Azure Machine Learning

Flight Delay ML Demo: Setup Guide

with Azure Machine Learning

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

© 2021 Microsoft. All rights reserved.

# Infrastructure Deployment

To run our demo, we first need to deploy all the required infrastructure pieces.

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
|  | 1. From the Azure Portal, click on the **Create a resource** button. 2. Search for **Template deployment** and click **Create**. 3. Click on **Build your own template in the editor**. |
|  | 1. Click on **Load file**. 2. Locate the ARM Template **aml.json** in the **armTemplates** folder. 3. Click **Save**. |
|  | 1. Select your **Subscription** and **Resource Group** and **Region.** 2. Click **Review + create**. 3. Click **Create**. |

# Azure ML

## Compute Instance & Setup Notebook

Each AML compute VM is restricted to a single user. If this is your first time presenting, make sure you follow the setup scripts.

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
| A screenshot of a cell phone  Description automatically generated | 1. From inside the portal, click **Compute**. |
|  | 1. Within the **Compute Instances** tab. 2. Click on the **+ New** button. |
|  | 1. Select a name and size for your new notebook. The notebook name should be unique within the region. A **STANDARD\_D3\_V2** size is sufficient for this demo. 2. Click **Next** button. 3. Select a name (e.g. **aml-notebook-<username>**). 4. Click **Create.** |
| NotebookCreating | 1. The provisioning of the machine usually takes a couple of minutes. The **Status** will remain as **Creating** during the provisioning process. |
|  | 1. Select the **Notebooks** tab. 2. Click on the **+** icon and then on **Upload folder.** 3. Upload the root folder for this package. |
|  | 1. Update the **workspace** details (these details are available from the Azure Portal):    1. **Subscription ID**    2. **Resource Group**    3. **Workspace Name** |
|  | 1. Go to the **Notebooks** tab. 2. Navigate to **/notebooks/setup/setup.ipynb** 3. Execute the notebook until the end. |

## Git [optional]

Instead of uploading the notebook content, you may prefer to clone the content of the demo from a Git repo. This enables features like Git commit hash tracking when you are running experiments.

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
| Graphical user interface, application  Description automatically generated | 1. Go to **Azure Portal** > Resource Group > Workspace. 2. Go to the **Notebooks** side item and create a directory for installing **git-lfs.** |
| A screen shot of a computer screen  Description automatically generated with low confidence  Graphical user interface, text, application  Description automatically generated | 1. Go to [Git LFS Docs](https://docs.github.com/en/github/managing-large-files/versioning-large-files/installing-git-large-file-storage) for installations instructions. 2. On the [Github release](https://github.com/git-lfs/git-lfs/releases/tag/v2.13.3) page, download **Linux AMD64** asset and unzip it. 3. Unzip the file and upload the unzipped directory into the previously created directory. 4. Click on the **Terminal** button in the AML workspace. 5. Within the terminal, navigate to the uploaded directory and run **sudo ./install.sh**    1. Note: You should now see Git LFS initialized.    2. Note: The LFS directory may now be removed. |
| Graphical user interface, application, Teams  Description automatically generated | 1. Go to the **Notebooks** side item and create a directory for cloning/importing repo. 2. Click on the **Terminal** button. |
| Graphical user interface, text, application  Description automatically generated | 1. **Checkout** the repository. Make sure to use HTTPS to clone. |

## Visual Studio Code

This is only required if you want to demo the Classical ML with VS Code Deployment notebook.

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
|  | 1. Open **Visual Studio Code** on your local machine. |
|  | 1. Head over to the extension tab, and in the search input type in **Azure Machine Learning**. 2. Click on **Install**. |

## ONNX C#

This is only required if you want to demo the ONNX runtime as part of the MLOps demo.

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
|  | 1. Open the extracted source code within a terminal and navigate to the **OnnxRuntime** directory. |
|  | 1. From within the **OnnxRuntime** directory, run the **dotnet build** command. 2. Run the dotnet run command. |

## Deep Learning

This is only required if you want to demo the Deep Learning & Labeling notebook.

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
| Graphical user interface, text, application, email  Description automatically generated | 1. Go to **Azure Portal** > Resource Group > Container Registry. |
| Graphical user interface, text, application  Description automatically generated | 1. Select **Access Keys** from the left side panel and take note of the following:    1. Login server    2. Username    3. Password |
| Graphical user interface, text, application, email  Description automatically generated | 1. Navigate to **Azure Portal** > Resource Group > Workspace > Launch studio. 2. Go to the **Notebooks** nav item and open the environment\_definition.py file. 3. **Update** <address>, <username> and <password> with values recorded earlier. |

## Data Labeling

This is only required if you want to demo the Deep Learning & Labeling notebook.

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
|  | 1. Sign into **Azure Machine Learning**. 2. Select the subscription and the. 3. Select **Data labeling** on the left-hand nav. 4. Click the **Create** option to create a new project. |
| Graphical user interface, application  Description automatically generated | 1. Enter a name for the project. 2. Select **Image** for the **Media type**. 3. Select **Object identification (Bounding box)** for the **Labeling Task Type** option. 4. Click the **Next** button. |
| Graphical user interface, application, Teams  Description automatically generated | 1. Click **Create dataset**. |
| Graphical user interface, application  Description automatically generated | 1. Input a **name** for the dataset (e.g. **birds\_ds**). 2. Click the **Next** button. 3. Click the **Upload** button and select image files from the **notebooks/flight-delay-dl/images/train** directory. 4. Click the **Next** button, followed by the **Create** button. 5. **Select** your dataset from the list. 6. Click the **Next** button. |
| Graphical user interface, text, application  Description automatically generated | 1. Click the **Next** button. |
| Graphical user interface, text, application  Description automatically generated | 1. Enter **Bird** as the label. 2. Click the **Next** button. |
| Graphical user interface, text, application  Description automatically generated | 1. **Optionally,** enter a URL and instructions. 2. Click the **Next** button. |
| Graphical user interface, text, application, email  Description automatically generated | 1. Click the **Create project** button. |
| Graphical user interface, text, application, email  Description automatically generated | 1. Wait for the project to be created. 2. Click the **project name** in the list. |
| Graphical user interface, application  Description automatically generated | 1. Click the **Label data** option. |
| A picture containing text, screenshot  Description automatically generated | 1. Click the **Tasks** tab. |
| A picture containing text, screenshot, display  Description automatically generated | 1. **Draw a box** in each of the birds and set the label. 2. After labeling all the objects, click the **Submit** button at the bottom.  N.B. Labeling a single image is sufficient. |

## Designer

This is only required if you want to demo the Designer module.

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
|  | 1. From inside your portal, click **Designer**. 2. Click the **Easy-to-use prebuilt modules** option. |
|  | 1. In the right panel click the **Select compute target** option. 2. Select the created instance. 3. Click the **Save** button. |
|  | 1. Drag the **flightdelayweather\_ds** dataset from the components panel to the canvas. |
|  | 1. In the components panel, enter **Clean Missing Data** into the search input field. 2. Drag to the canvas the **Clean Missing Data** component. 3. Click the component in the canvas to show the configuration in the right panel. 4. Click the **Edit column** for the **Columns to be cleaned** option. 5. Leave the first option as **Column names**. 6. Enter the value **ArrDelay15**. 7. Click the **Save** button. 8. Connect the two components. |
|  | 1. In the components panel enter **Select Columns in Dataset** into the search input field. 2. Drag to the canvas the component and click it. 3. In the right panel click the **Edit column** for the **Select columns** option. 4. Select the **All columns**. 5. Click the **Save** button. 6. Connect the component with the Clean Missing Data one. |
|  | 1. In the components panel enter **Apply SQL Transformation** into the search input field. 2. Drag to the canvas the component and click it. 3. In the right panel enter the following query to the SQL query script space:   select Month,DayofMonth,DayOfWeek,CRSDepTime,CRSArrTime,UniqueCarrier,CRSElapsedTime,Origin,Dest,Distance,ArrDelay15,Origin\_Lat,Origin\_Lon,Origin\_State,Dest\_Lat,Dest\_Lon,Dest\_State,CAST(Origin\_dayl AS float) AS Origin\_dayl,CAST(Dest\_dayl AS float) AS Dest\_dayl,CAST(Origin\_prcp AS float) AS Origin\_prcp,CAST(Dest\_prcp AS float) AS Dest\_prcp,CAST(Origin\_srad AS float) AS Origin\_srad,CAST(Dest\_srad AS float) AS Dest\_srad,CAST(Origin\_swe AS float) AS Origin\_swe,CAST(Dest\_swe AS float) AS Dest\_swe,CAST(Origin\_tmax AS float) AS Origin\_tmax,CAST(Dest\_tmax AS float) AS Dest\_tmax,CAST(Origin\_tmin AS float) AS Origin\_tmin,CAST(Dest\_tmin AS float) AS Dest\_tmin,CAST(Origin\_vp AS float) AS Origin\_vp,CAST(Dest\_vp AS float) AS Dest\_vp  from t1  where Dest\_dayl != '--' and Dest\_prcp != '--' and Dest\_srad != '--' and Dest\_swe != '--' and Dest\_tmax != '--' and Dest\_tmin != '--' and Dest\_vp != '--'   1. Connect the component with the previous one. |
|  | 1. In the components panel enter **Summarize Data** into the search input field. 2. Drag to the canvas. 3. Connect with the previous component. |
|  | 1. In the components panel enter **Filter Based Feature Selection** to the search input field. 2. Drag to the canvas and click it. 3. In the right panel for the **Target column** option click the **Edit column**. 4. Leave **Column names** option selected and select the **ArrDelay15** column. 5. Click the **Save** button. 6. Connect the component with the **Apply SQL Transformation** component. |
|  | 1. In the components panel enter **Two-Class Logistic Regression in**to the search input field. 2. Drag to the canvas. 3. Do not connect this component with previous components. |
|  | 1. In the components panel enter **Split Data** into the search input field. 2. Drag to the canvas. 3. Connect the **Filter Based Feature Selection** component with this one. |
|  | 1. In the components panel enter **Two-Class Averaged Perceptron** into the search input field. 2. Drag to the canvas. 3. Do not connect this component with previous components. |
|  | 1. In the components panel enter **Train Model** into the search input field. 2. Drag to the canvas and click it. 3. In the right panel for the **Label column** option click the **Edit column**. 4. Leave **Column names** option selected and select the **ArrDelay15** column. 5. Click the **Save** button. 6. Connect the component with the previous **Two-Class Logistic Regresssion**. 7. Also connect the component with the previous **Split Data** component. |
|  | 1. In the components panel enter **Train Model** into the search input field. 2. Drag to the canvas and click it. 3. In the right panel for the **Label column** option click the **Edit column**. 4. Leave **Column names** option selected and select the **ArrDelay15** column. 5. Click the **Save** button. 6. Connect the component with the previous **Two-Class Averaged Perceptron**. 7. Also connect the component with the previous **Split Data** component. |
|  | 1. In the components panel enter **Score Model** to the search input field. 2. Drag to the canvas. 3. Connect the component with the previous **Train Model** component that is connected with the **Two-Class Logistic Regression**. 4. Also connect the component with the previous **Split Data** component. |
|  | 1. In the components panel enter **Score Model** to the search input field. 2. Drag to the canvas. 3. Connect the component with the previous **Train Model** component that is connected with the **Two-Class Averaged Perceptron**. 4. Also connect the component with the previous **Split Data** component. |
|  | 1. In the components panel enter **Evaluate Model** to the search input field. 2. Drag to the canvas. 3. Connect the component with the 2 previous **Score Model** components. |
|  | 1. The final design should look something like this. |
|  | 1. Click “Submit” button to run the designer experiment. |
|  | 1. Select “Create new” 2. Enter an experiment name. 3. Click the “Submit” button. |
|  | 1. Once the designer experiment has finished. 2. Click on one of the “Train Model” 3. Click the “Create inference pipeline” dropdown. 4. Select “Real-time inference pipeline”. |
|  | 1. Remove the “Evaluate Model” item from the bottom of the Designer. 2. Click the “Submit” button |
|  | 1. Once the designer experiment has finished. 2. Click on one of the “Deploy”. |
|  | 1. Fill in the endpoint name. 2. Select a Compute Type: AKS or ACI 3. Click “Deploy” |

## Azure Arc

This is only required if you want to demo the Hybrid ML with Azure Arc module.

For this section, we will need two resources:

1. Kubernetes cluster: we will use an AKS cluster under another resource group to simulate “on premises.” Follow the instructions [here](https://docs.microsoft.com/en-us/azure/aks/kubernetes-walkthrough-portal#create-an-aks-cluster). Make sure to set the Authentication method to Service Principal and grant the principal the role of Kubernetes Cluster - Azure Arc Onboarding.
2. Azure Arc-enabled Kubernetes cluster: we will enroll our AKS cluster in Arc to complete this step. Follow the steps described [here](https://docs.microsoft.com/en-us/azure/azure-arc/kubernetes/quickstart-connect-cluster).

Once these resources are available, attach the Azure Arc-enabled Kubernetes cluster to your Azure Machine Learning workspace using the instructions [here](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-attach-arc-kubernetes).

## Private Workspace

This is only required if you want to demo the Security & Enterprise Readiness module.

This requires the Azure CLI. The following instructions assume using a Windows command prompt but can be adopted Linux.

1. Open a command prompt.
2. Execute: az login
3. Execute: az account set -s <your subscription id>
4. Execute: setx rg <resource-group-name>
5. Restart command prompt.
6. Navigate to the **armTemplates** folder.
7. Execute: az deployment group create -n deploymentName -g %rg% -f azuredeploy.json -p azuredeploy.parameters.json
8. Create a Data Science Virtual Machine within your **subnet,** follow the instructions from the official [Microsoft Documentation](https://docs.microsoft.com/en-us/azure/machine-learning/data-science-virtual-machine/provision-vm#create-your-dsvm) on how to deploy the virtual machine.
9. Connect to the Data Science virtual machine through RDP. From here you can launch Edge browser and access the Azure Machine Learning workspace.
10. Enable ML Managed Identity by following the steps in the [Microsoft Documentation](https://docs.microsoft.com/en-us/azure/machine-learning/how-to-enable-studio-virtual-network#datastore-azure-storage-account).
11. Create a Compute cluster behind the VNet:
    1. Execute: az network nsg rule create -n batch --nsg-name <network-security-group-name> -g %rg% --direction Inbound --priority 400 --source-address-prefixes BatchNodeManagement --source-port-ranges \* --destination-port-ranges 29876-29877 --protocol Tcp --access Allow
    2. Execute: az network nsg rule create -n aml --nsg-name <network-security-group-name> -g %rg% --direction Inbound --priority 410 --source-address-prefixes AzureMachineLearning --source-port-ranges \* --destination-port-ranges 44224 --protocol Tcp --access Allow
    3. Execute: az ml computetarget create amlcompute -n cpu-cluster --min-nodes 1 --max-nodes 8 -s STANDARD\_D3\_V2 --vnet-name <vnet-name> --vnet-resourcegroup-name %rg% --subnet-name <subnet-name> --workspace-name <ml-workspace-name> -g %rg%
    4. Execute: az ml computetarget create computeinstance -n gdm-vm -s "STANDARD\_D3\_V2" --vnet-name <vnet-name> --vnet-resourcegroup-name %rg% --subnet-name <subnet-name> --workspace-name <ml-workspace-name> -g %rg%
12. Upload the flight-delay-automl-private.ipynb into the workspace and execute the cells. After the AKS Cluster creation step stand by and grant the following access to the AKS Service Principal:
    1. Grant “Network Contributor” access to AKS Service Principal:
       1. Execute: az aks show -n <aks-cluster-name> -g %rg% --query servicePrincipalProfile.clientId -o tsv
       2. Execute: az group show -n %rg% --query id -o tsv
       3. Execute: az aks show -g %rg% -n <aks-cluster-name> --query "identity"
       4. Execute: az role assignment create --assignee <principal-object-id> --role "Network Contributor" --scope <resource-group-resource-id>

# Power BI

## Consume Model

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
|  | 1. Open **Power BI Desktop** locally. 2. Open the `**Flight Delay Report 2020.pbix**` report from the **powerbi** folder. |
|  | 1. Click the **Edit query** in the popup menu for the **flightdatasetwithprediction** dataset. |
|  | 1. Right click the **Predicted** column and click on the **Remove** option. |
|  | 1. Click the **Azure Machine Learning** button in the top panel. |
|  | 1. Select the created model from the previous section and click the **OK** button. |
|  | 1. If you get this warning, click the **Continue** button. 2. Select **Ignore Privacy Levels checks for this file**. 3. Click on the **Save** button. |
|  | 1. Check the new created column at the end. 2. Right click the column and rename it to **Predicted**. 3. Click the **Close and Apply** button. |
|  | 1. Click the **Apply changes** to update the data and wait for the prediction to be completed. |

## 

## Adding Gauge Chart for Predicted Value

|  |  |
| --- | --- |
| **Screenshot** | **Steps** |
|  | 1. After updating the data with the predicted value from the model, click the **Gauge** icon in the visualizations list to add the new chart. 2. From the **flightdatasetwithprediction** dataset find the **Predicted** field and drag it to the **Value** input. |
|  | 1. Make sure that the chart is selected to allow the edition of the properties. 2. Click the **Format icon** to edit the title text. 3. Change the value of the **Title text** to **Delays**. |