

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

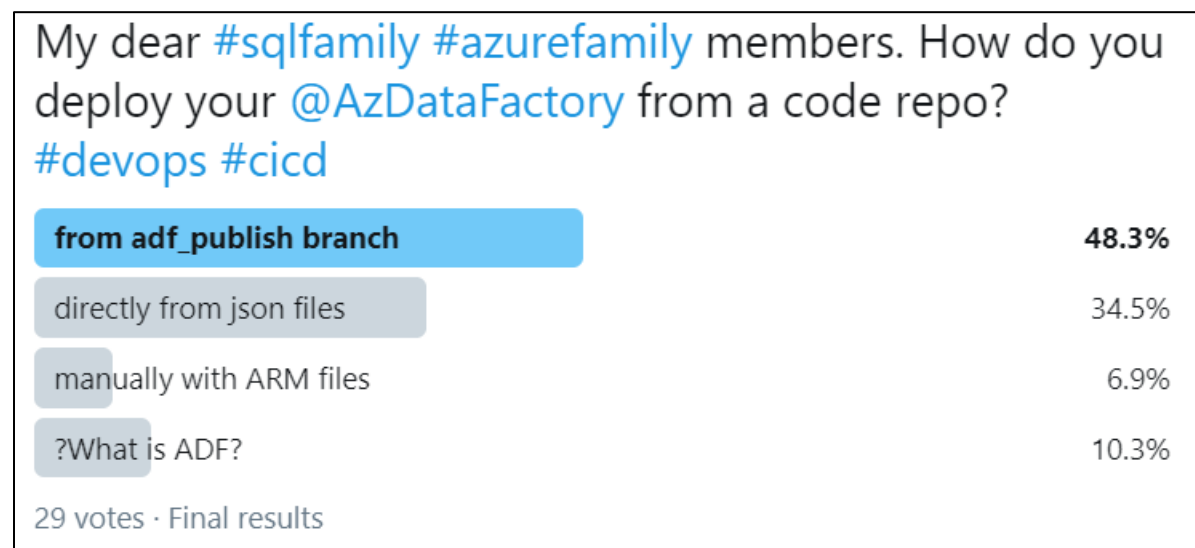
Azure Labs by Roque Daudt (rdaudt@yahoo.com)

## 07 – Deployment from JSON files

On March 25<sup>th</sup>, 2020, Kamil Nowinski ran a quick poll 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.

```

1 # script: createUATEnvironment.ps1
2 # functionality:
3 # - setup UAT azure environments in support of demo about
4 #   Azure Data Factory integration with Azure DevOps
5
6 # enter a different prefix. Ideally, use your initials as part of the prefix
7 $prefix = "rd2020"
8
9 # leave it as it is or enter the Azure location that best works for you
10 $loc = "West US 2"
11
12 # create the DEVELOPMENT environment
13 $env = "uat"
14 $rg = $prefix + "rg" + $env
15 # create resource group for uat environment services
16 New-AzResourceGroup -Name $rg -Location $loc
17 # create services for uat environment
18 New-AzResourceGroupDeployment -ResourceGroupName $rg -TemplateFile ".\azuredeploy.json" -prefix $prefix -env $env
19

```

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**

```

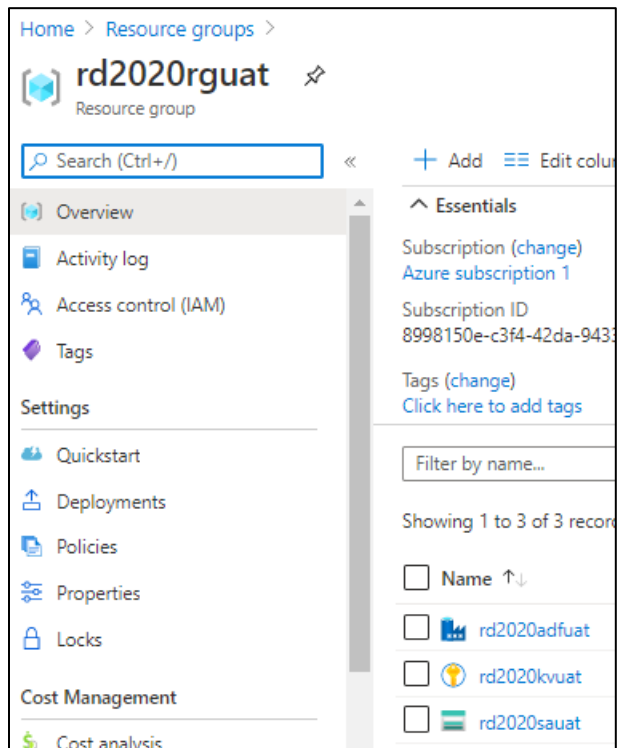
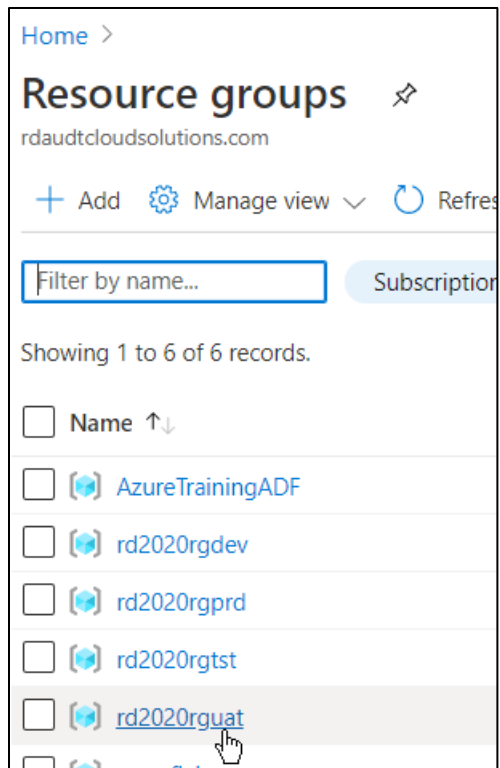
PS C:\adfdevops\resources> Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

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

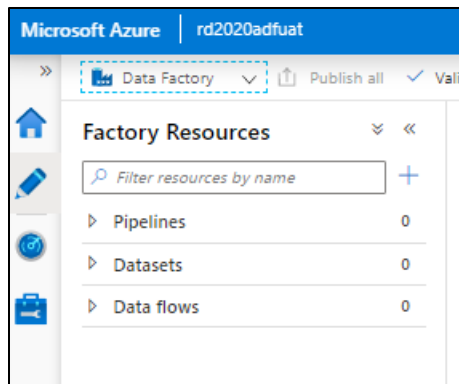
ResourceGroupName : rd2020rguat
Location           : westus2
ProvisioningState  : Succeeded
Tags               :
ResourceId         : /subscriptions/8998150e-c3f4-42da-9433-76eb32c20dea/resourceGroups/rd2020rguat

```

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.

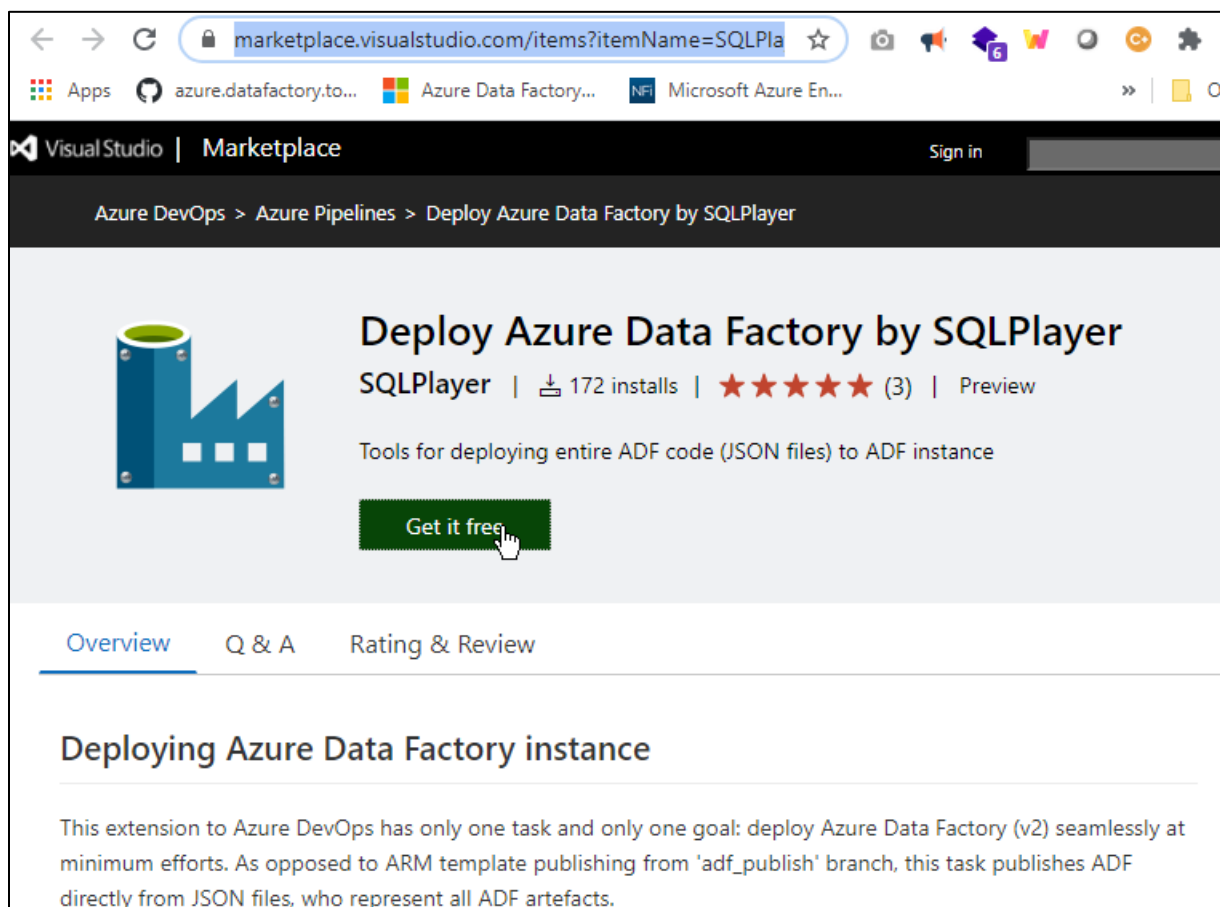


## Install Azure DevOps extension “Deploy Azure Data Factory” by SQL Player

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.

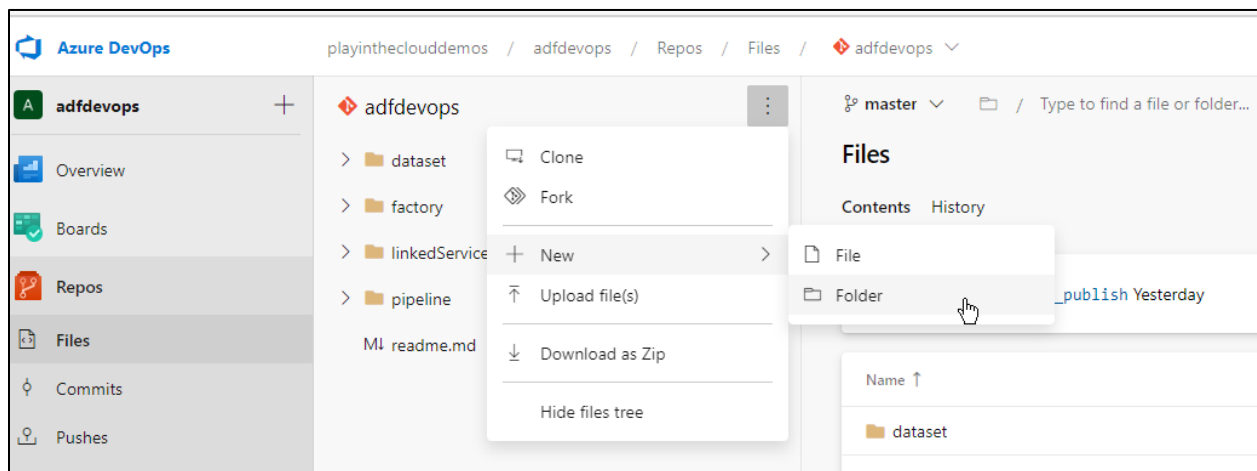
Full documentation about the tool is at <https://github.com/SQLPlayer/azure.datafactory.tools>.

## 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 "rd2020kkuat"** by the **name of the key vault in your Uat environment**.

*type,name,path,value*

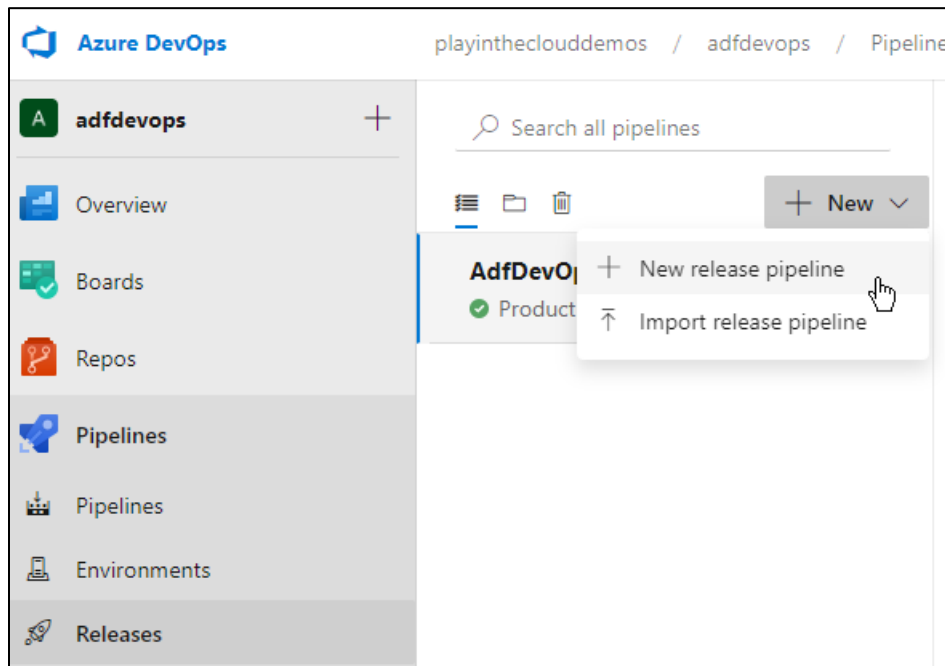
*linkedService,AzureKeyVault1,typeProperties.baseUrl,"https://rd2020kkuat.vault.azure.net/"*

Click **Commit**.

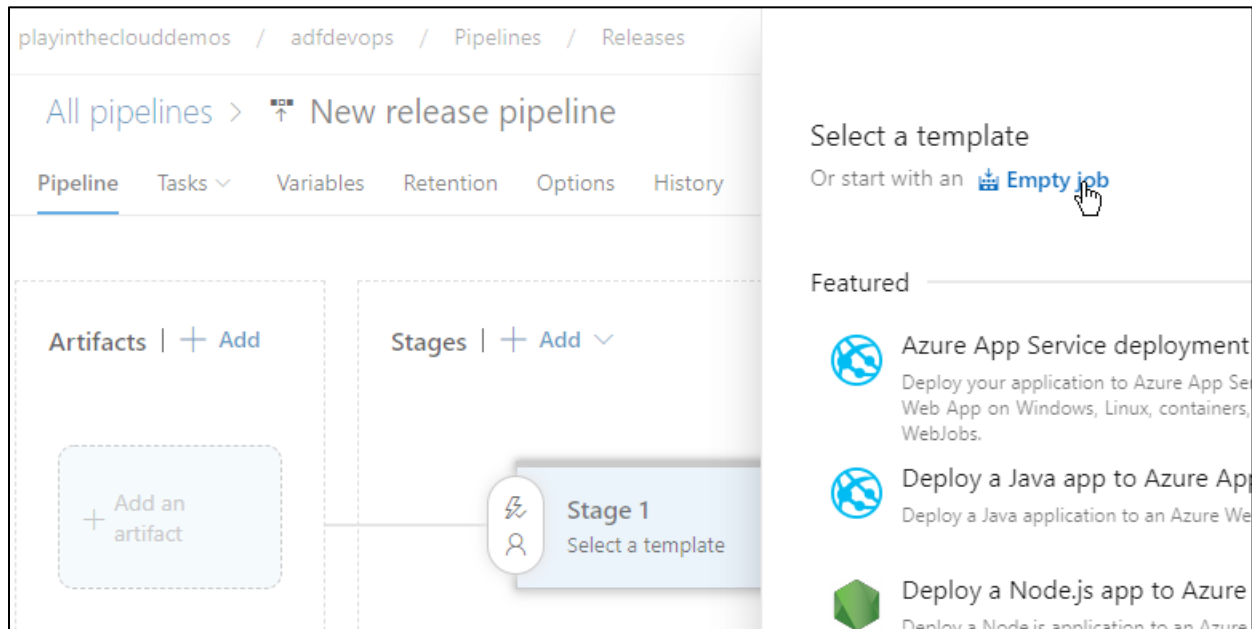


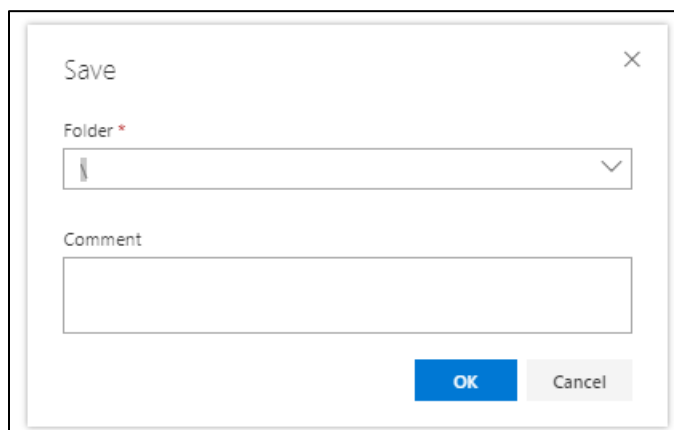
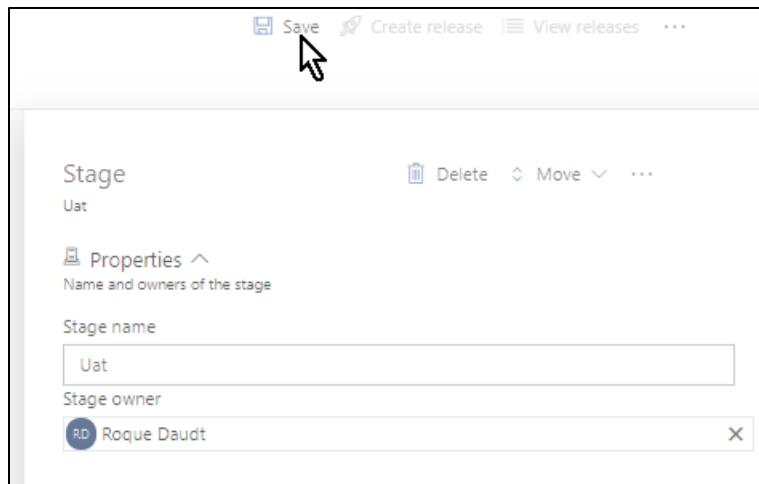
## Create a new release pipeline

In Azure DevOps, create a new release pipeline.



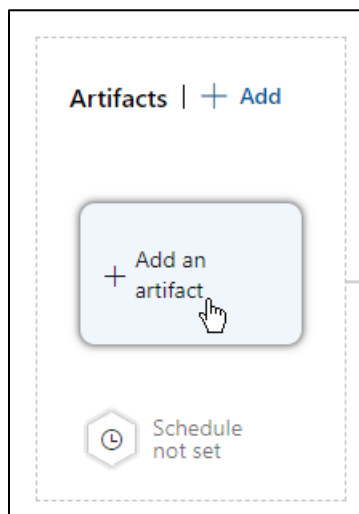
As before, start by an **Empty job** and **Save**.





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**.


### Add an artifact

Source type

  
Build

  
✓ Azure Re...

  
GitHub

  
TFVC

5 more artifact types ▾

Project \* ⓘ

adfddevops ▾

Source (repository) \* ⓘ

adfddevops ▾

Default branch \* ⓘ

master ▾

Default version \* ⓘ

Latest from the default branch ▾

☐ Checkout submodules ⓘ

☐ Checkout files from LFS ⓘ

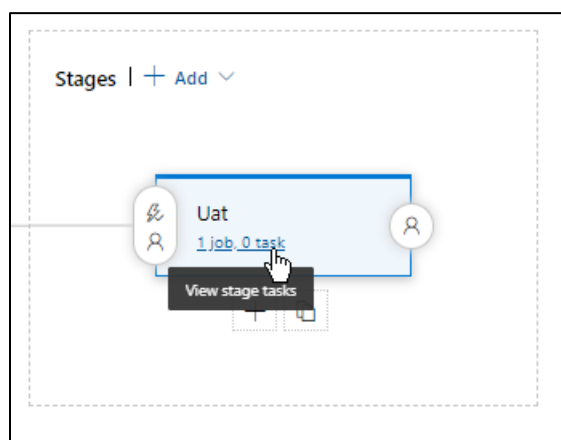
Shallow fetch depth ⓘ

Source alias \* ⓘ

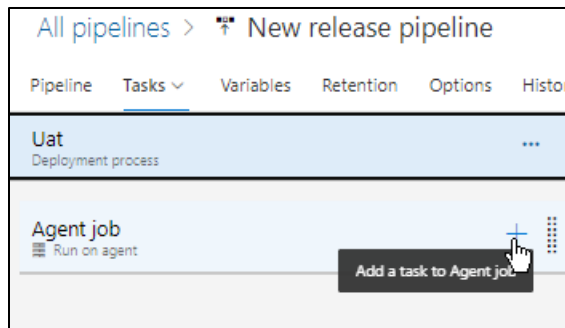
Development

**Add**

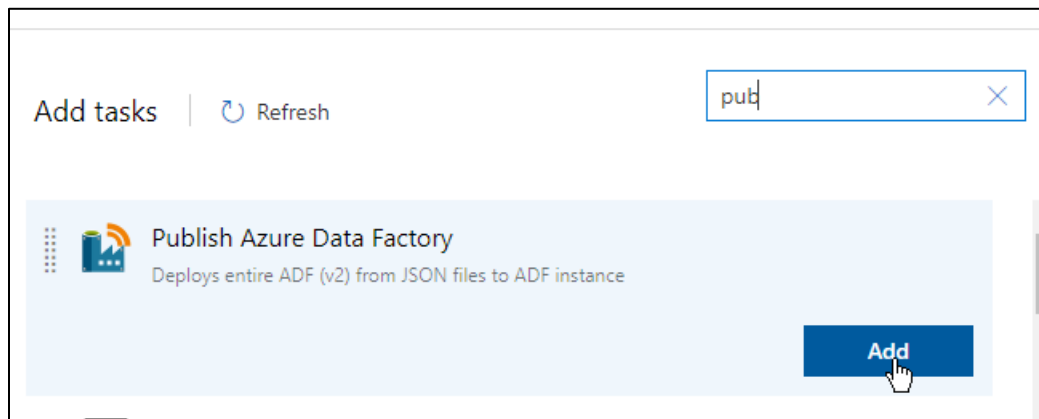
Let's configure the deployment task now.



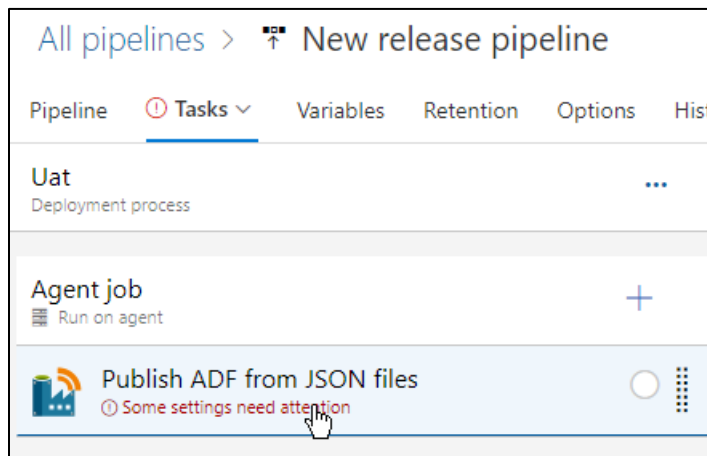




Look for the **Publish Azure Data Factory** task. This is the extension that was installed at the start of this lab.



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).

Publish Azure Data Factory (Preview) ⓘ

Task version: 0.\* (preview) ▼

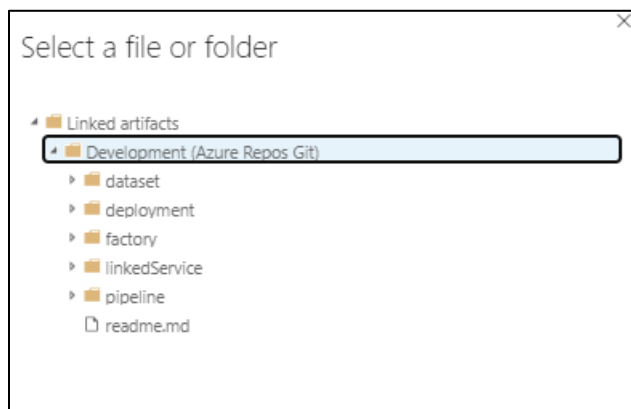
Display name \*: Publish ADF from JSON files

Azure Subscription \*: ⓘ | Manage ⓘ

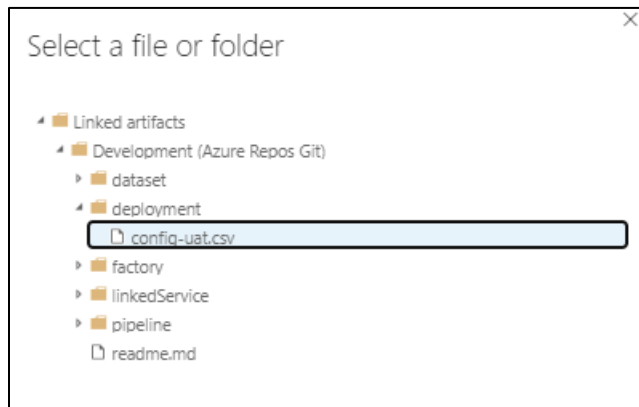
Azure subscription 1 (8998150e-c3f4-42da-9433-76eb32c20dea) ▼ Authorize ▼ ↻

ⓘ Click Authorize to configure an Azure service connection. A new Azure service principal will be created and added to the Contributor role, having access to all resources in the selected subscription. To restrict the scope of the service principal to a specific resource group, see [connect to Microsoft Azure](#) ⓘ

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.

[Save](#)
[Create release](#)
[View releases](#)
...

---

Publish Azure Data Factory (Preview) ⓘ
 [View YAML](#)
[Remove](#)

Task version: 0.\* (preview) ▼

Display name \*  
 Publish ADF rd2020adfuat from JSON files

Azure Subscription \* ⓘ | [Manage](#) ↗  
 Azure subscription 1 (8998150e-c3f4-42da-9433-76eb32c20dea) ▼ ↻

ⓘ Scoped to subscription 'Azure subscription 1'

Resource Group Name \* ⓘ  
 rd2020rguat ▼ ↻

Target Azure Data Factory Name \* ⓘ  
 rd2020adfuat

Azure Data Factory Path \* ⓘ  
 \$(System.DefaultWorkingDirectory)/Development ...

Target Region \* ⓘ  
 westus2

Data Factory Deployment Options ^

Environment Config Type  
☐ Stage ☒ File Path

Environment (stage) Config File Path ⓘ  
 \$(System.DefaultWorkingDirectory)/Development/deployment/config-uat.csv ...

☐ Delete objects not in source ⓘ  
☒ Stop/Start triggers ⓘ  
☒ Create new ADF instance ⓘ

Filtering Type ⓘ  
☒ None ☐ Inline ☐ File Path

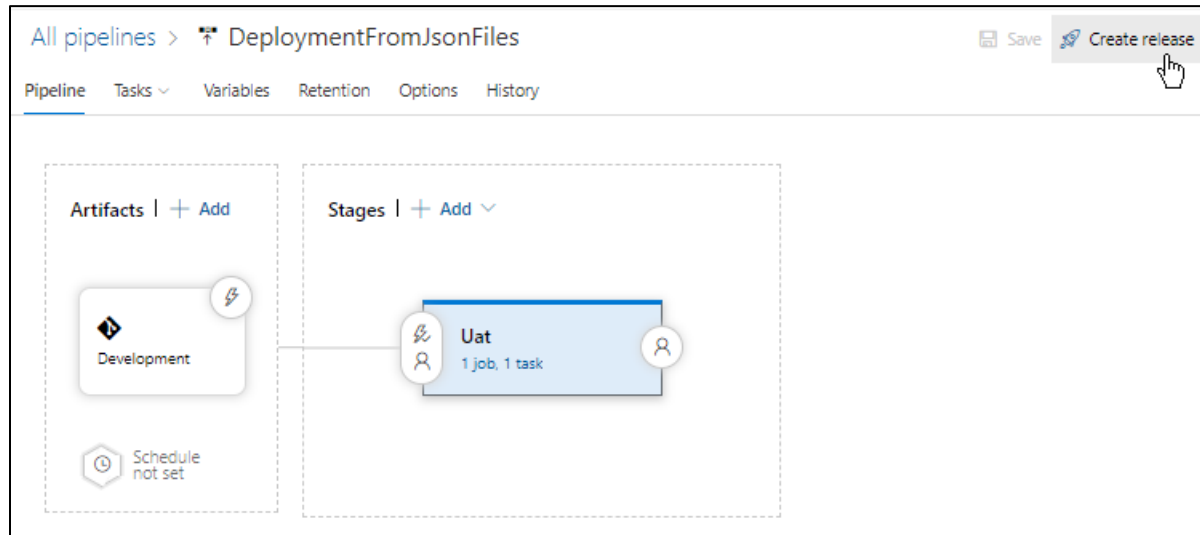
Save

Comment

Optionally, give the pipeline a better name.



Create a release.



Click **Create**.

### Create a new release

DeploymentFromJsonFiles

Pipeline ^

Click on a stage to change its trigger from automated to manual.

Uat

Stages for a trigger change from automated to manual. ⓘ

Artifacts ^

Select the version for the artifact sources for this release

Source alias	Version
Development	ef68fce5 (Added config-uat.csv)

Release description

Create

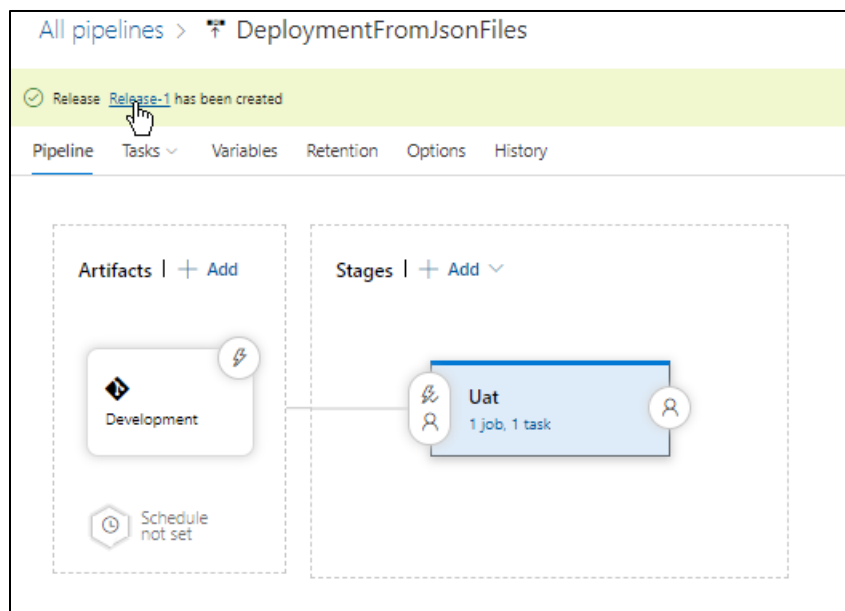
Cancel

Watch the success message.

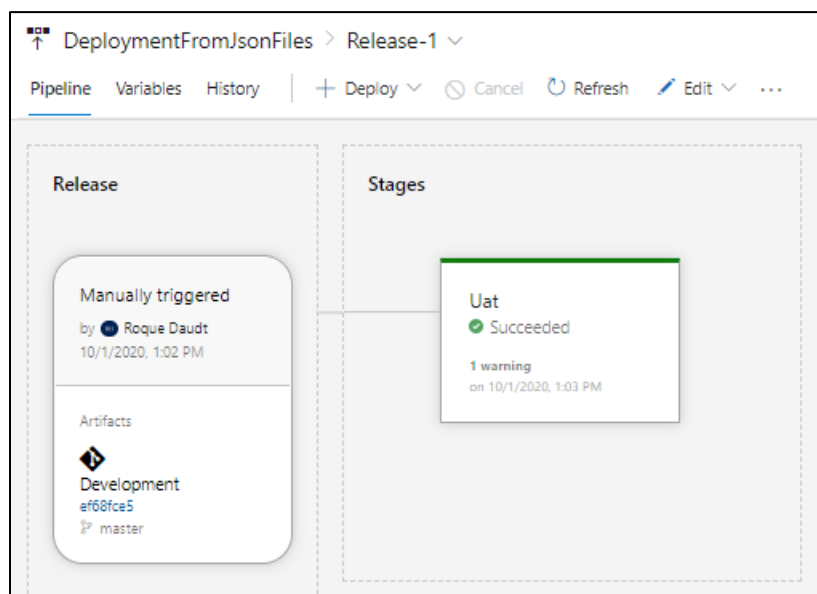
All pipelines > DeploymentFromJsonFiles

Release **Release-1** has been created

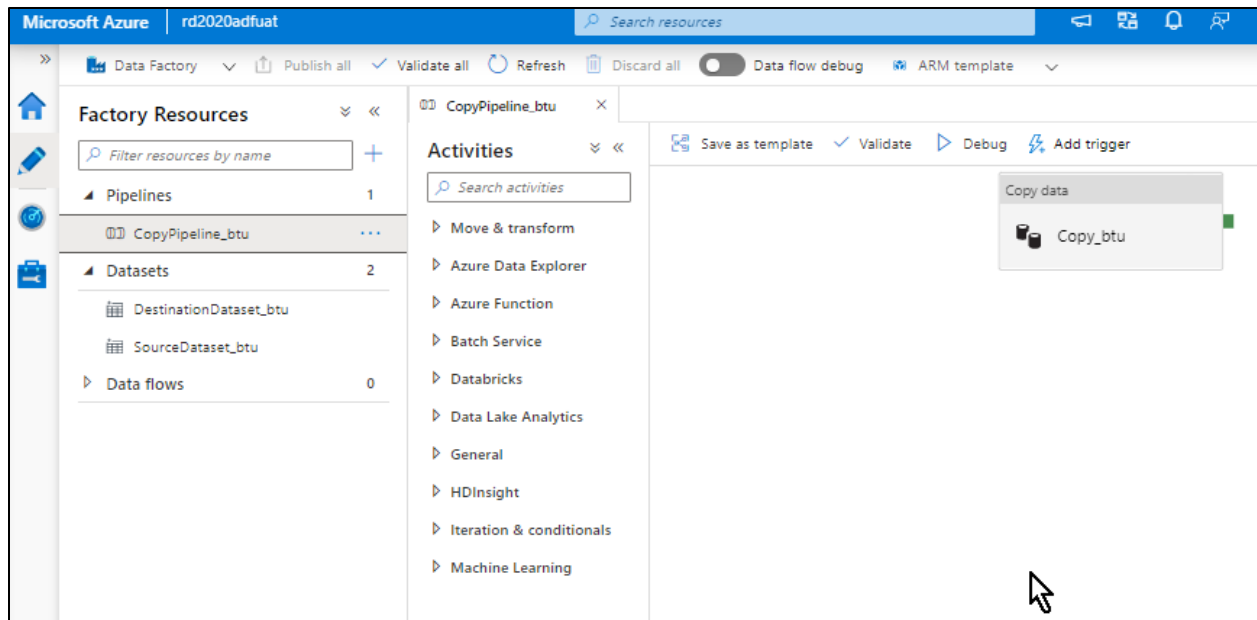
Click the link to run the release.



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**.