How to bring your Data Science Project in production

Using Azure Databricks with Spark, Azure ML and Azure DevOps

René Bremer Follow
Jan 20, 2019 · 10 min read

1. Introduction

A lot of companies struggle to bring their data science projects into production. A common issue is that the closer the model is to production, the harder it is to answer the following question:

- Why did the model predict this?

Having a build/release pipeline for these data science project can help to answer this question. It enables you to trace back that:

- Model M was trained on dataset D with algorithm A by person P.
- Model M was deployed in production in release R on time T

This audit trail is essential for every model running in production and is required in a lot of industries, e.g. finance.

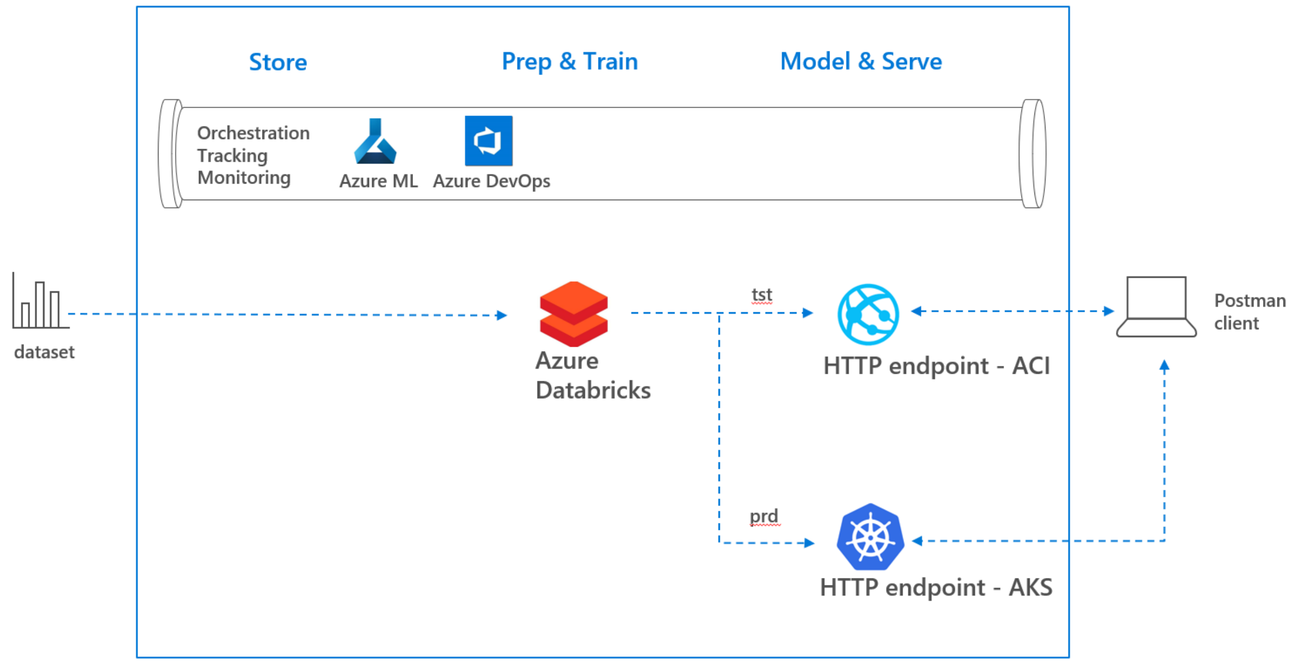
Last major update of blog/code on github: Jan 12, 2020

2. Objective

In this tutorial, a build/release pipeline for a machine learning project is created as follows:

- An HTTP endpoint is created that predicts if the income of a person is higher or lower than 50k per year using features as age, hours of week working, education.
- Azure Databricks with Spark, Azure ML and Azure DevOps are used to create a model and endpoint. Azure Container Instance (ACI) is used for testing, Azure Kubernetes Service (AKS) as production environment.

The project can be depicted in the following high level overview:



2. High level overview

In the remainder of this blog, the following steps will be executed:

- 3. Prerequisites
- 4. Create machine learning model in Azure Databricks
- 5. Manage model in Azure Machine Learning Service
- 6,7. Build and release model in Azure DevOps
- 8. Conclusion

The follow-up of the blog can be found here in which security is embedded in the build/release pipeline. Furthermore, the details of the audit trail are discussed in this blog. Finally, if you interested how to use Azure Databricks with Azure Data Factory, refer to this blog.

3. Prerequisites

The following resources are required in this tutorial:

- Azure Databricks
- Azure Machine Learning Service
- Azure DevOps

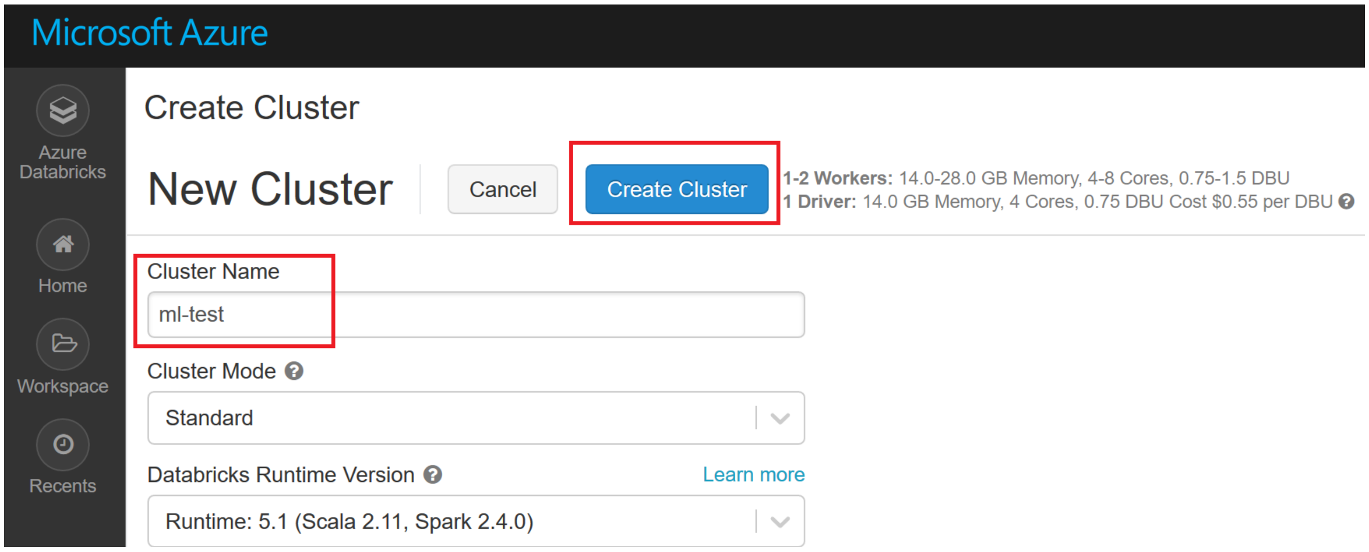
4. Create machine learning model in Azure Databricks

Azure Databricks is an Apache Spark-based analytics platform optimized for Azure. It can be used for many analytical workloads, amongst others machine learning and deep learning. In this step, the following is done:

- 4a. Create new cluster
- 4b. Import notebook
- 4c. Run notebook

4a. Create new cluster

Start your Azure Databricks workspace and go to Cluster. Create a new cluster with the following settings:

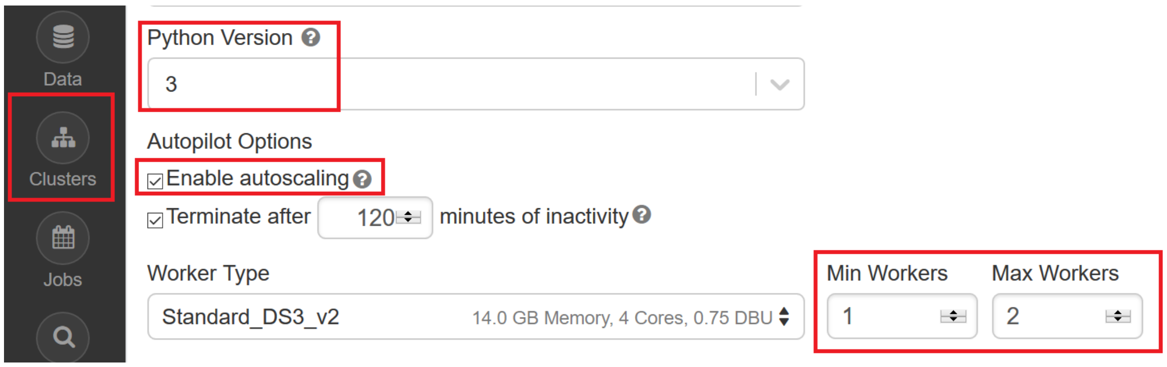


1-2 Workers: 14.0-28.0 GB Memory, 4-8 Cores, 0.75-1.5 DBU
1 Driver: 14.0 GB Memory, 4 Cores, 0.75 DBU Cost $0.55 per DBU

Cluster Name: ml-test

Cluster Mode: Standard

Databricks Runtime Version: Runtime: 5.1 (Scala 2.11, Spark 2.4.0)



Python Version: 3
Autopilot Options
☑ Enable autoscaling
☑ Terminate after 120 minutes of inactivity
Worker Type: Standard_DS3_v2 14.0 GB Memory, 4 Cores, 0.75 DBU
Min Workers: 1
Max Workers: 2

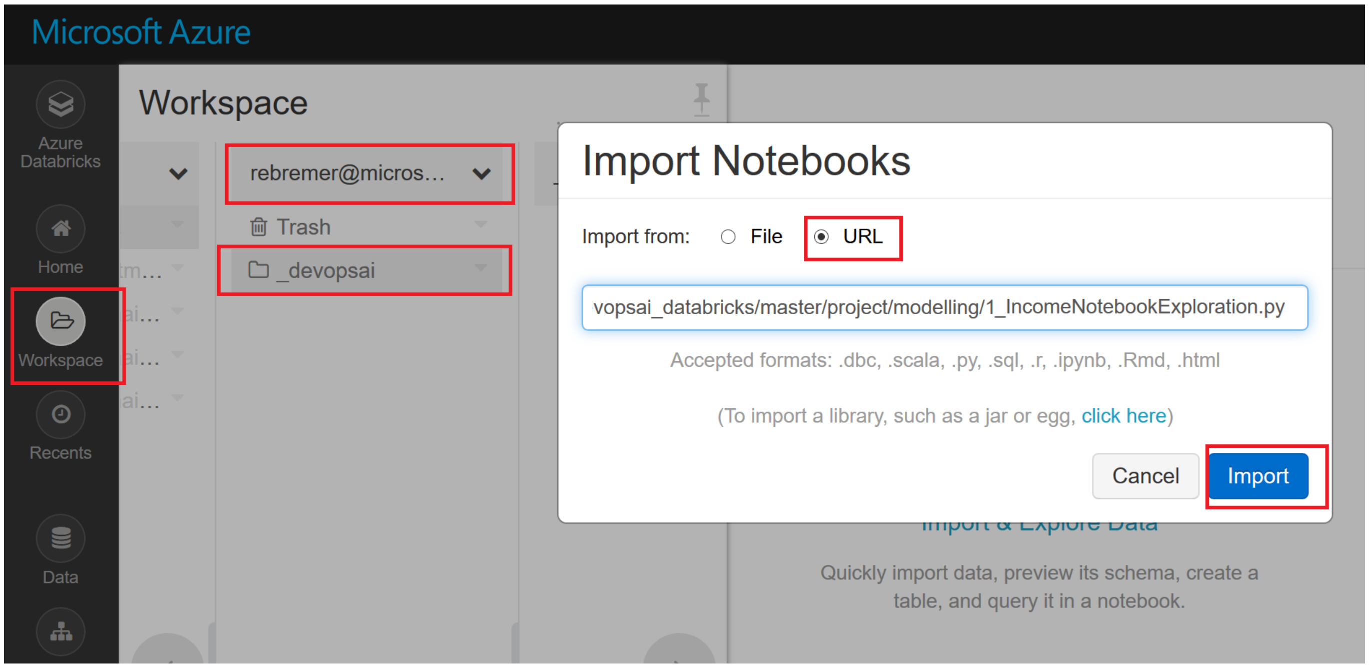
4a1. Create cluster

4b. Import notebook

Go to your Azure Databricks workspace, right-click and then select import. In the radio button, select to import the following notebook using URL:

```
https://raw.githubusercontent.com/rebremer/devopsai_databricks/maste
r/project/modelling/1_IncomeNotebookExploration.py
```

See also picture below:



Import Notebooks

Import from: ○ File [✓ ● URL]

vopsai_databricks/master/project/modelling/1_IncomeNotebookExploration.py

Accepted formats: .dbc, .scala, .py, .sql, .r, .ipynb, .Rmd, .html

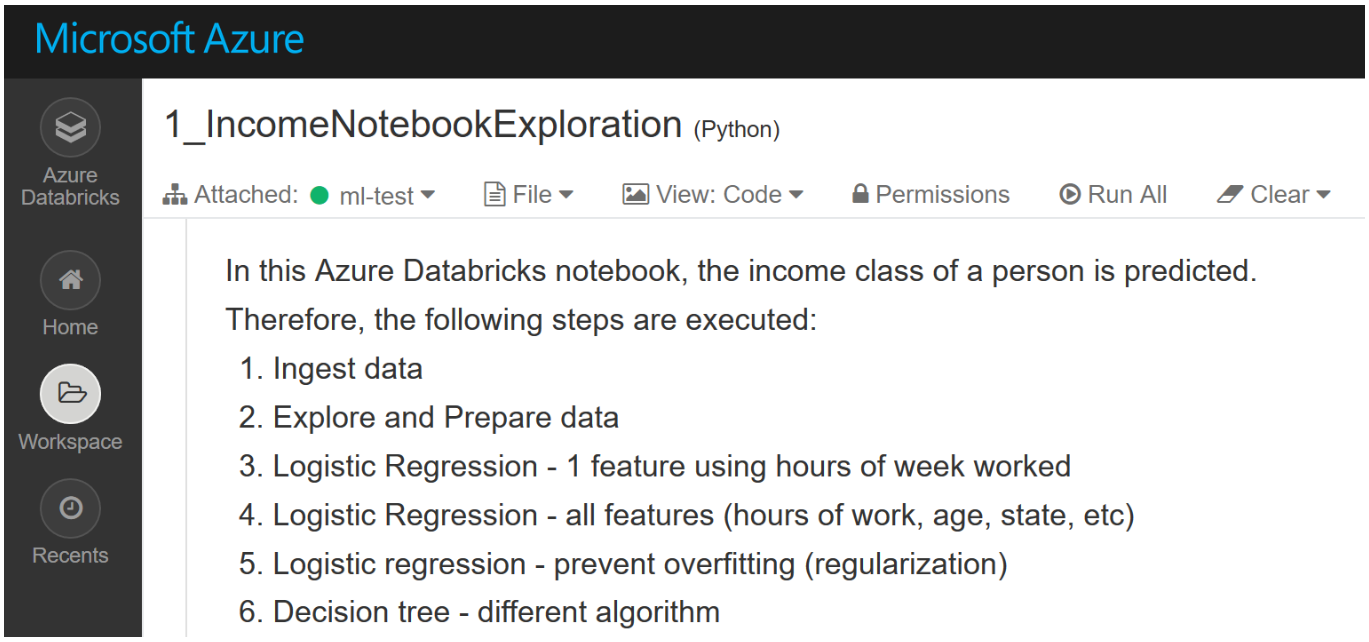
(To import a library, such as a jar or egg, click here)

Cancel Import

4b1. Import notebook

4c. Run notebook

Select the notebook you imported in 4b and attach the notebook to the cluster you created in 4a. Make sure that the cluster is running and otherwise start it. Read the steps in the notebook, in which the data is explored and several settings and algorithms are tried to create a model that predicts the income class of a person. Walk through the notebook cell by cell by using shortcut SHIFT+ENTER.



In this Azure Databricks notebook, the income class of a person is predicted. Therefore, the following steps are executed:

1. Ingest data
2. Explore and Prepare data
3. Logistic Regression - 1 feature using hours of week worked
4. Logistic Regression - all features (hours of work, age, state, etc)
5. Logistic regression - prevent overfitting (regularization)
6. Decision tree - different algorithm

4c1. Steps in notebook

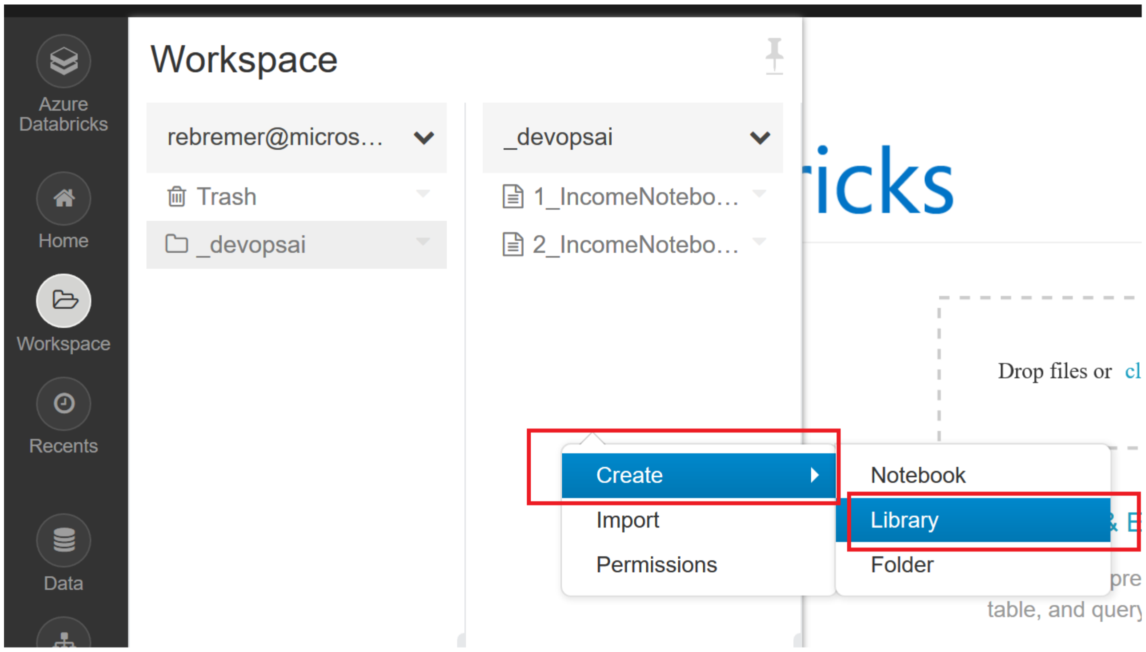
5. Manage model in Azure Machine Learning Service

Azure Machine Learning Service is a cloud service that you use to train, deploy, automate, and manage machine learning models. In this context, the model that was created in previous step will be added to your AMLS instance. The following steps will be executed

- 5a. Add library to Databricks cluster
- 5b. Import notebook using Azure ML to Azure Databricks
- 5c. Review results in Azure ML

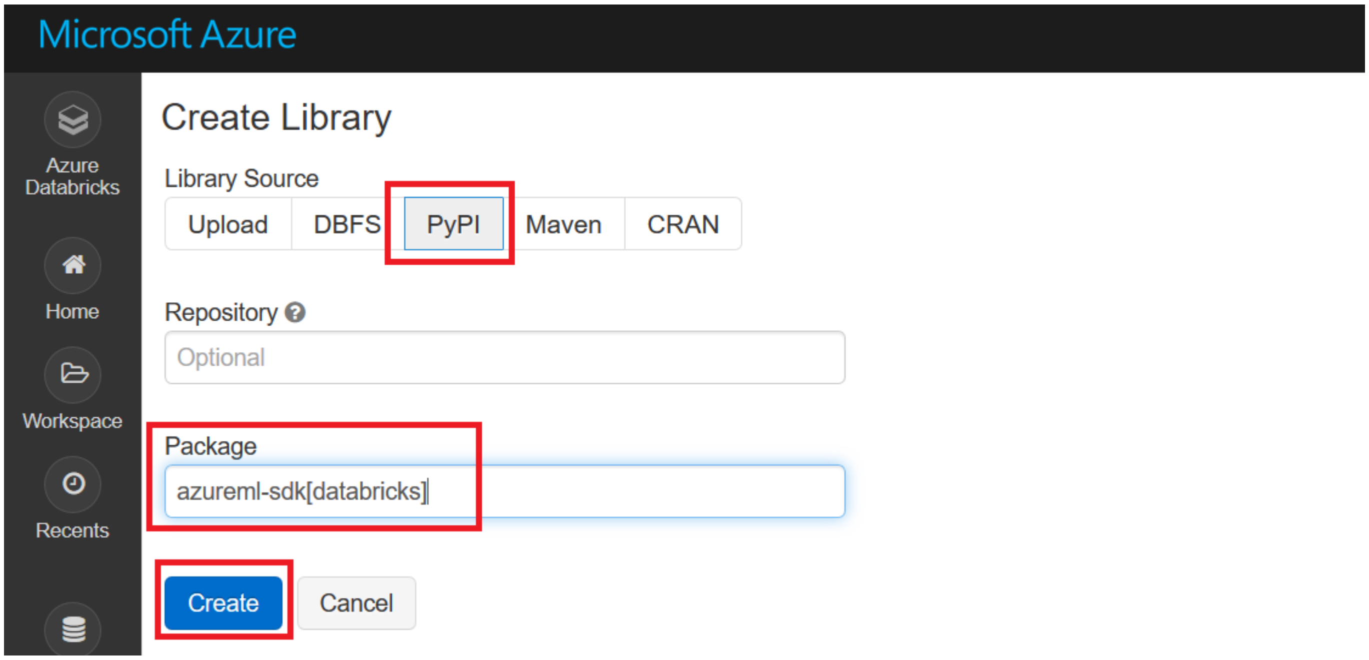
5a. Add library to Databricks cluster

Right click in your workspace and select to "create library"



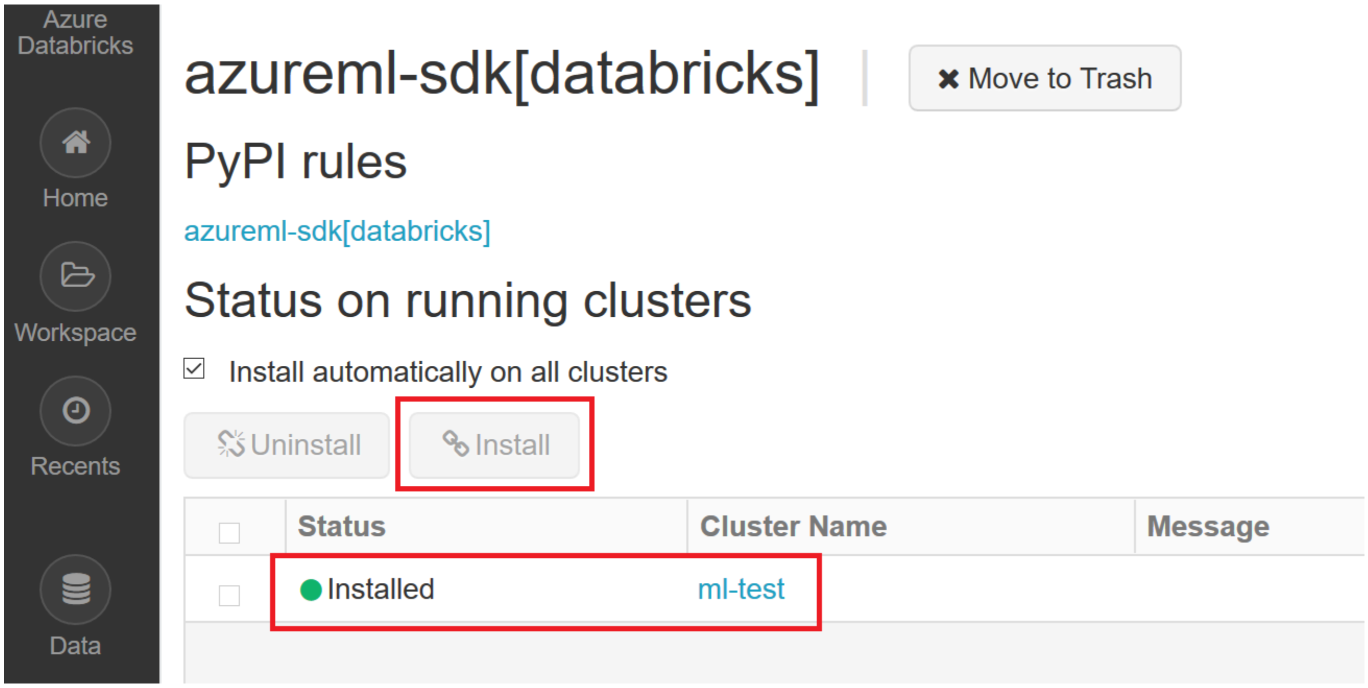
5a1. Create library

Select PyPi and then fill in: azureml-sdk[databricks]



5a2. Add PyPi library with azureml-sdk[databricks]

Finally, attach the library to the cluster.



	Status	Cluster Name	Message
	● Installed	ml-test	

5a3. Attach library to cluster

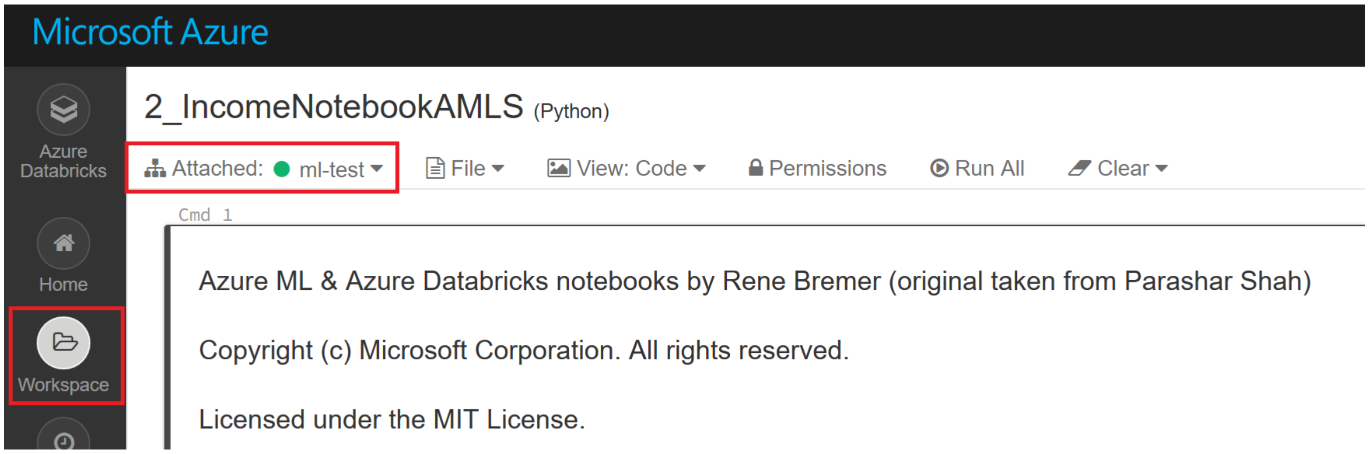
5b. Import notebook using Azure ML to Azure Databricks

In the prevous part of this tutorial, a model was created in Azure Databricks. In this part you are going to add the created model to Azure Machine Learning Service.

Go to your Databricks Service again, right click, select import and import the a notebook using the following URL:

```
https://raw.githubusercontent.com/rebremer/devopsai_databricks/maste
r/project/modelling/2_IncomeNotebookAMLS.py
```

Again, make sure it is attached to a cluster and the cluster is running



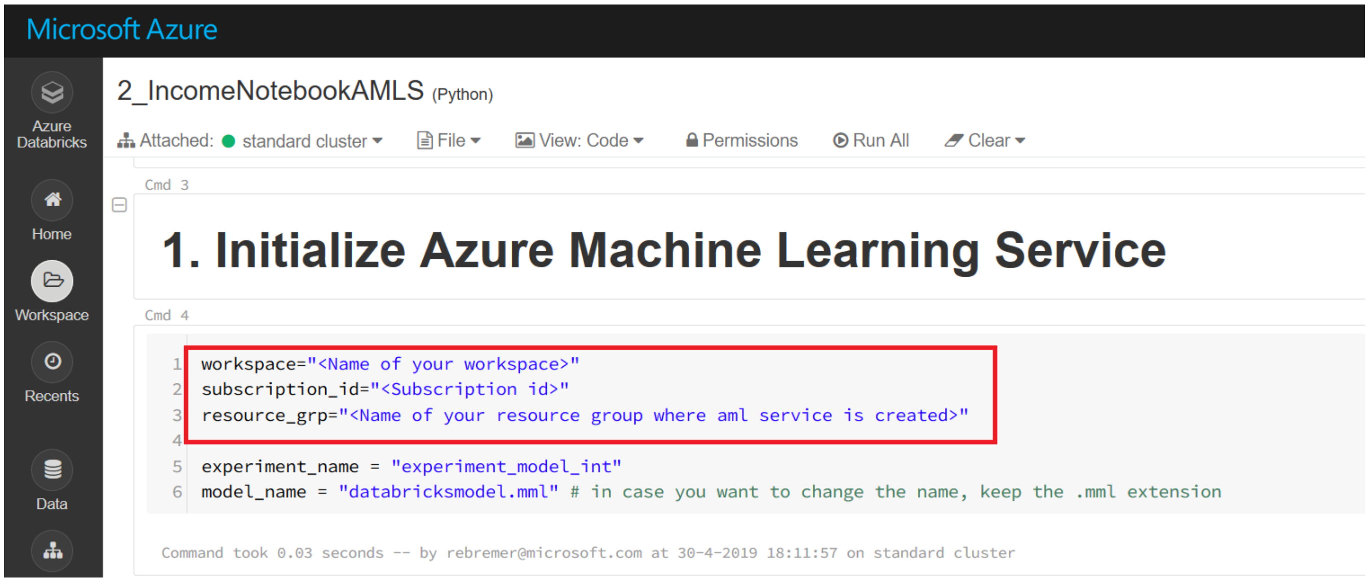
Azure ML & Azure Databricks notebooks by Rene Bremer (original taken from Parashar Shah)

Copyright (c) Microsoft Corporation. All rights reserved.

Licensed under the MIT License.

5b1. Import AMLS notebook

Subsequently, fill in the correct values for workspace, subscription_id and resource_grp. All values can be found in the overview tab of your Azure Machine Learning Service Workspace in the Azure Portal.



```
workspace="<Name of your workspace>"
subscription_id="<Subscription id>"
resource_grp="<Name of your resource group where aml service is created>"

experiment_name = "experiment_model_int"
model_name = "databricksmodel.mml" # in case you want to change the name, keep the .mml extension
```

```
Command took 0.03 seconds -- by rebremer@microsoft.com at 30-4-2019 18:11:57 on standard cluster
```

5b2. Add variables notebook

Now run the notebook cell for cell by using shortcut SHIFT+ENTER.

In cell 6, you will need to authenticate to Azure Machine Learning Service in the notebook. Follow the instruction in the notebook by opening the URL and enter the generated code to authenticate.

5c. Review results in Azure ML

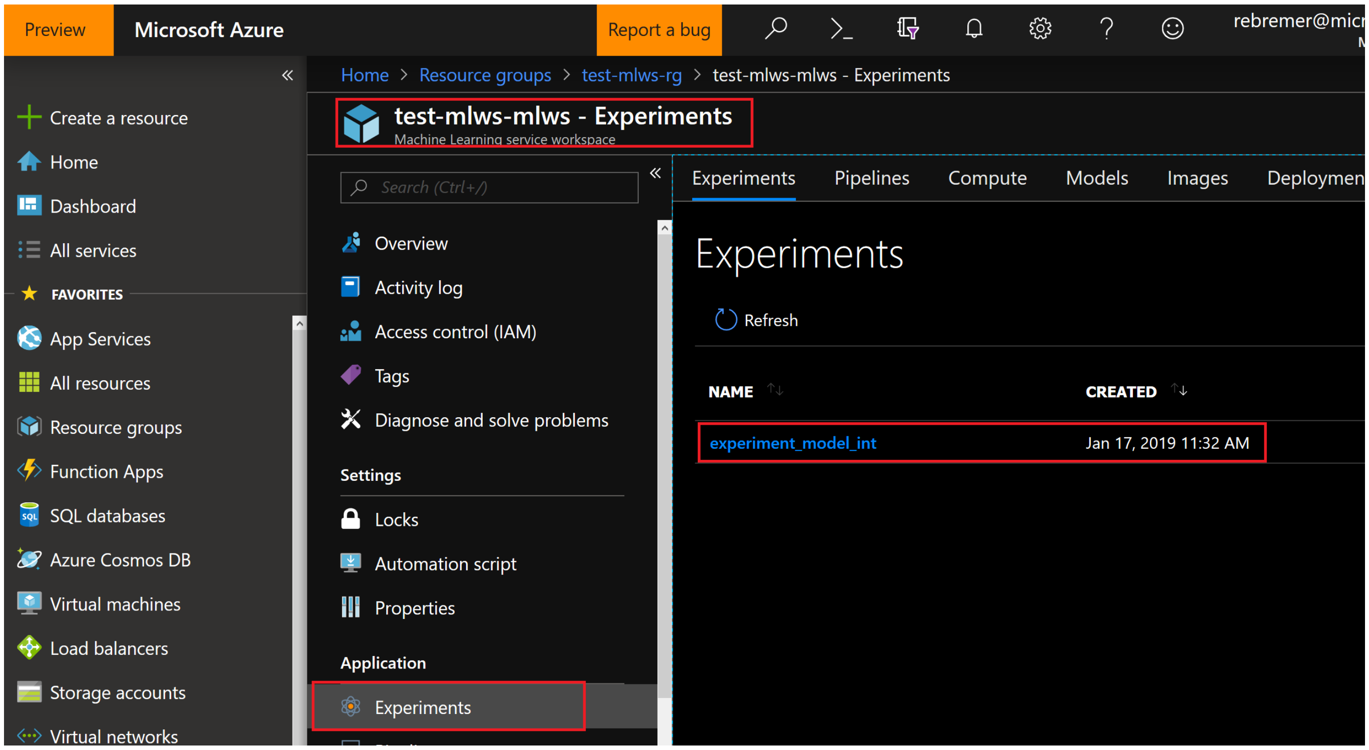
In step 5b, a notebook was run in which the results were written to Azure Machine Learning Service. In this, the following was done:

- A new experiment was created in you Azure ML
- With in this experiment, a root run with 6 child runs were the different attempts can be found.
- A childrun contains a description of the model (e.g. Logistic Regression with regularization 0) and the most important logging of the attempt (e.g. accuracy,

number of false postives)

- The model artificact (.mml) is also part of a childrun. The artifact of the best childrun can be taken and deployed into production.

Go to you Azure ML instance. Select the experiment name that was used in the notebook (e.g. experiment_model_int).

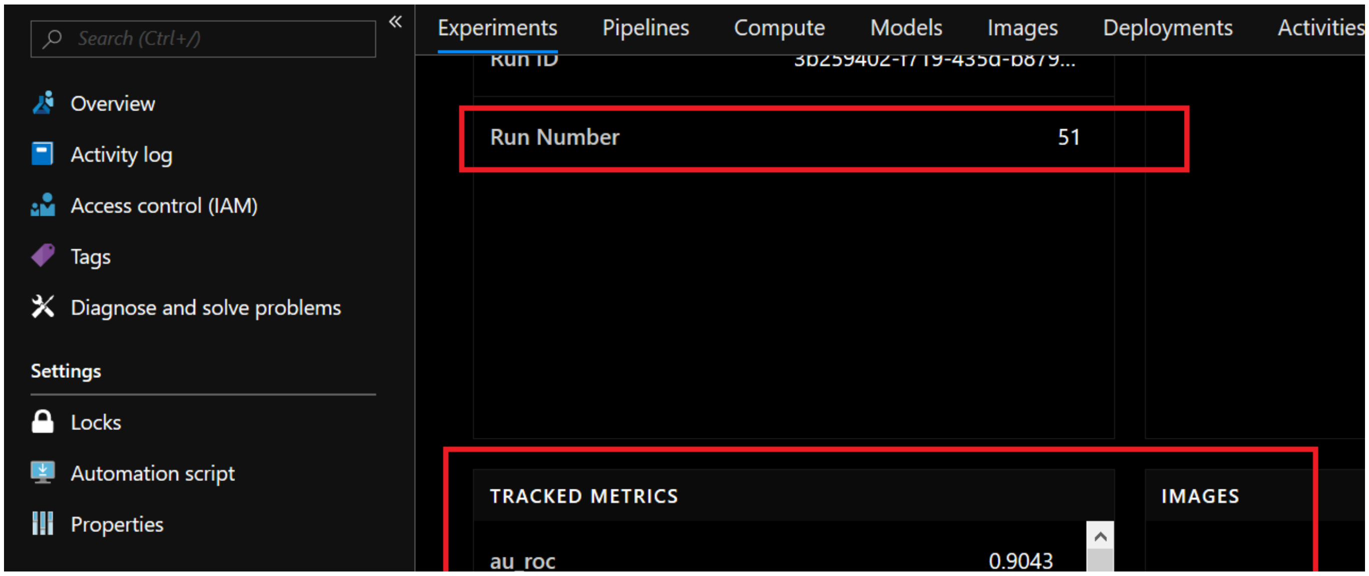


Experiments

NAME	CREATED
experiment_model_int	Jan 17, 2019 11:32 AM

5c1. Find experiment in Azure Machine Learning Service

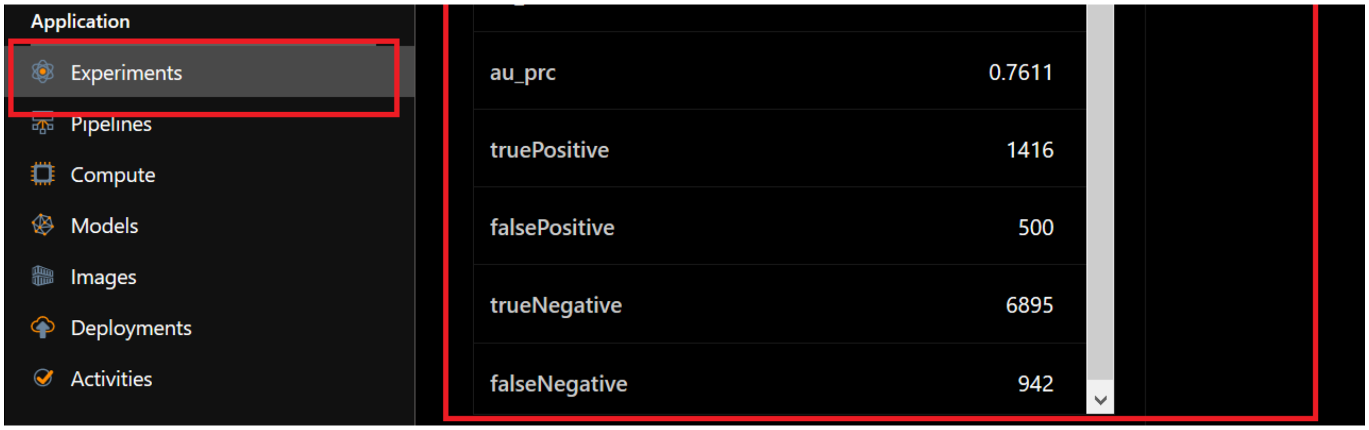
Now click on the experiment, click on the run and childrun you want to see the metrics.



Run ID	3b259402-f719-435d-b879...
Run Number	51

TRACKED METRICS

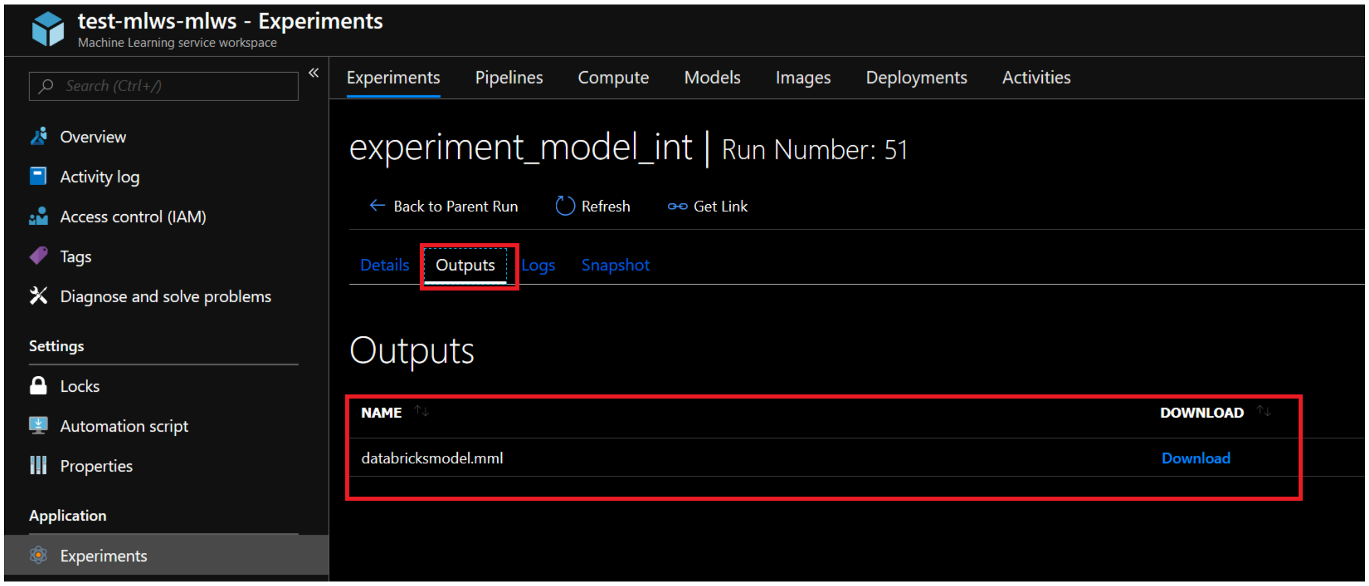
au_roc	0.9043



au_prc	0.7611
truePositive	1416
falsePositive	500
trueNegative	6895
falseNegative	942

5c2. Find metrics of a childrun in Azure Machine Learning Service

When you go to output, you will find the model artifact, which you can also download. The model artifact of the best run will be used as the base of the containter that is deployed using Azure DevOps in the next part of this tutorial.



NAME	DOWNLOAD
databricksmodel.mml	Download

5c3. Model artifact

6. Create and prepare Azure DevOps project

Azure DevOps is the tool to continuously build, test, and deploy your code to any platform and cloud. In chapter 6, an Azure DevOps will be created and prepared. The project will be prepared using the following steps:

- 6a. Create Personal Access Token in Databricks
- 6b. Create AKS cluster (optional)

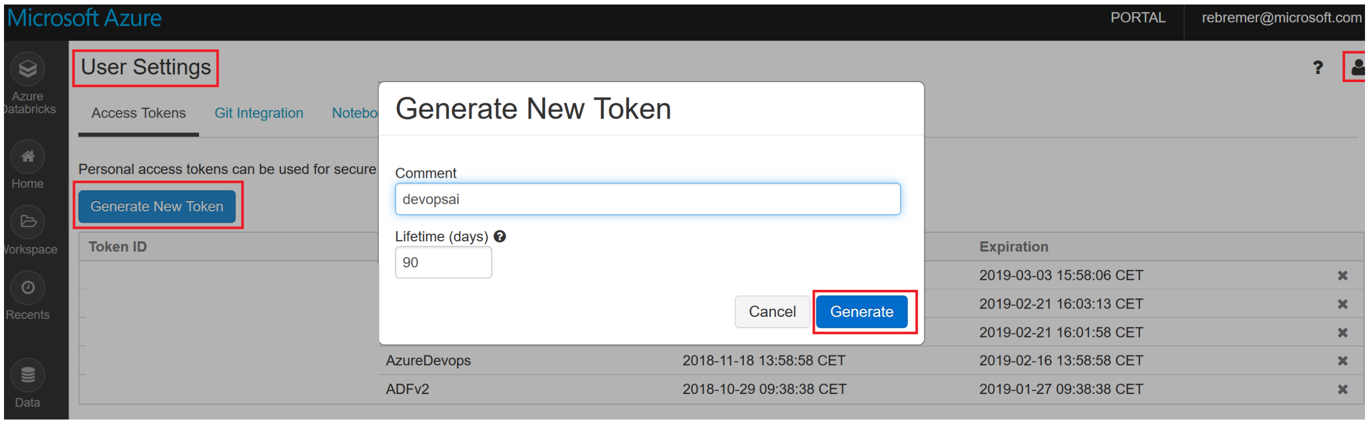
- 6c. Create Azure DevOps project and service connection
- 6d. Add variables to code

In chapter 7, the actual build-release pipeline will be created and run to create an endpoint of the model.

6a. Create Personal Access Token in Databricks

To run Notebooks in Azure Databricks triggered from Azure DevOps (using REST APIs), a Databrics Access Token (PAT) is required for authentication.

Go to Azure Databricks and click to the person icon in the upper right corner. Select User Settings and then generate a new token.



Personal access tokens can be used for secure

Generate New Token

Comment: devopsai

Lifetime (days): 90

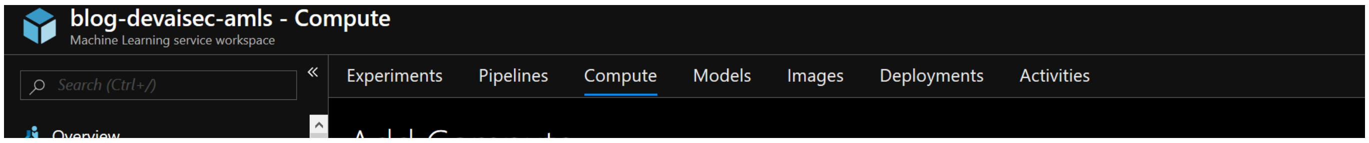
Token ID			Expiration
			2019-03-03 15:58:06 CET
			2019-02-21 16:03:13 CET
			2019-02-21 16:01:58 CET
	AzureDevops	2018-11-18 13:58:58 CET	2019-02-16 13:58:58 CET
	ADFv2	2018-10-29 09:38:38 CET	2019-01-27 09:38:38 CET

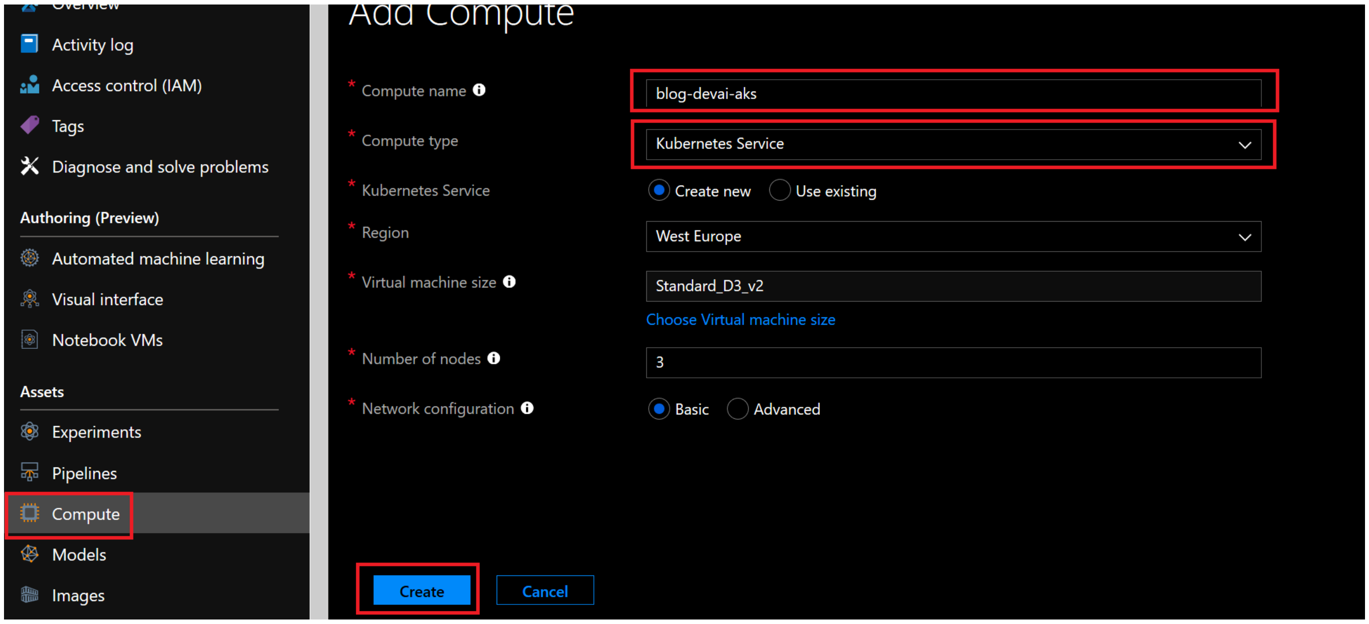
6a1. Generate Databricks Access Token

Make sure to copy the token now. You won't be able to see it again. Token is needed to access Databricks from the Azure DevOps build pipeline later

6b. Create AKS cluster (optional)

Typically, ACI is used for testing, whereas webaps or AKS cluster is used for a production situation. In this optional step, an prd environment is created in which the container is deployed to an AKS cluster. First, go to to you Azure ML Service Workspace and select Compute. Take as compute name **blog-devai-aks** and select Kubernetes Service as compute type, see also below.





Compute name: blog-devai-aks
Compute type: Kubernetes Service
Kubernetes Service: [✓ Create new] Use existing
Region: West Europe
Virtual machine size: Standard_D3_v2
Choose Virtual machine size
Number of nodes: 3
Network configuration: [✓ Basic] Advanced

6b1. Add AKS compute to Azure ML Service

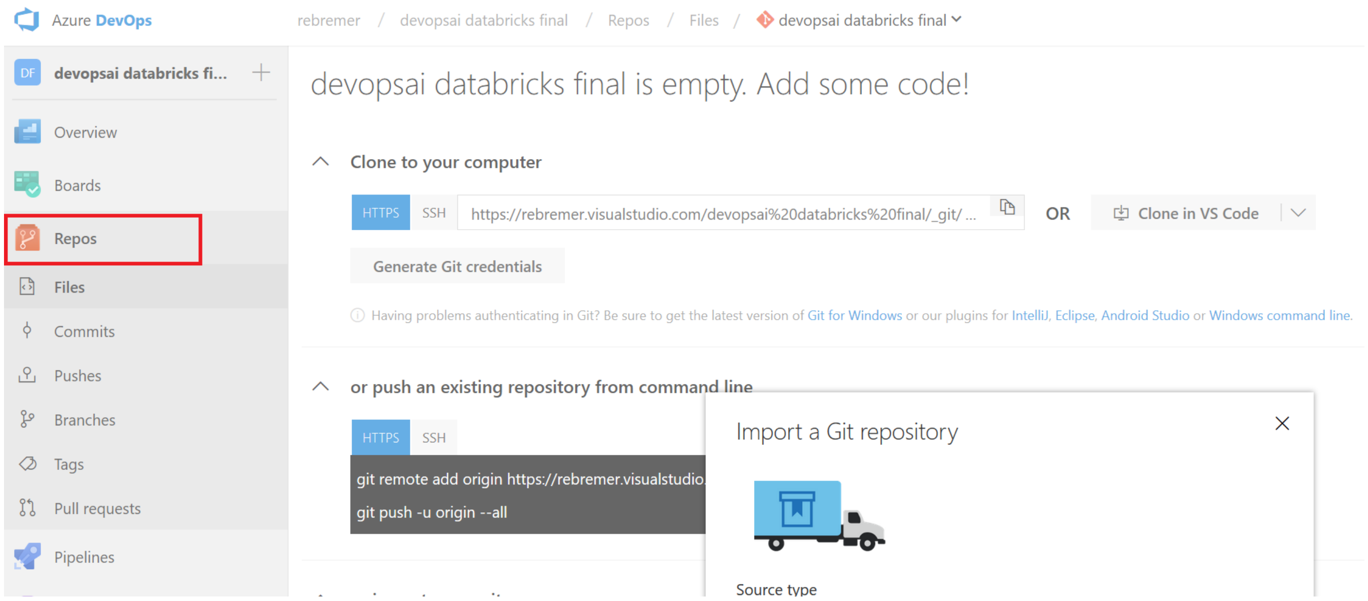
Creating an AKS cluster takes approximately 10 minutes. Continue to the next step.

6c. Create Azure DevOps project with service connection

Create a new project in Azure DevOps by following this tutorial. Once you create a new project, click on the repository folder and select to import the following repository:

```
https://github.com/rebremer/devopsai_databricks.git
```

See also picture below:



devopsai databricks final is empty. Add some code!

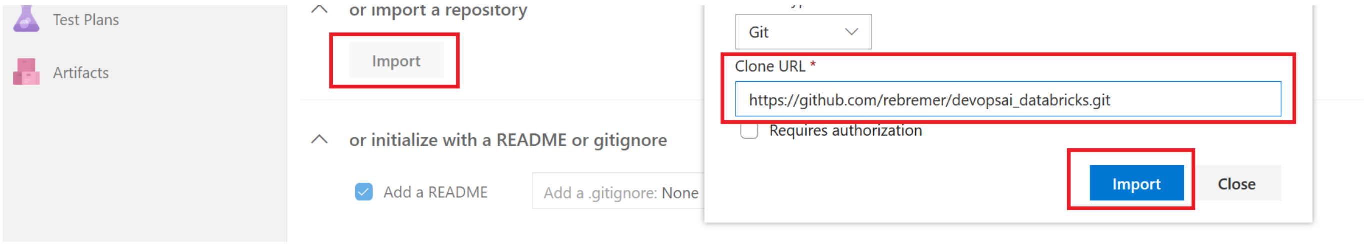
Clone to your computer

https://rebremer.visualstudio.com/devopsai%20databricks%20final/_git/ ...

Having problems authenticating in Git? Be sure to get the latest version of Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

or push an existing repository from command line

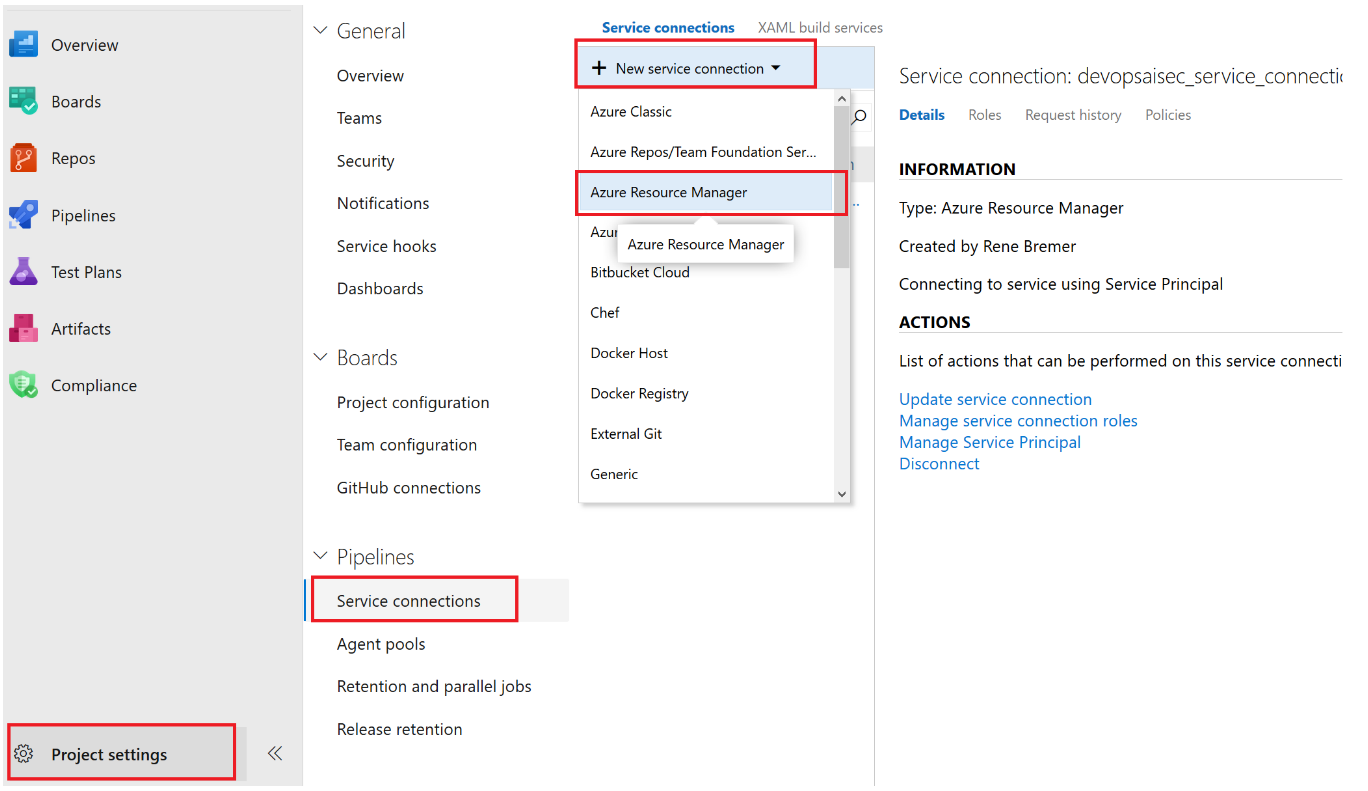
```
git remote add origin https://rebremer.visualstudio
git push -u origin --all
```



Clone URL *
https://github.com/rebremer/devopsai_databricks.git

6c1. Add repository to your Azure DevOps project

A Service connection is needed to access the resources in the resource group from Azure DevOps. Go to project settings, service connection and then select Azure Resource Manager.



Service connection: devopsaisec_service_connecti

INFORMATION

Type: Azure Resource Manager

Created by Rene Bremer

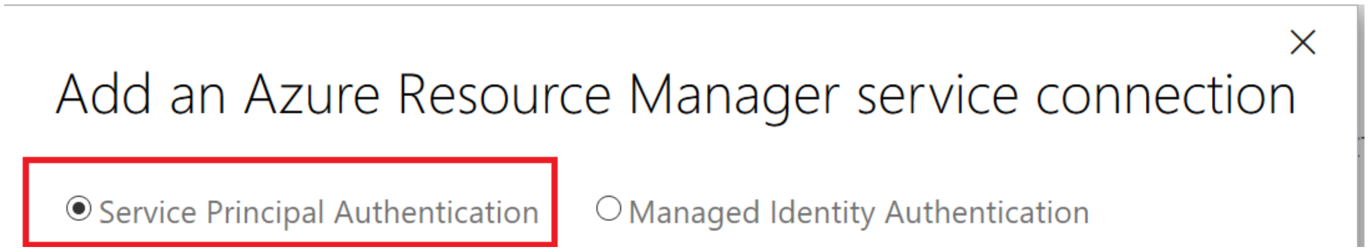
Connecting to service using Service Principal

ACTIONS

List of actions that can be performed on this service connecti

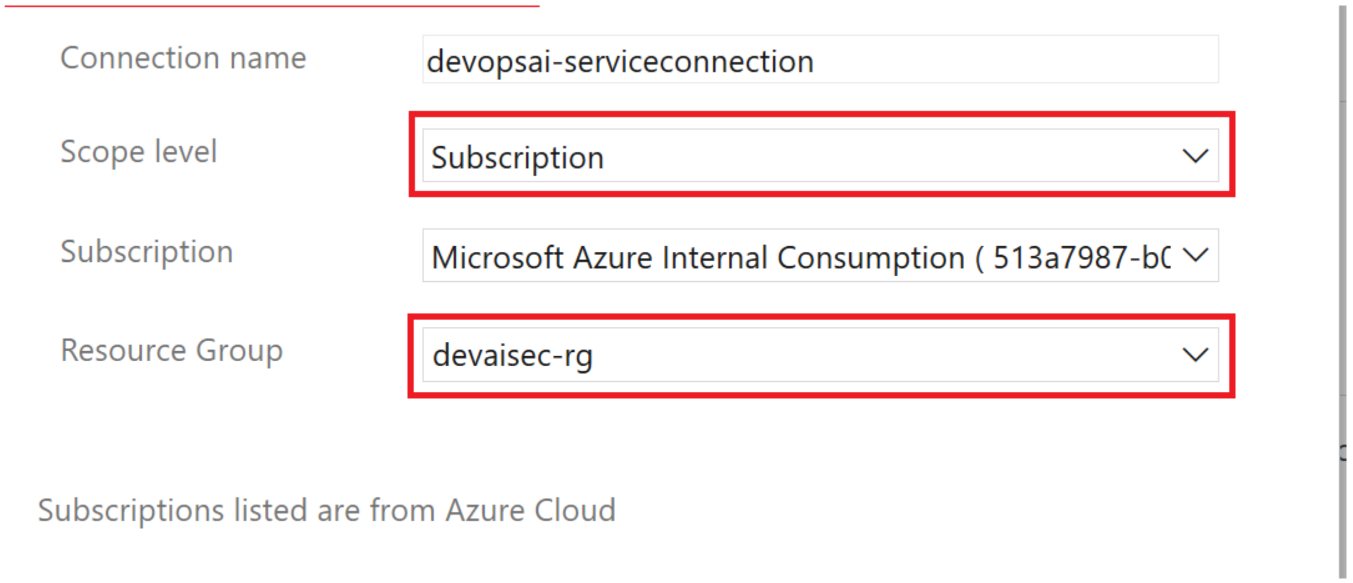
6c2. Go to Service Connection

Select Service Principal Authentication and limit scope to your resource group in which your Machine Learning Workspace Service is deployed. Make sure that you name the connection as follows: **devopsaisec_service_connection.**



Add an Azure Resource Manager service connection

[✓ Service Principal Authentication] Managed Identity Authentication


Connection name: devopsai-serviceconnection

Scope level: Subscription

Subscription: Microsoft Azure Internal Consumption (513a7987-b(

Resource Group: devaisec-rg

Subscriptions listed are from Azure Cloud

6c3. Create Azure Resource Manager service connection

6d. Add variables to code

In the Repos you created in the previous step, the following files shall be changed:

- \project\configcode_build_release.yml

With the same variables for workspace, subscription_id and resource with values of your Machine Learning Service Workspace as in step 5b. Also, fill in your Databricks Personal Access Token generated in step 6a.

```
variables:
  # change 5 variables below with your own settings, make sure that
  # : with a space is kept and not replaced with =
  workspace: '<<Name of your workspace>>'
  subscription_id: '<<Subscription id>>'
  resource_grp: '<<Name of your resource group with aml service>>'
  domain: 'westeurope.azuredatabricks.net' # change loc.when needed
  dbr_pat_token_raw: '<<your Databricks Personal Access Token>>'
```

The files can be changed in Azure DevOps by looking up the file in the Repos, click on “edit”, change the variables and then “commit” the file. You can also clone the project and work from there. Notice that in a production situation, keys must never be added to a code. Instead, secret variables in an Azure DevOps pipeline shall be used and is dealt with in this follow-up tutorial.

In this chapter, an Azure DevOps project is created and prepared. Now the model is ready to be built and released in the Azure DevOps project.

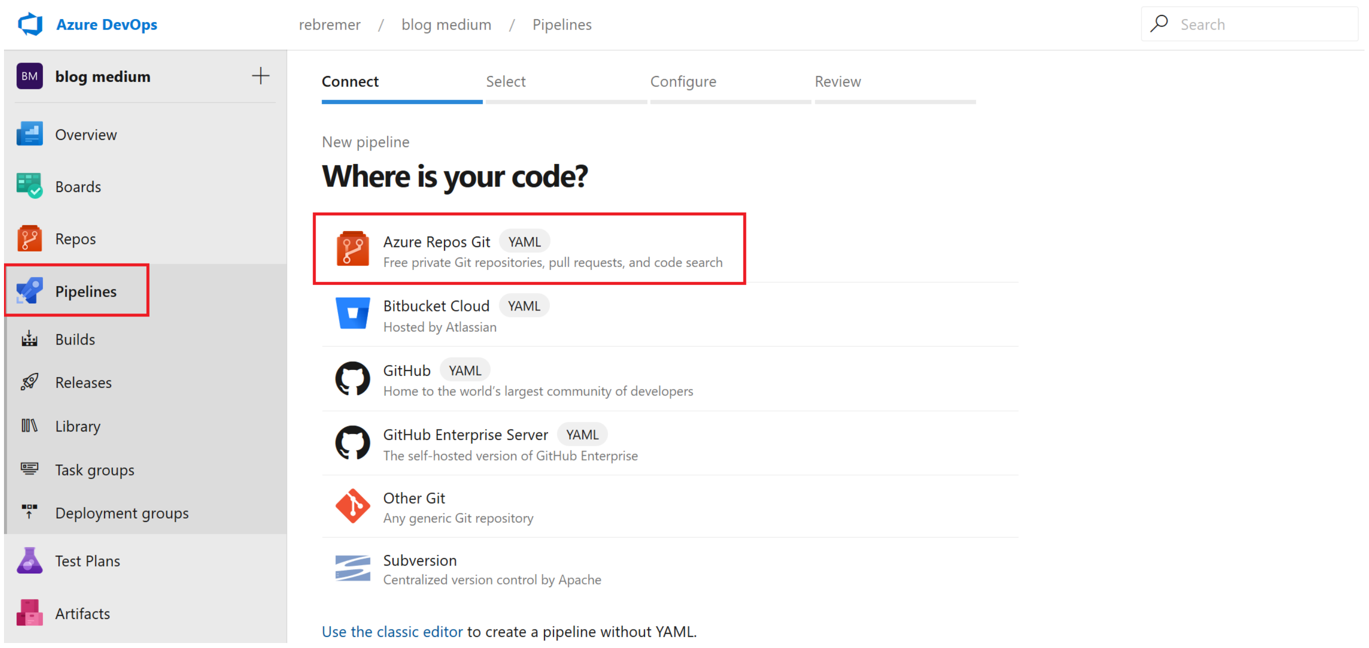
7. Build and release model in Azure DevOps

In this part, the model is built and released in the Azure DevOps using the following steps:

- 7a. Create build-release pipeline
- 7b. Run build-release pipeline
- 7c. Consume HTTP endpoint with Postman

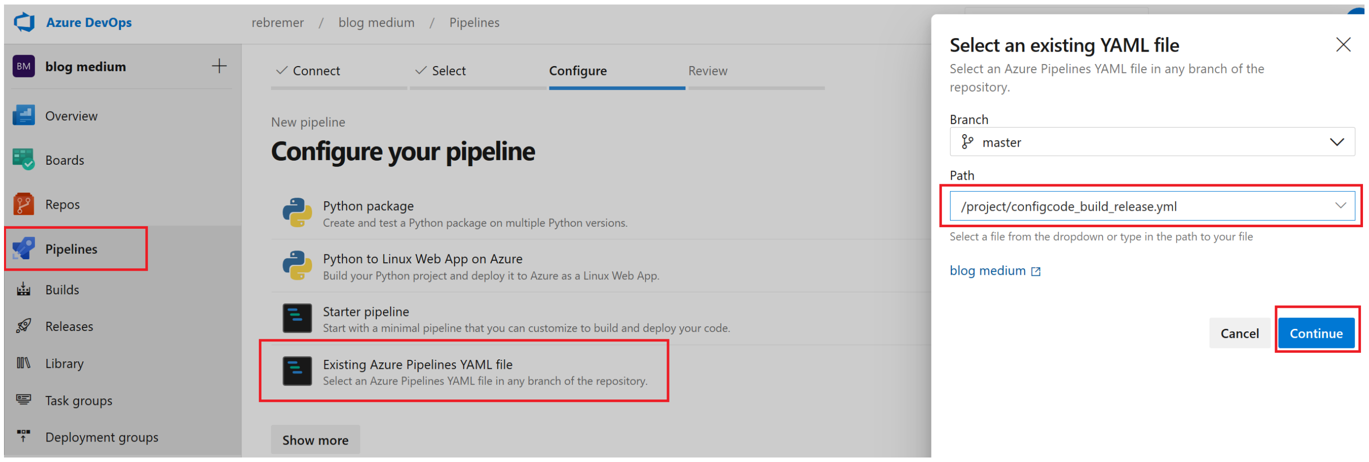
7a. Create build-release pipeline

In this step, you are going to create a build-release pipeline. Go to Azure DevOps project you have created in 6c and then click on Pipelines. A wizard is shown in which your Azure Repos Git shall be selected, see also below.


Azure DevOps
rebremer / blog medium / Pipelines
Search
blog medium
Overview
Boards
Repos
Pipelines
Builds
Releases
Library
Task groups
Deployment groups
Test Plans
Artifacts
Connect
Select
Configure
Review
New pipeline
Where is your code?
Azure Repos Git YAML
Free private Git repositories, pull requests, and code search
Bitbucket Cloud YAML
Hosted by Atlassian
GitHub YAML
Home to the world's largest community of developers
GitHub Enterprise Server YAML
The self-hosted version of GitHub Enterprise
Other Git
Any generic Git repository
Subversion
Centralized version control by Apache
Use the classic editor to create a pipeline without YAML.

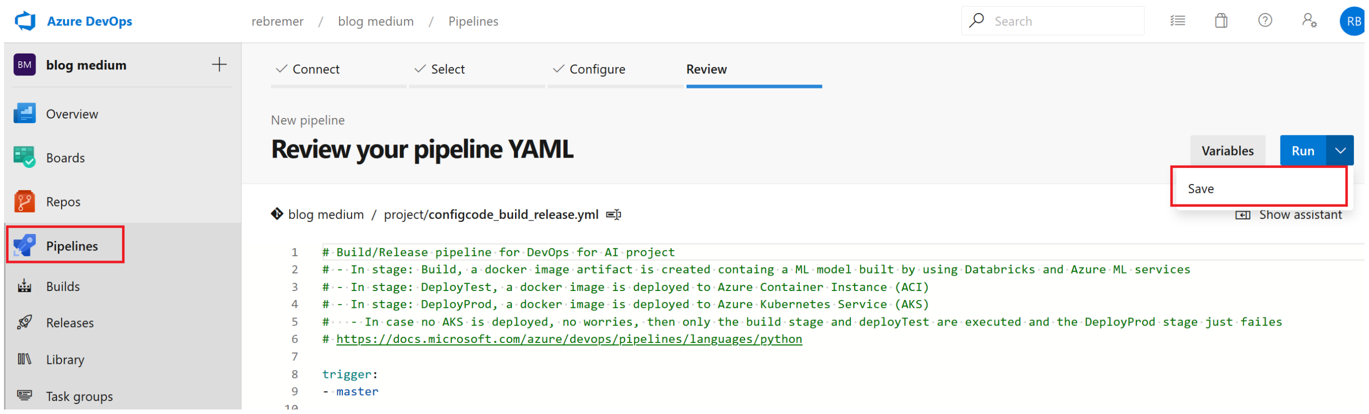
7a1. Create Pipeline

Subsequently, select your Git repo attached to this project and then select "Existing Azure Pipelines YAML file". Then browse the directory "project/configcode_build_release.yml", see also below.


Azure DevOps
rebremer / blog medium / Pipelines
blog medium
Overview
Boards
Repos
Pipelines
Builds
Releases
Library
Task groups
Deployment groups
Connect
Select
Configure
Review
New pipeline
Configure your pipeline
Python package
Create and test a Python package on multiple Python versions.
Python to Linux Web App on Azure
Build your Python project and deploy it to Azure as a Linux Web App.
Starter pipeline
Start with a minimal pipeline that you can customize to build and deploy your code.
Existing Azure Pipelines YAML file
Select an Azure Pipelines YAML file in any branch of the repository.
Show more
Select an existing YAML file
Select an Azure Pipelines YAML file in any branch of the repository.
Branch
master
Path
/project/configcode_build_release.yml
Select a file from the dropdown or type in the path to your file
blog medium
Cancel
Continue

7a2. Select build-release YAML file

Finally review your pipeline and save your pipeline, see also below.



```
# Build/Release pipeline for DevOps for AI project
# - In stage: Build, a docker image artifact is created containg a ML model built by using Databricks and Azure ML services
# - In stage: DeployTest, a docker image is deployed to Azure Container Instance (ACI)
# - In stage: DeployProd, a docker image is deployed to Azure Kubernetes Service (AKS)
#   - In case no AKS is deployed, no worries, then only the build stage and deployTest are executed and the DeployProd stage just failes
# https://docs.microsoft.com/azure/devops/pipelines/languages/python

trigger:
- master
```

7a3. Save pipeline

In this chapter, the pipeline was configured. In this pipeline the following steps will be executed:

Build:

- Select Python 3.6 and install dependencies
- Upload notebook to Databricks
- Create model using Azure Databricks by running notebook. Add model to Azure Machine Learning service
- Create a docker image from model that will be deployed in release step
- Creation of build artifact as input for release deployTest and deployProd

Release deployTest:

- Retrieve docker image created in Build step
- Deploy docker image to Azure Container Instance (ACI)
- Test endpoint in ACI

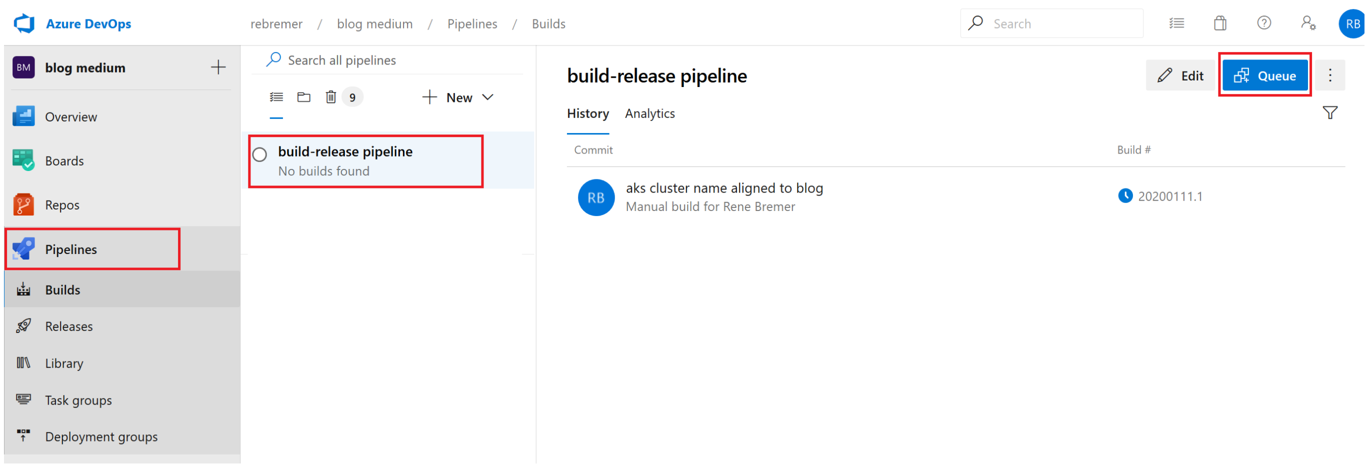
Release deployProd (optional, when AKS cluster is deployed in step 7a):

- Retrieve docker image created in Build step
- Deploy docker image to Azure Kubernetes Service (AKS)

In the next part, the pipeline will be run.

7b. Run build-release pipeline

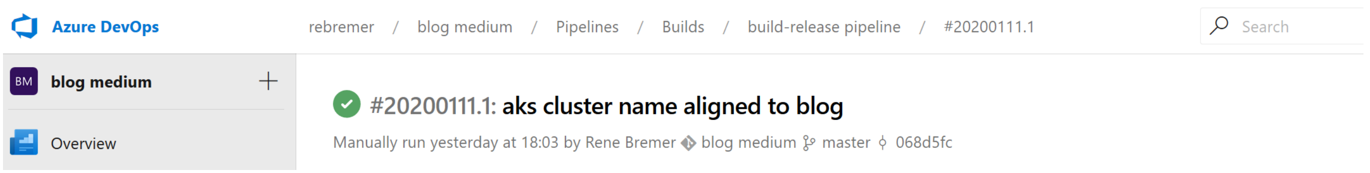
In this step, the build-release pipeline will be run in Azure DevOps. Go to your pipeline deployed in the previous step, select the pipeline and then select queue, see also below.



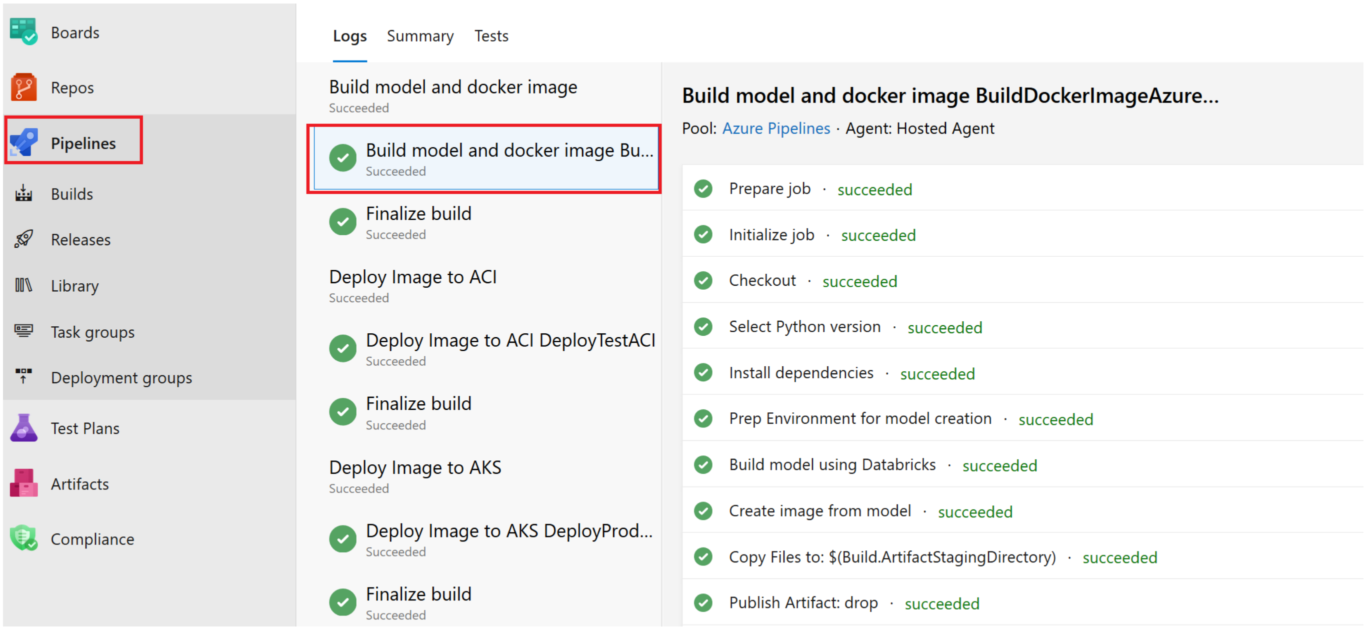
aks cluster name aligned to blog
Manual build for Rene Bremer
20200111.1

7b1. Run build-release pipeline

When the pipeline is started, a docker image is created containing an ML model using Azure Databricks and Azure ML in the build step. Subsequently, the docker image is deployed/released in ACI and AKS. A successful run can be seen below.


Azure DevOps
rebremer / blog medium / Pipelines / Builds / build-release pipeline / #20200111.1
Search
blog medium
Overview

#20200111.1: aks cluster name aligned to blog
Manually run yesterday at 18:03 by Rene Bremer blog medium master 068d5fc



Build model and docker image
Succeeded
Build model and docker image Bu...
Succeeded
Finalize build
Succeeded
Deploy Image to ACI
Succeeded
Deploy Image to ACI DeployTestACI
Succeeded
Finalize build
Succeeded
Deploy Image to AKS
Succeeded
Deploy Image to AKS DeployProd...
Succeeded
Finalize build
Succeeded

Build model and docker image BuildDockerImageAzure...
Pool: Azure Pipelines · Agent: Hosted Agent

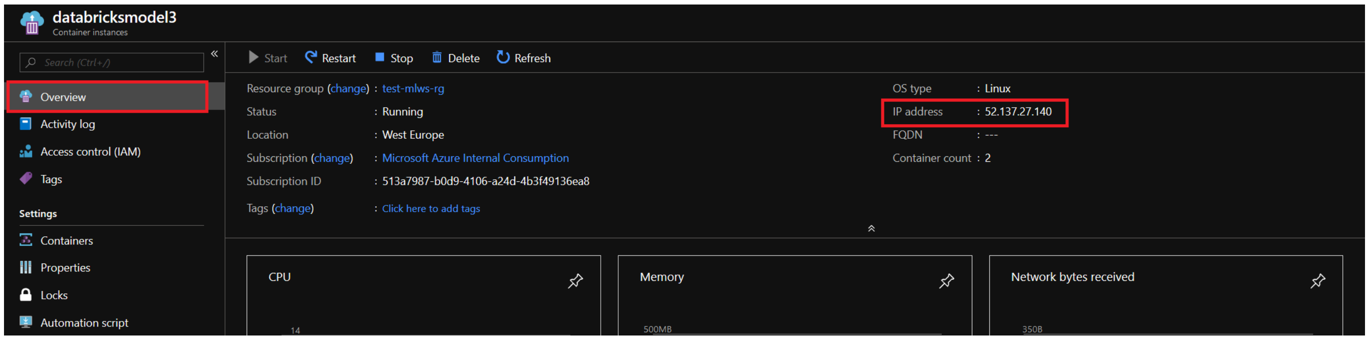
Prepare job · succeeded
Initialize job · succeeded
Checkout · succeeded
Select Python version · succeeded
Install dependencies · succeeded
Prep Environment for model creation · succeeded
Build model using Databricks · succeeded
Create image from model · succeeded
Copy Files to: $(Build.ArtifactStagingDirectory) · succeeded
Publish Artifact: drop · succeeded

7b2. Successful run of build-release pipeline

Notice that if you decided to not deploy the docker image in AKS, the previous steps will still be executed and the AKS step will fail. For detailed logging, you can click on the various steps.

7c. Consume HTTP endpoint with Postman

When you go to the Azure Portal, you can find the Azure Container Instance deployed in 7b and subsequently, you can find the public IP address of the HTTP endpoint.



Resource group (change) : test-mlws-rg
Status : Running
Location : West Europe
Subscription (change) : Microsoft Azure Internal Consumption
Subscription ID : 513a7987-b0d9-4106-a24d-4b3f49136ea8
Tags (change) : Click here to add tags

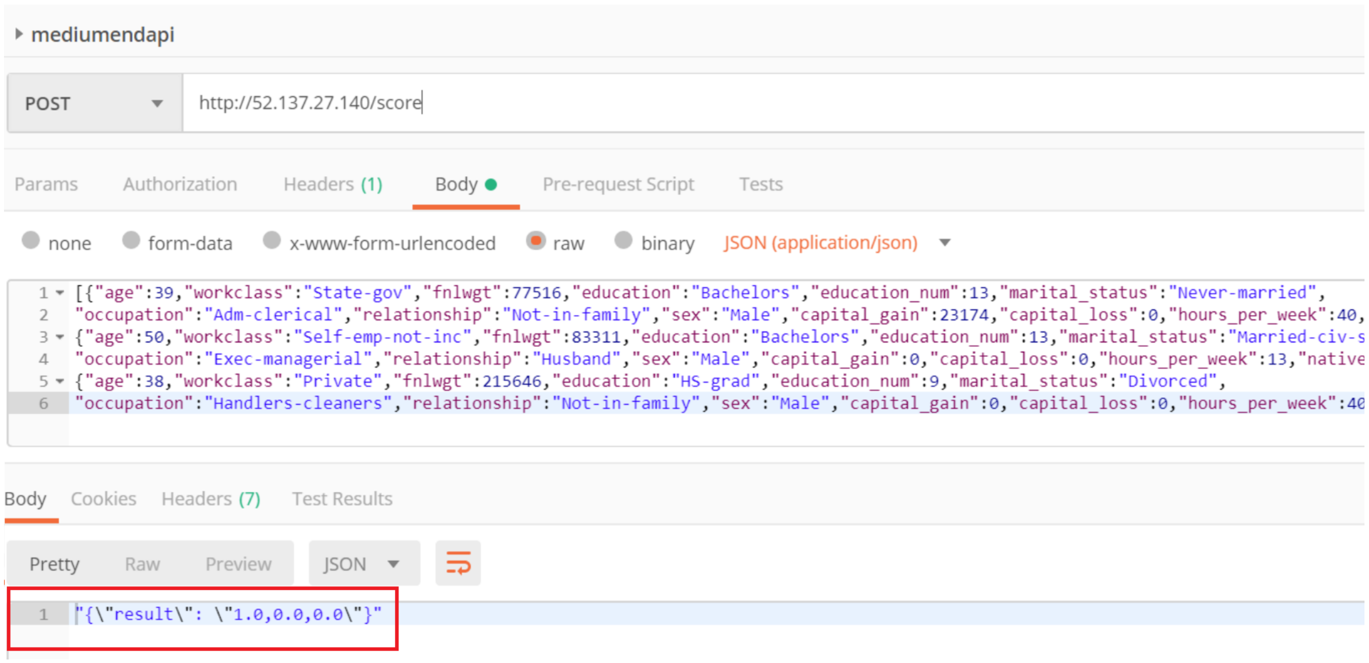
OS type : Linux
IP address : 52.137.27.140
FQDN : ---
Container count : 2

7c1. IP address of HTTP endpoint

This IP address will now be consumed by Postman to create predictions. An example payload can be found in the project/services/50_testEndpoint.py in the project. In this example, the income class of three persons is predicted.

- The prediction for the first person is that the income is higer than 50k,

- For the other two persons the prediction is lower than 50k.



```
http://52.137.27.140/score
```

```
[{"age":39,"workclass":"State-gov","fnlwgt":77516,"education":"Bachelors","education_num":13,"marital_status":"Never-married",
"occupation":"Adm-clerical","relationship":"Not-in-family","sex":"Male","capital_gain":23174,"capital_loss":0,"hours_per_week":40,
{"age":50,"workclass":"Self-emp-not-inc","fnlwgt":83311,"education":"Bachelors","education_num":13,"marital_status":"Married-civ-s
"occupation":"Exec-managerial","relationship":"Husband","sex":"Male","capital_gain":0,"capital_loss":0,"hours_per_week":13,"native
{"age":38,"workclass":"Private","fnlwgt":215646,"education":"HS-grad","education_num":9,"marital_status":"Divorced",
"occupation":"Handlers-cleaners","relationship":"Not-in-family","sex":"Male","capital_gain":0,"capital_loss":0,"hours_per_week":40
```

```
"{\"result\": \"1.0,0.0,0.0\"}"
```

7c2. Use HTTP endpoint to create predictions

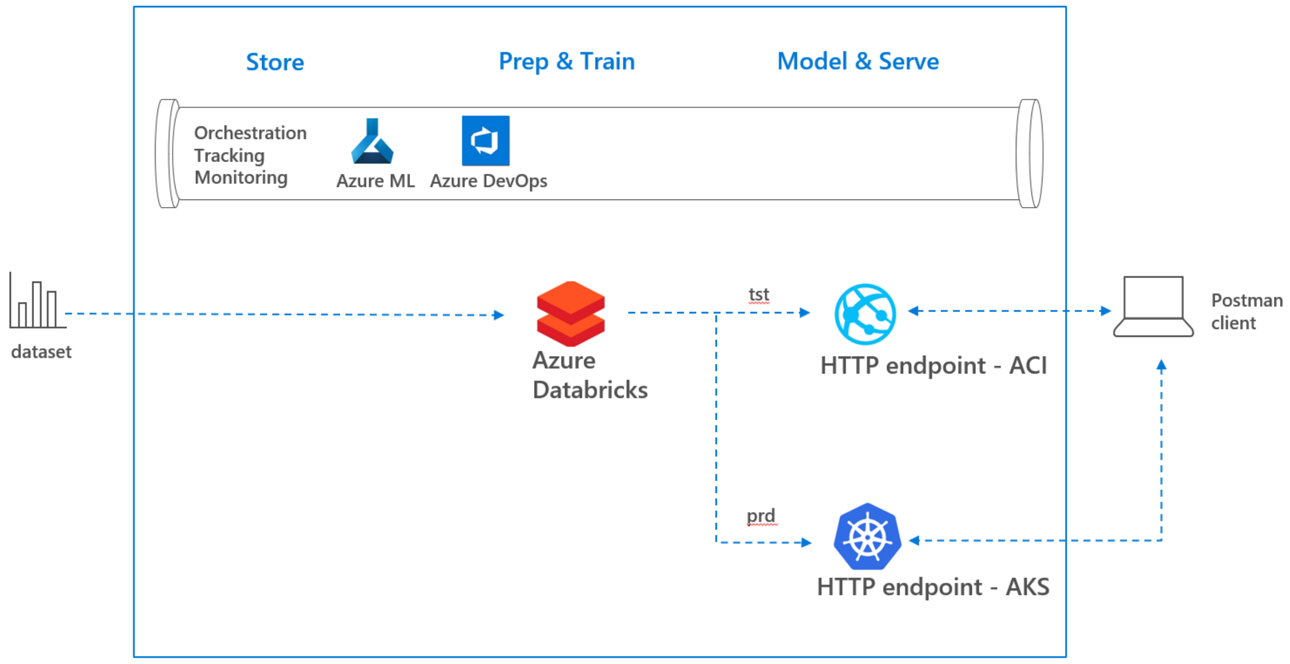
8. Conclusion

In this tutorial, an end to end pipeline for a machine learning project was created. In this:

- Azure Databricks with Spark was used to explore the data and create the machine learning models.
- Azure Machine Learning Service was used to keep track of the models and its metrics.
- Azure Devops was used to build an image of the best model and to release it as an endpoint.

This way you can orchestrate and monitor the entire pipeline from idea to the moment that the model is brought into production. This enables you to answer to question: *Why did the model predict this?*

The architecture overview can be found below. In this follow-up tutorial, security of the pipeline is enhanced.



8. High level overview

Azure Azure Databricks Continuous Integration Data Science Machine Learning