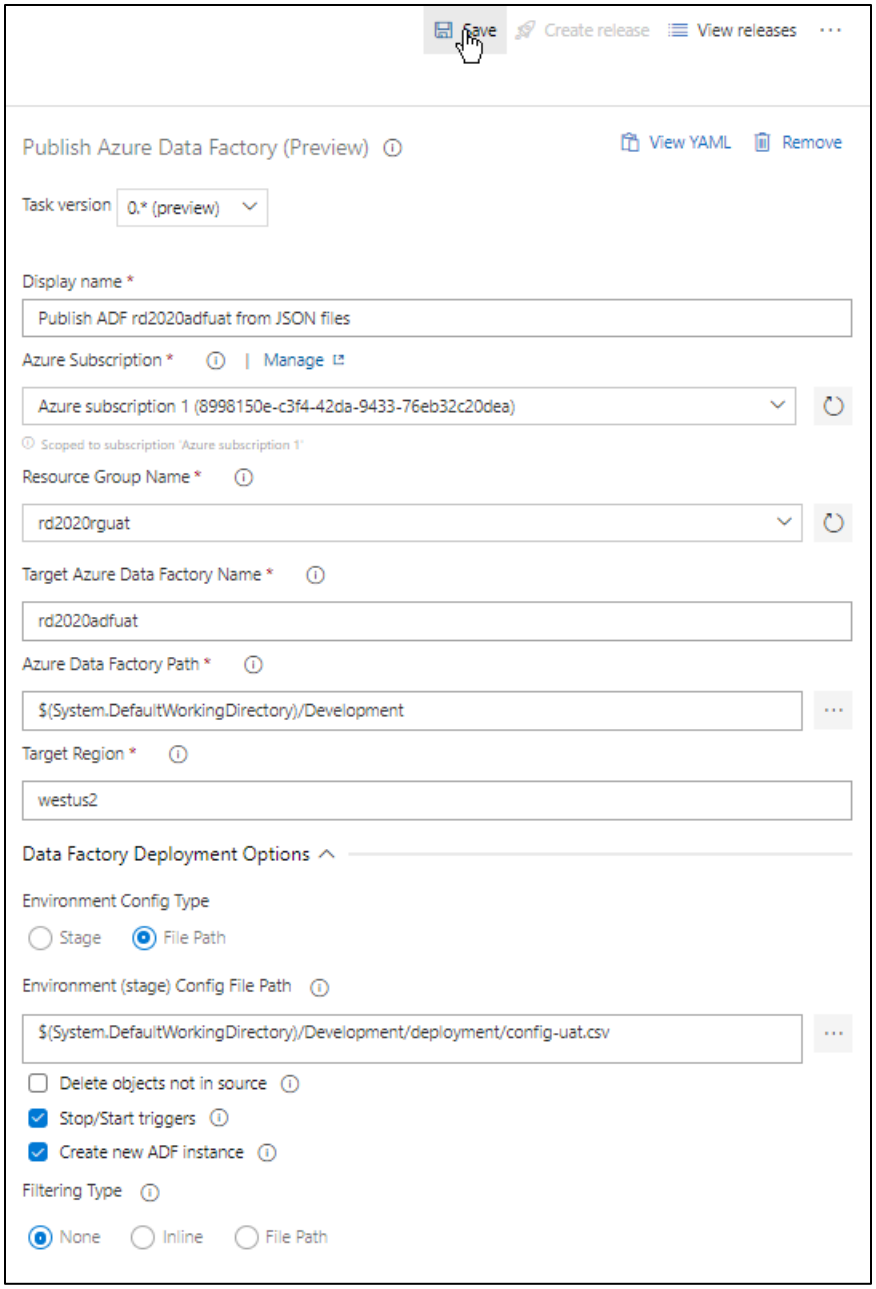

For the Azure Data Factory Path field, open the file explorer (…) and select the parent folder of the folders where the json files are.




For the Environment (config) File Path field, select the file below.



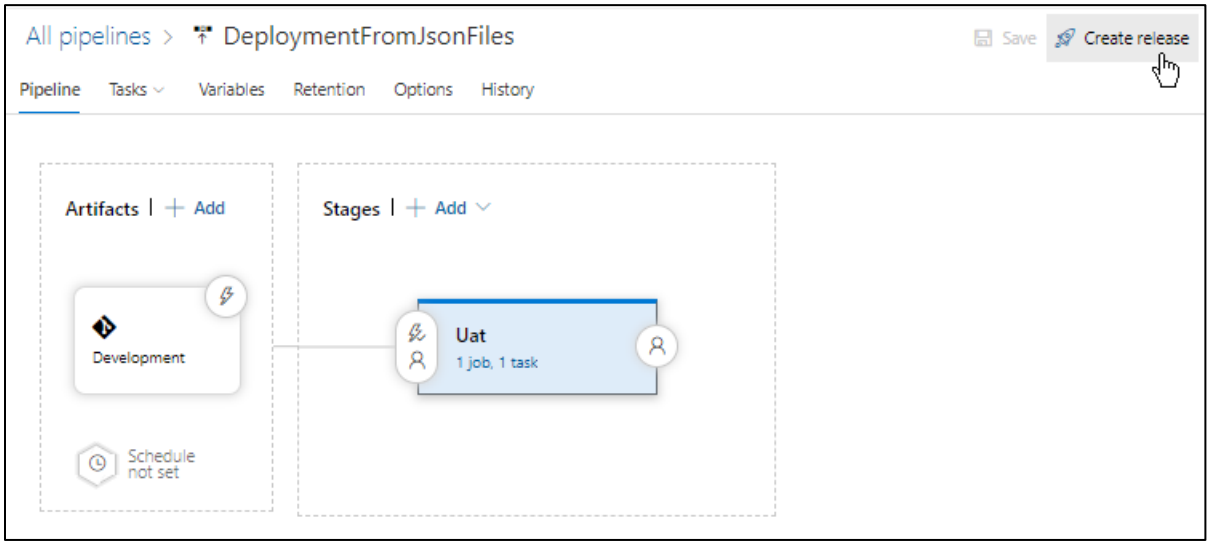

Remember to use the name of your Uat resource group, the name of your ADF Uat and the location of your choice. Save the work.



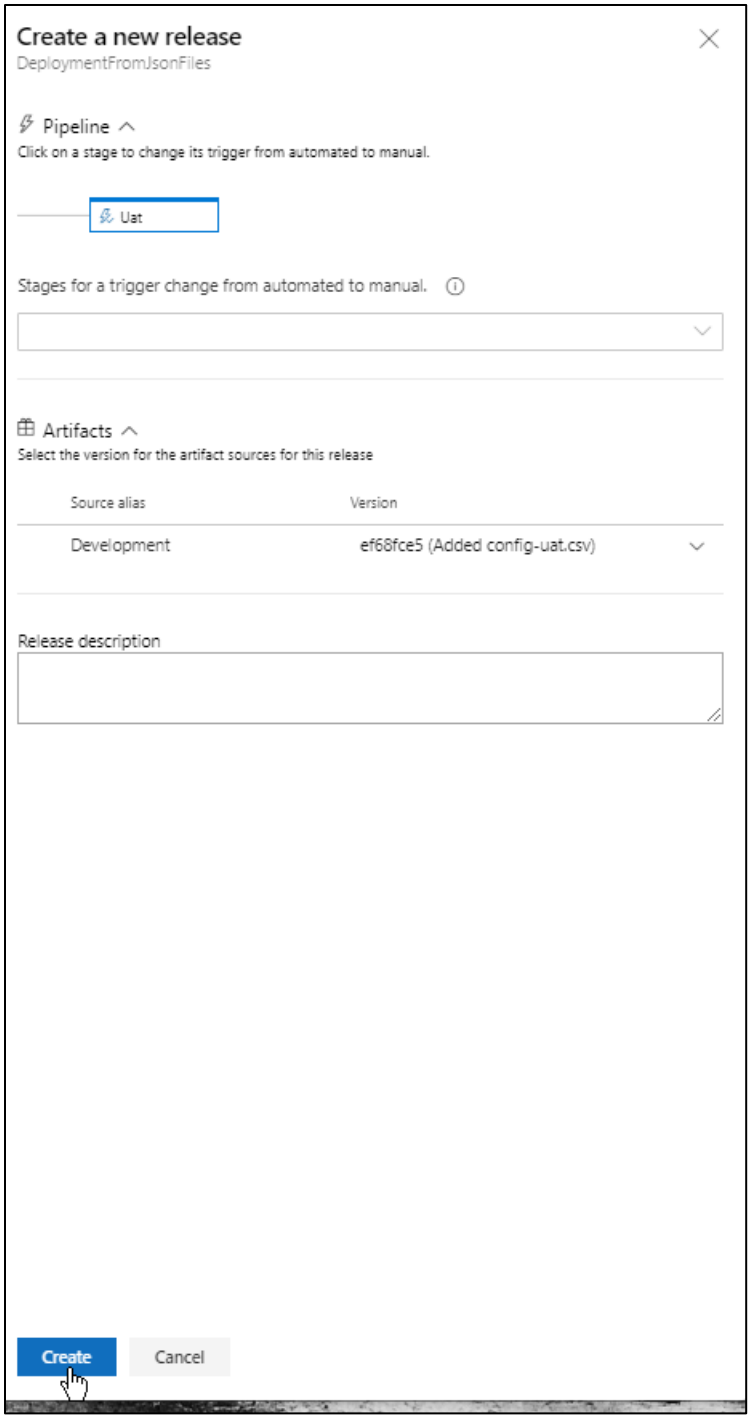


Optionally, give the pipeline a better name.

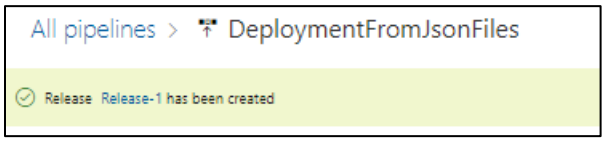

Create a release.



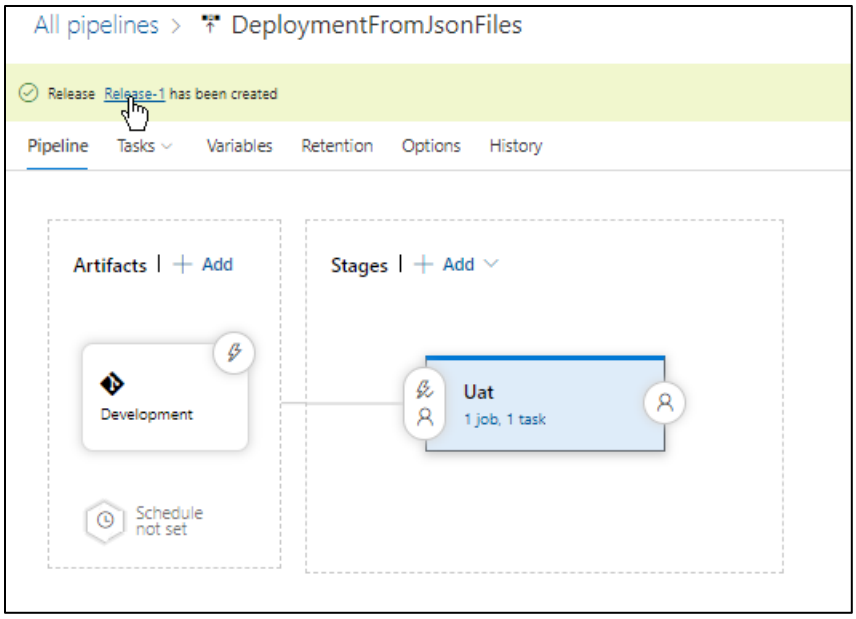

Click **Create**.



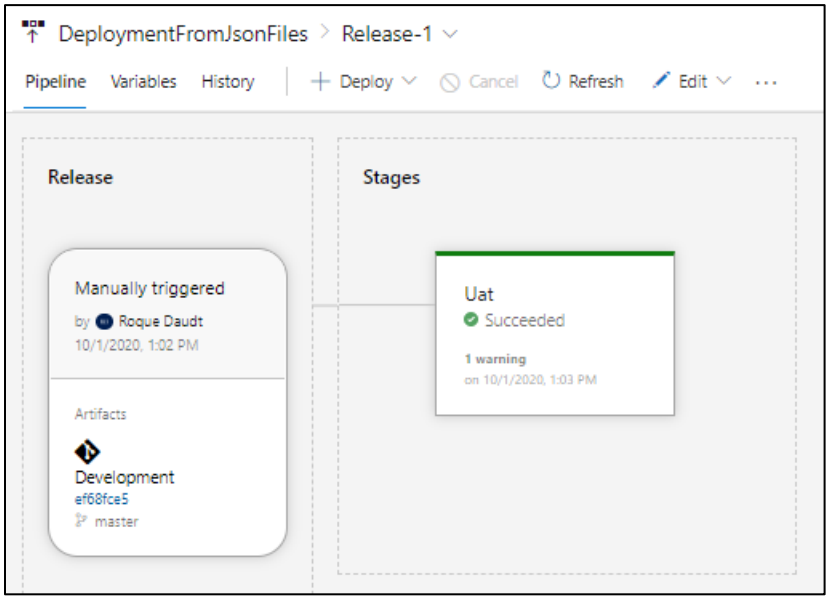

Watch the success message.



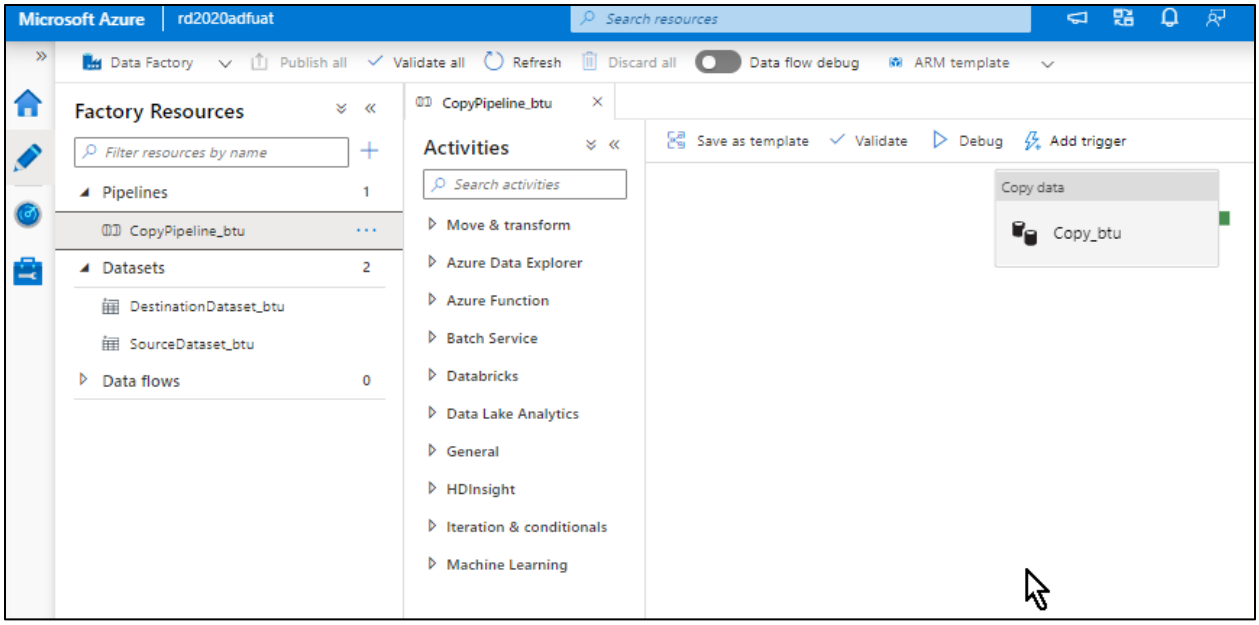

Click the link to run the release.




Wait and observe the results.




Once it succeeds, open ADF Uat to verify that the pipeline was indeed deployed.




Inspect the AzureKeyVault1 linked service and verify that the base URL points is correct (your UAT's KV).

## Summary

In this lab we deployed ADF objects from json files in the master (collaboration) branch. Additionally, we made it using a free too.

Check Nowinski's blog here for much more.

## Next

Go to **08 – Pipelines as code**.