www.advancinganalytics.co.uk

DEVOPS FOR DATABRICKS

@ADVANCINGANALYTICS @ADVANALYTICSUK /ADVANCING ANALYTICS

Microsoft® Most Valuable Professional MVP

Award Categories
Data Platform
First year awarded:
2021
Number of MVP Awards:
2

Microsoft Data Platform MVP, Senior Advancing Analytics Consultant specializing in Data Engineering & Cloud.

Over 16 years’ experience working in Software & Data Engineering, most recently working with Microsoft Data Platform, Scala, Kafka and various cloud tech

BSc in Multimedia Computing & Business, and a HND in Visual Communication

ANNA-MARIA-WYKES

@ANNAWYKES

ADVANCING ANALYTICS

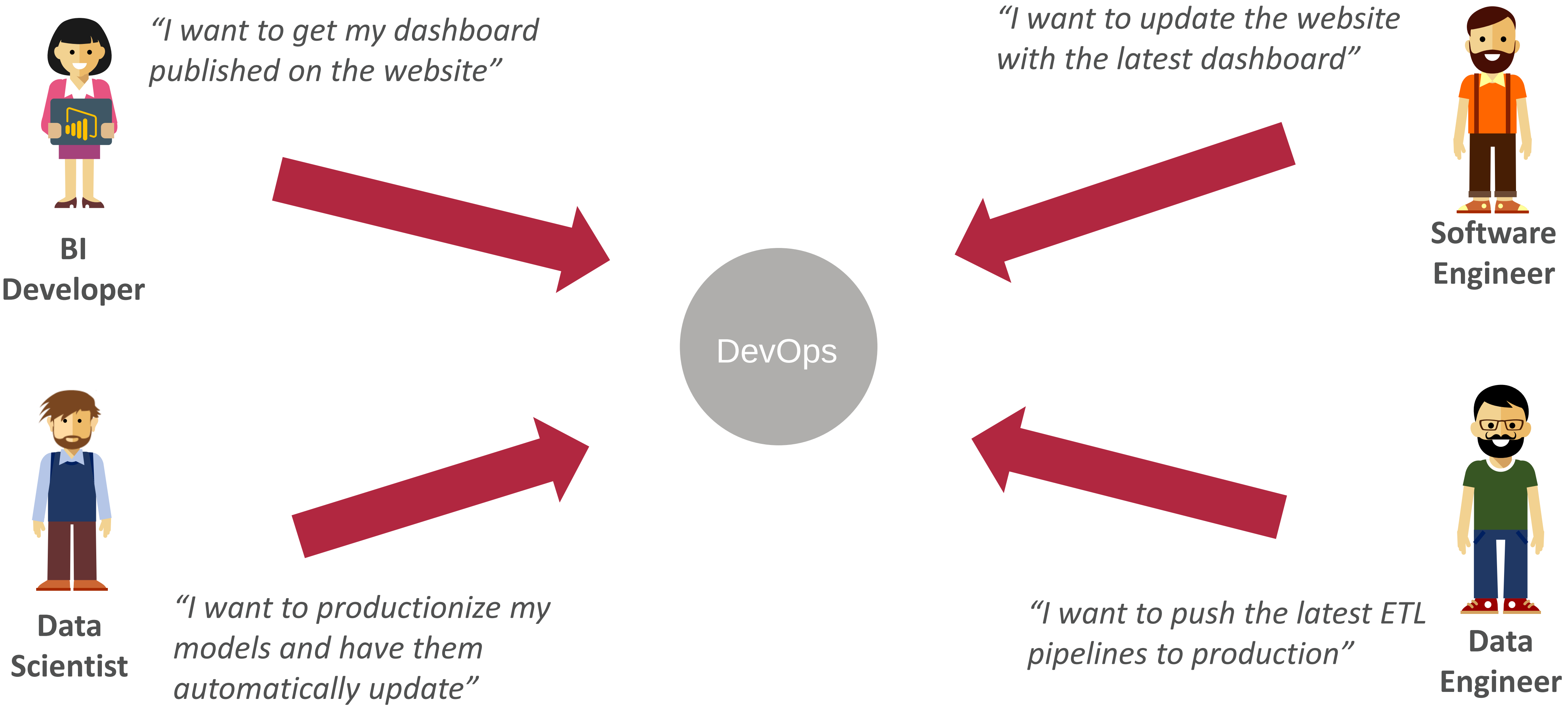
WHAT'S THE PROBLEM?

As a consultant and Data Engineer I’ve lost count of the number of times I’ve been asked “how do we do DevOps for Databricks?”, and the simple answer is, it depends. I know this sounds like a typical consultant’s response, but please bear with me.

AGENDA

- DevOps Theory
  - CI/CD (Continuous Integration/Continuous Deployment)
  - IaC (Infrastructure as Code)
- IaC & CI/CD tools
  - Databricks Rest API
  - Terraform
  - Pulumi
- DevOps Tools
  - Azure DevOps
  - Github Actions

WHAT IS DEVOPS



BI Developer

"I want to get my dashboard published on the website"

Software Engineer

"I want to update the website with the latest dashboard"

DevOps

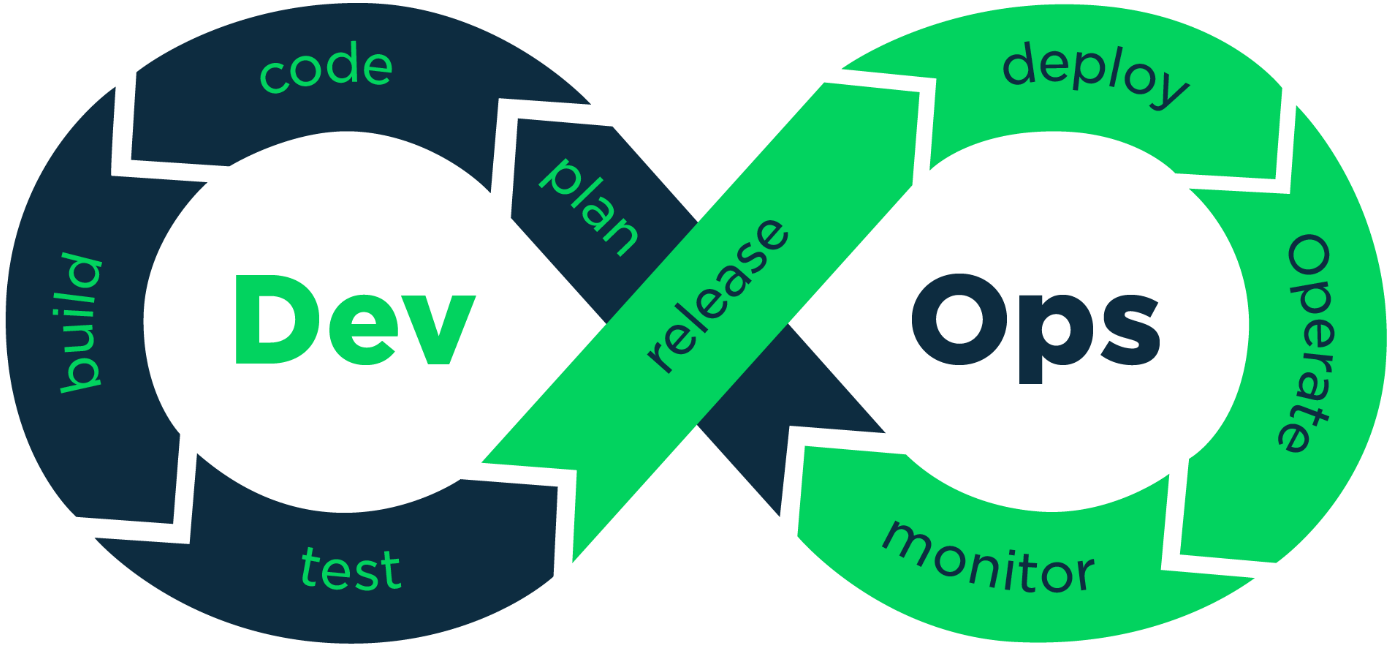
Data Scientist

"I want to productionize my models and have them automatically update"

Data Engineer

"I want to push the latest ETL pipelines to production"

THE DEVOPS DREAM



DEVOPS TOOLS

Continuous Integration/Continuous Deployment (CI/CD)

Azure DevOps

circleci

Jenkins

GitLab

Octopus Deploy

Infrastructure as Code (IAC)

ARM
(Azure Resource Manager)
Templates

pulumi

Terraform

Azure Bicep

THE BIG QUESTION

How do we do DevOps for Databricks

?

THREE DIFFERENT APPROACHES

- Databricks REST API
- Terraform
- Pulumi

DATABRICKS REST API & PYTHON

RESOURCES

- https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/
- https://github.com/AnnaWykes/devops-for-databricks

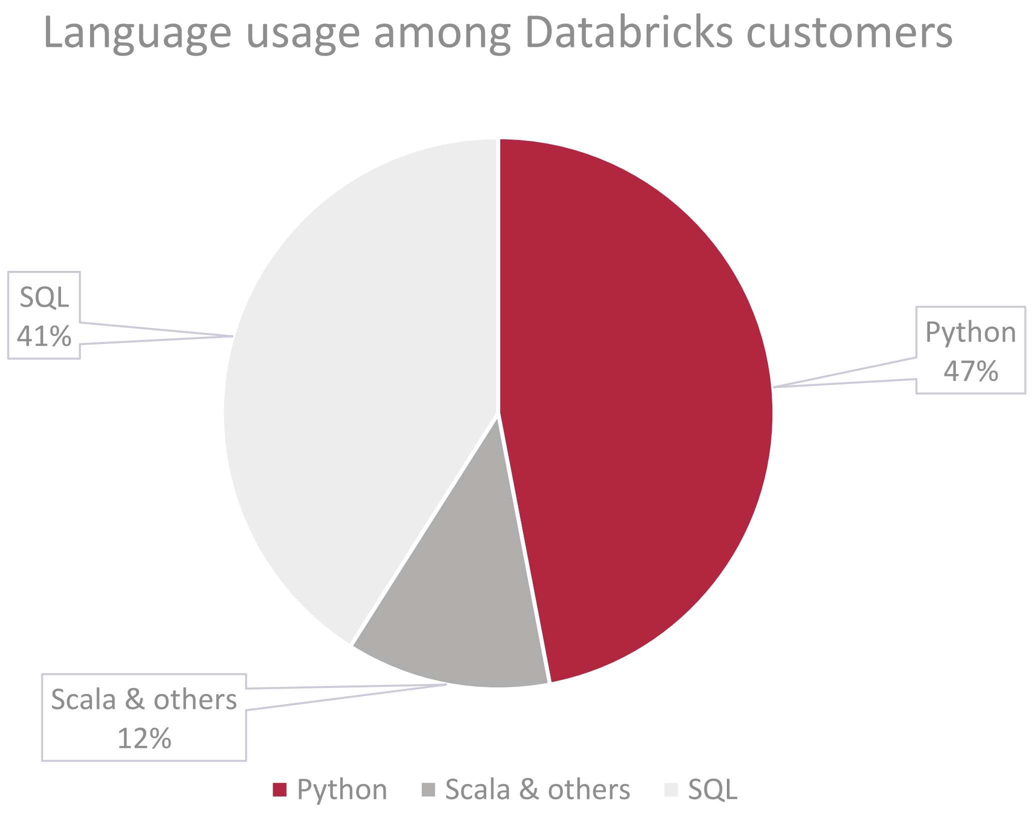
WHAT IS IT

- Rest API provided as part of any Databricks instance

- Azure documentation can be found here https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/

- Can be used in any programming language of choice, or start off using tooling such as Postman

WHY PYTHON?


Language usage among Databricks customers
SQL
41%
Python
47%
Scala & others
12%
■ Python
■ Scala & others
■ SQL

CREATE A CLUSTER

```
DBRKS_CLUSTER_ID = {'cluster_id': os.environ['CLUSTER-ID']}



def create_cluster():
    DBRKS_START_ENDPOINT = 'api/2.0/clusters/create'
    """{
     "cluster_name": "my-cluster",
     "spark_version": "7.3.x-scala2.12",
    "node_type_id": "Standard_D3_v2",
    "spark_conf": {
    "spark.speculation": true
    },
   "num_workers": 2
    }"""

    response = requests.post(os.environ['DBX-WORKSPACE-URL'] + DBRKS_START_ENDPOINT,
                             headers=DBRKS_REQ_HEADERS, json=DBRKS_CLUSTER_ID)
    if response.status_code != 200:
        raise Exception(json.loads(response.content))
```

Create cluster method

Call to Rest API

MONITOR CLUSTER

```
def get_dbrks_cluster_info():
    DBRKS_INFO_ENDPOINT = 'api/2.0/clusters/get'
    response = requests.get(os.environ['DBX-WORKSPACE-URL'] + DBRKS_INFO_ENDPOINT,
                            headers=DBRKS_REQ_HEADERS, params=DBRKS_CLUSTER_ID)
    if response.status_code == 200:
        return json.loads(response.content)
    else:
        raise Exception(json.loads(response.content))
```

Get cluster info

```
def start_dbrks_cluster():
    DBRKS_START_ENDPOINT = 'api/2.0/clusters/start'
    response = requests.post(os.environ['DBX-WORKSPACE-
URL'] + DBRKS_START_ENDPOINT, headers=DBRKS_REQ_HEADERS, json=DBRKS_CLUSTER_ID)
    if response.status_code != 200:
        raise Exception(json.loads(response.content))
```

Start cluster

```
def restart_dbrks_cluster():
    DBRKS_RESTART_ENDPOINT = 'api/2.0/clusters/restart'
    response = requests.post(
        os.environ['DBX-WORKSPACE-URL'] + DBRKS_RESTART_ENDPOINT,
        headers=DBRKS_REQ_HEADERS,
        json=DBRKS_CLUSTER_ID)
    if response.status_code != 200:
        raise Exception(json.loads(response.content))
```

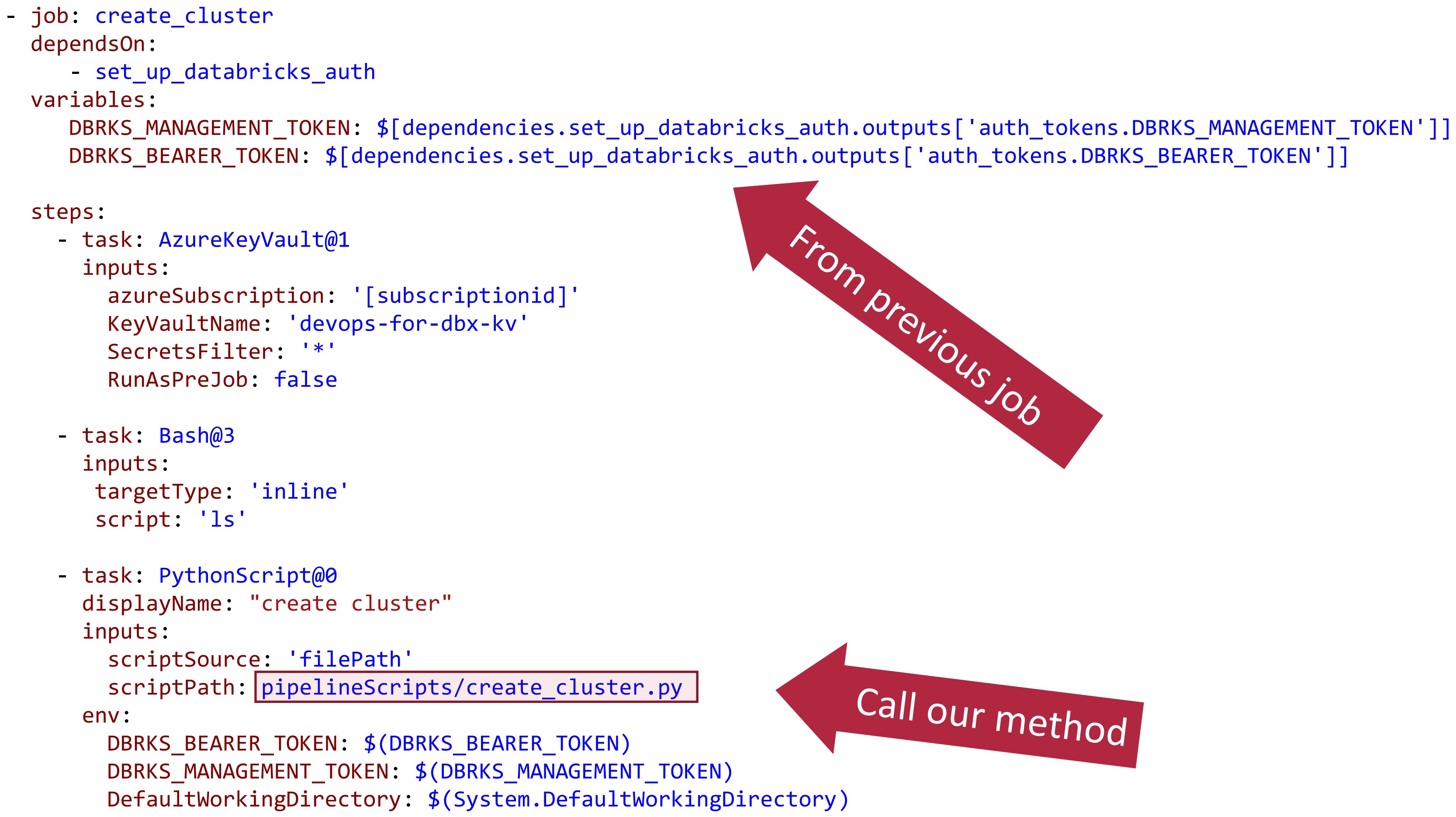
Restart cluster

MONITOR CLUSTER

```python
def manage_dbrks_cluster_state():
    await_cluster = True
    started_terminated_cluster = False
    cluster_restarted = False
    start_time = time.time()
    loop_time = 1200  # 20 Minutes
    while await_cluster:
        current_time = time.time()
        elapsed_time = current_time - start_time
        if elapsed_time > loop_time:
            raise Exception('Error: Loop took over {} seconds to run.'.format(loop_time))
        if get_dbrks_cluster_info()['state'] == 'TERMINATED':
            print('Starting Terminated Cluster')
            started_terminated_cluster = True
            start_dbrks_cluster()
            time.sleep(60)
        elif get_dbrks_cluster_info()['state'] == 'RESTARTING':
            print('Cluster is Restarting')
            time.sleep(60)
        elif get_dbrks_cluster_info()['state'] == 'PENDING':
            print('Cluster is Pending Start’) time.sleep(60)
        elif get_dbrks_cluster_info()['state'] == 'RUNNING' and not cluster_restarted and not started_terminated_cluster:
            print('Restarting Cluster’) cluster_restarted = True
            restart_dbrks_cluster()
        else:
            print('Cluster is Running’) await_cluster = False
```

Looping

AZURE DEVOPS: MONITOR CLUSTER



```
- job: create_cluster
  dependsOn:
     - set_up_databricks_auth
  variables:
     DBRKS_MANAGEMENT_TOKEN: $[dependencies.set_up_databricks_auth.outputs['auth_tokens.DBRKS_MANAGEMENT_TOKEN']]
     DBRKS_BEARER_TOKEN: $[dependencies.set_up_databricks_auth.outputs['auth_tokens.DBRKS_BEARER_TOKEN']]

  steps:
    - task: AzureKeyVault@1
      inputs:
        azureSubscription: '[subscriptionid]'
        KeyVaultName: 'devops-for-dbx-kv'
        SecretsFilter: '*'
        RunAsPreJob: false

    - task: Bash@3
      inputs:
       targetType: 'inline'
       script: 'ls'

    - task: PythonScript@0
      displayName: "create cluster"
      inputs:
        scriptSource: 'filePath'
        scriptPath: pipelineScripts/create_cluster.py
      env:
        DBRKS_BEARER_TOKEN: $(DBRKS_BEARER_TOKEN)
        DBRKS_MANAGEMENT_TOKEN: $(DBRKS_MANAGEMENT_TOKEN)
        DefaultWorkingDirectory: $(System.DefaultWorkingDirectory)
```

From previous job

Call our method

AZURE DEVOPS: UPLOAD NOTEBOOK

```
- job: upload_notebooks
  dependsOn:
     - set_up_databricks_auth
  variables:
     DBRKS_MANAGEMENT_TOKEN: $[dependencies.set_up_databricks_auth.outputs['auth_tokens.DBRKS_MANAGEMENT_TOKEN']]
     DBRKS_BEARER_TOKEN: $[dependencies.set_up_databricks_auth.outputs['auth_tokens.DBRKS_BEARER_TOKEN']]

  steps:
    - task: AzureKeyVault@1
      inputs:
        SubscriptionName: '[subscriptionid]'
        KeyVaultName: 'devops-for-dbx-kv'
        SecretsFilter: '*'
        RunAsPreJob: false

    - task: PythonScript@0
      displayName: "upload notebooks to DBX"
      inputs:
        scriptSource: 'filePath'
        scriptPath: pipelineScripts/upload_notebooks_to_dbx.py
      env:
        DBRKS_BEARER_TOKEN: $(DBRKS_BEARER_TOKEN)
        DBRKS_MANAGEMENT_TOKEN: $(DBRKS_MANAGEMENT_TOKEN)
        DefaultWorkingDirectory: $(System.DefaultWorkingDirectory)
```

From previous job

Call our method

ADVANCING ANALYTICS

DATABRICKS REST API & PYTHON

- Azure DevOps Pipeline
- Github Action

TERRAFORM

RESOURCES

- https://registry.terraform.io/providers/databrickslabs/databricks/latest/docs
- https://github.com/AnnaWykes/devops-for-databricks

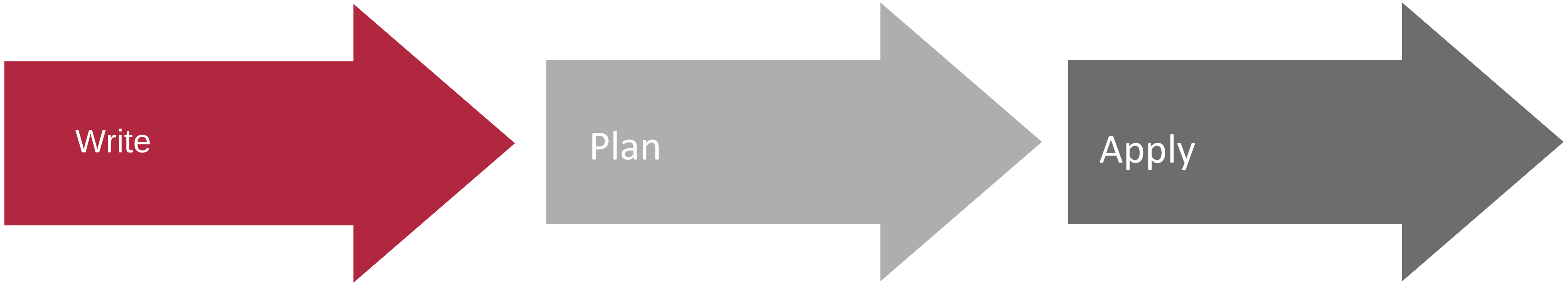
WHAT IS TERRAFORM?

One of the most popular IAC (Infrastructure as Code) tools

Terraform is a tool for building, changing, and versioning infrastructure safely and efficiently.

Terraform can manage existing and popular service providers as well as custom in-house solutions.

Configuration files describe to **Terraform** the components needed to run a single application or your entire datacenter.



BENEFITS

- State Management
- Cross Cloud (Azure, AWS, GCP)
- Solve Issues of provisioning complicated infrastructure (often encountered with other IAC tools)
- Simple syntax that allows for easy modularity
- High level description of infrastructure

GETTING STARTED

```
 terraform {
  required_providers {
   azurerm = {
    source = "hashicorp/azurerm"
    version = "~>2.31.1"
   }
   databricks = {
    source  = "databrickslabs/databricks"
    version = "0.3.2"
   }
  }
 }

provider "azurerm" {
   features {}
}

provider "databricks" {
  azure_workspace_resource_id = azurerm_databricks_workspace.databricks_workspace.id
}
```

Azure Provider

Databricks Provider

DATABRICKS WORKSPACE

```
/* Create a resource group for our databricks workspace to be deployed to*/
resource "azurerm_resource_group" "rg" {
    name     = var.resource_group_name
    location = var.azure_region
}

/* Create a Databricks workspace */
resource "azurerm_databricks_workspace" "databricks_workspace" {
    name                        = var.databricks_name
    resource_group_name         = azurerm_resource_group.rg.name
    managed_resource_group_name = var.databricks_managed_resource_group_name
    location                    = var.azure_region
    sku                         = var.databricks_sku_name
}
```

Resource group

Databricks workspace

DATABRICKS CLUSTER

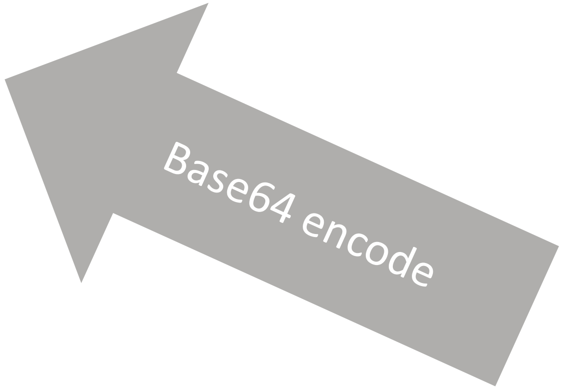
```
/* Create databricks cluster */
resource "databricks_cluster" "databricks_cluster_01" {
    cluster_name            = var.cluster_name
    spark_version           = var.spark_version
    node_type_id            = var.node_type_id
    autotermination_minutes = var.autotermination_minutes
    autoscale {
      min_workers = 1
      max_workers = 2
    }
    # Create Libraries
    library {
      pypi {
          package = "pyodbc"
          }
    }
    library {
      maven {
        coordinates = "com.microsoft.azure:spark-mssql-connector_2.12_3.0:1.0.0-alpha"
      }
    }
    custom_tags = {
      Department = "Data Engineering"
    }

    azure_attributes {
      availability       = "ON_DEMAND_AZURE"
      first_on_demand    = 1
      spot_bid_max_price = -1
    }
  }
```

Add Libraries

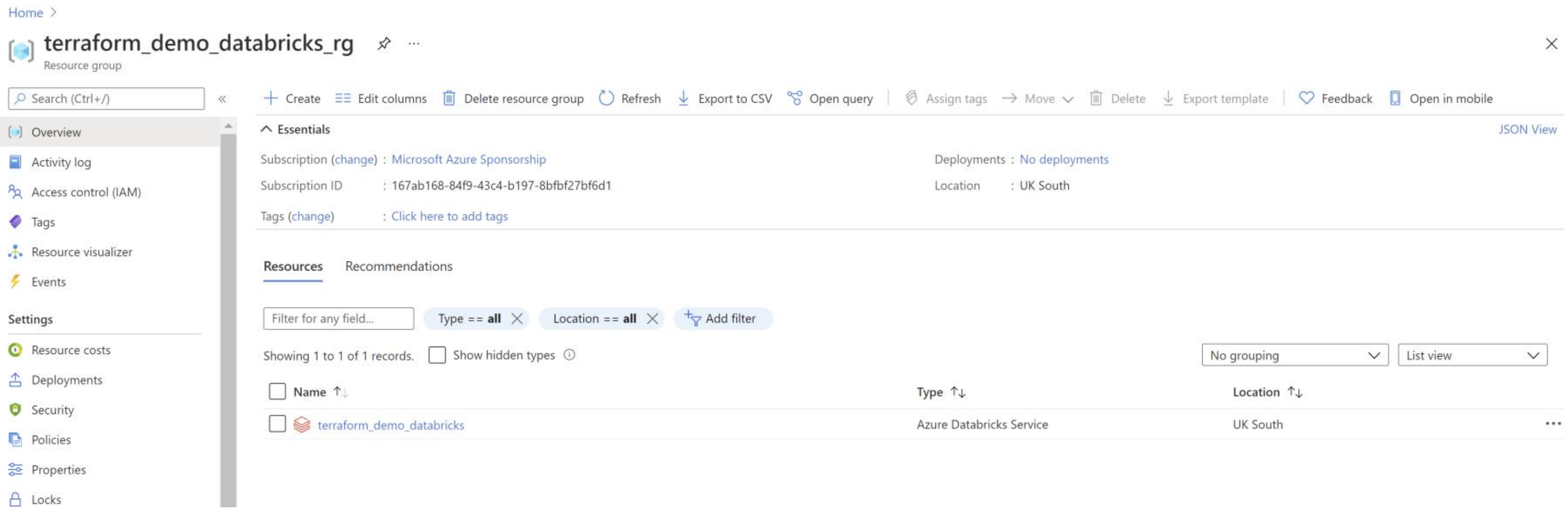
UPLOAD NOTEBOOKS

```
/* Create Databricks notebook */
resource "databricks_notebook" "notebook" {
    content_base64 = base64encode("print('Welcome to Databricks-Labs notebook')")
    path       = var.notebook_path
    language   = "PYTHON"
  }
```



Base64 encode

END RESULT



terraform_demo_databricks_rg

Resource group

Essentials

Subscription (change) : Microsoft Azure Sponsorship

Subscription ID : 167ab168-84f9-43c4-b197-8bfbf27bf6d1

Tags (change) : Click here to add tags

Deployments : No deployments

Location : UK South

Type == all

Location == all

Showing 1 to 1 of 1 records. □ Show hidden types

No grouping

List view

□ Name	Type	Location
□ terraform_demo_databricks	Azure Databricks Service	UK South

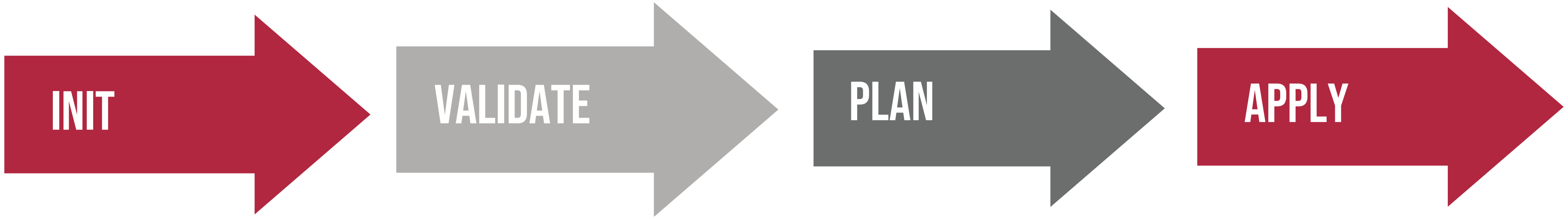
TERRAFORM HANDLES STATE

- Terraform recognises when you are creating vs when you are amending a resource

- You can import existing resources into Terraform and work with them

- Terraform picks up on changes that have happened outside of itself, and let you know if it’s going to change/delete anything

TERRAFORM CLI CORE COMMANDS



INIT

VALIDATE

PLAN

APPLY

TERRAFORM

- Demo

PULUMI & PYTHON

RESOURCES

- https://www.pulumi.com/registry/packages/azure/api-docs/databricks/
- https://www.pulumi.com/docs/get-started/azure/
- https://github.com/AnnaWykes/devops-for-databricks

WHY PULUMI?

- Write IAC (Infrastructure as Code) in you language of choice: C#, Python, Go, Typescript
- State Management
- Utilizes Terraform providers and can work along side Terraform
- If functionality doesn't exist it’s easy to use SDK’s/APIs along side Pulumi in the same language of choice

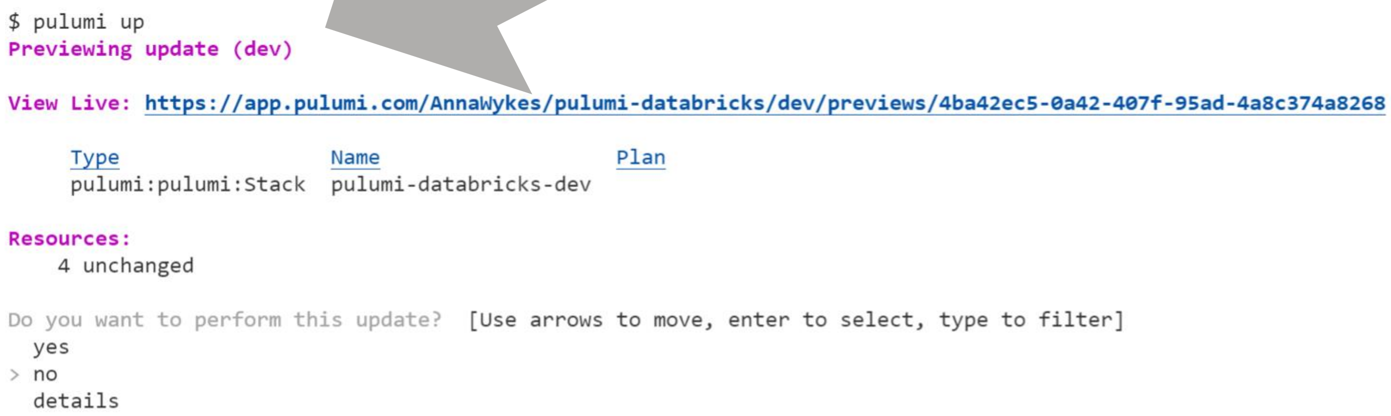
GETTING STARTED

```python
import pulumi
from pulumi_azure_native import storage
from pulumi_azure_native import resources
from pulumi_azure_native import databricks as dbx

# Create an Azure Resource Group
resource_group = resources.Res                  'pulumi_databricks_resource_group')
```

Creating a resource group

“Pulumi up” command



```
$ pulumi up
Previewing update (dev)

View Live: https://app.pulumi.com/AnnaWykes/pulumi-databricks/dev/previews/4ba42ec5-0a42-407f-95ad-4a8c374a8268

     Type                 Name                    Plan
     pulumi:pulumi:Stack  pulumi-databricks-dev

Resources:
    4 unchanged

Do you want to perform this update?  [Use arrows to move, enter to select, type to filter]
  yes
> no
  details
```

CREATE A STORAGE ACCOUNT

```
# Create an Azure resource (Storage Account)
account = storage.StorageAccount('pulumidbxsa',
    resource_group_name=resource_group.name,
    sku=storage.SkuArgs(
        name=storage.SkuName.STANDARD_LRS,
    ),
    kind=storage.Kind.STORAGE_V2)

# Export the primary key of the Storage Account
primary_key = pulumi.Output.all(resource_group.name, account.name) \
    .apply(lambda args: storage.list_storage_account_keys(
        resource_group_name=args[0],
        account_name=args[1]
    )).apply(lambda accountKeys: accountKeys.keys[0].value)

pulumi.export("primary_storage_key", primary_key)
```

Grab storage account key for later use

CREATE DATABRICKS WORKSPACE

Creating Databricks workspace

```
#create Databricks workspace
workspace = dbx.Workspace("workspace",
    location="westus",
    managed_resource_group_id="/subscriptions/[subscriptionid]/resourceGroups/pulumi_databricks_managed_resource_group",
    parameters=dbx.WorkspaceCustomParametersArgs(
        prepare_encryption=dbx.WorkspaceCustomBooleanParameterArgs(
            value=True,
        ),
    ),
    resource_group_name=resource_group.name,
    workspace_name="pulumi_databricks_workspace")
```

CREATE DATABRICKS GROUP

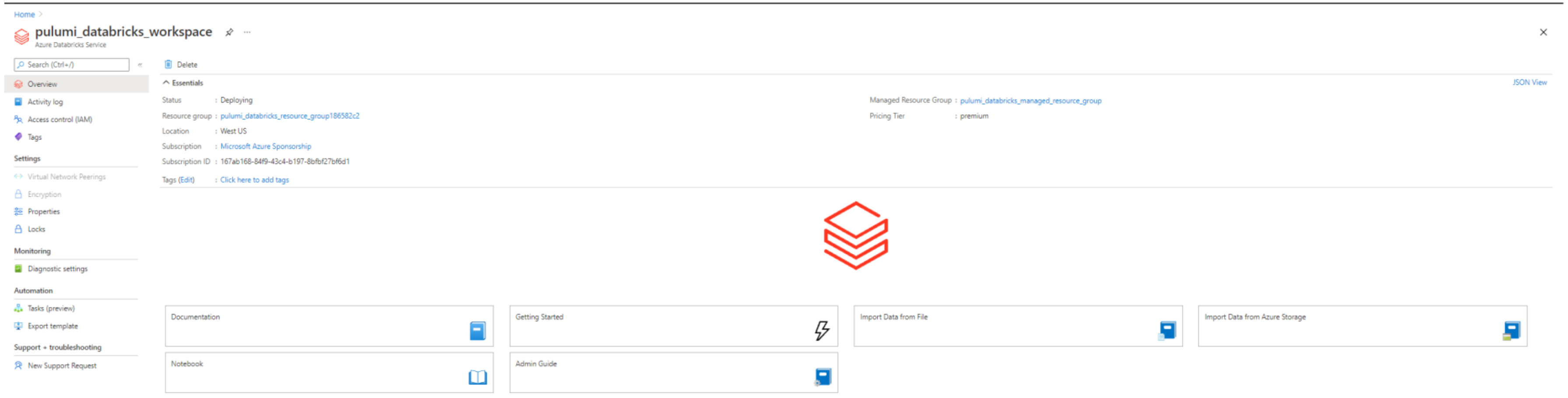
Import Databricks provider

```
import pulumi_databricks as databricks

group = databricks.Group("py-group", display_name="DataGrillen")
```

Create Databricks group

END RESULT



pulumi_databricks_workspace

Azure Databricks Service

Status : Deploying

Resource group : pulumi_databricks_resource_group186582c2

Location : West US

Subscription : Microsoft Azure Sponsorship

Subscription ID : 167ab168-84f9-43c4-b197-8bfbf27bf6d1

Tags (Edit) : Click here to add tags

Managed Resource Group : pulumi_databricks_managed_resource_group

Pricing Tier : premium

THANK YOU

RESOURCES

- https://www.pulumi.com/registry/packages/azure/api-docs/databricks/
- https://www.pulumi.com/docs/get-started/azure/
- https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/
- https://registry.terraform.io/providers/databrickslabs/databricks/latest/docs
- https://github.com/AnnaWykes/devops-for-databricks

ANNA-MARIA-WYKES

@ANNAWYKES