PSCONF.EU
PRAGUE23

How Does PowerShell Help Practically Enhance My Pipeline Automation?

Anh Tran

Many thanks to our sponsors:

Chocolatey

Centino systems

PATCH MY PC

ScriptRunner®
The #1 for PowerShell Management

About me.

- Lead Consultant @DevoTeam
- 12+ years of experience in data analytics and data management in different industries such as finance, technology, and wholesale across Asia and Europe
- Featured Articles on United States {CYBERSECURITY} Magazine THANH NIÊN
- **Blog:** *https://analysiswithanh.medium.com*
- in *https://www.linkedin.com/in/anh-tran-thi-lan*

Agenda

- Existing architecture
- New architecture
- Implementation
- Conclusion

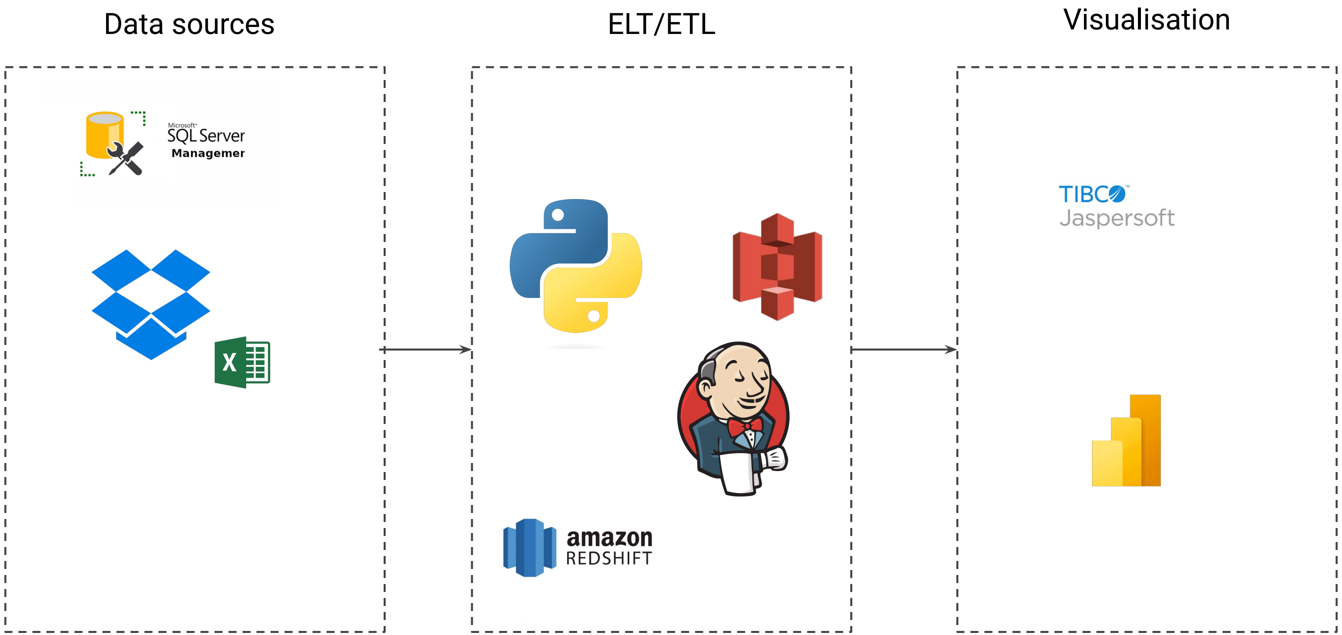
Existing architecture

Background

Problem

1. Lack of visibility
2. Unreliable reporting system
3. Inefficient manual works

Existing Architecture

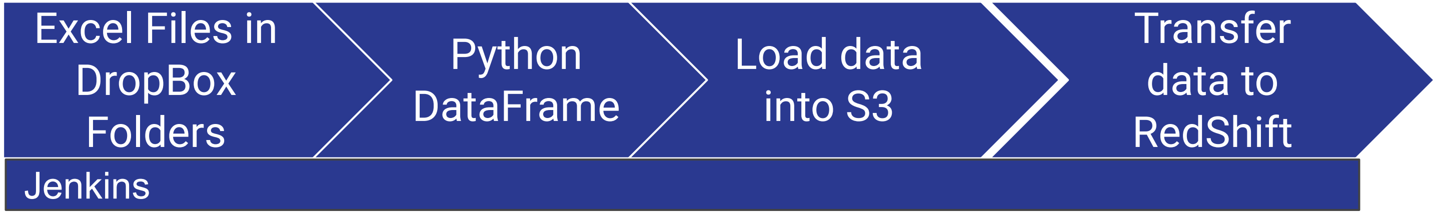


Data sources

ELT/ETL

Visualisation

Existing Architecture



Excel Files in DropBox Folders

Python DataFrame

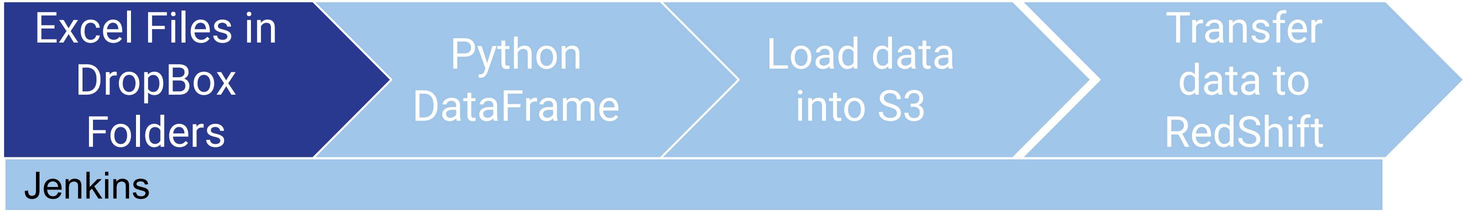
Load data into S3

Transfer data to RedShift

Jenkins

Existing Architecture

Data collection is still manual



Excel Files in DropBox Folders

Python DataFrame

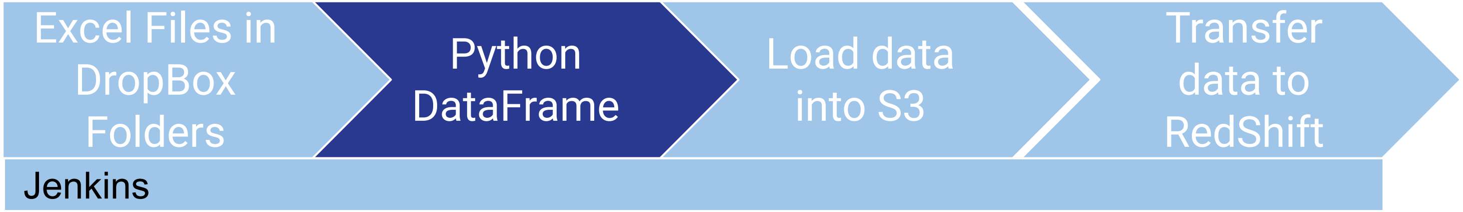
Load data into S3

Transfer data to RedShift

Jenkins

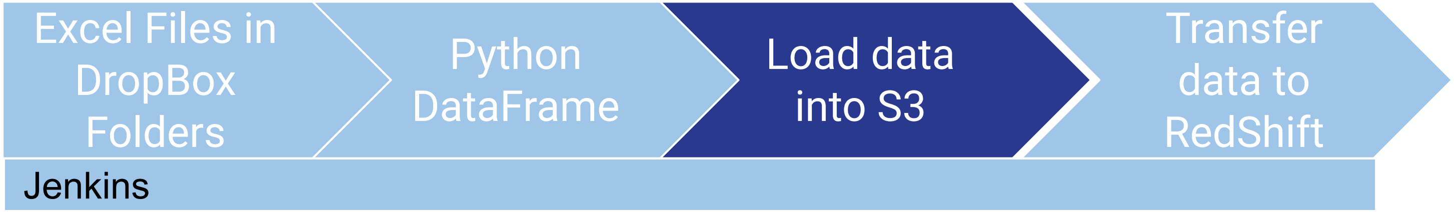
Existing Architecture

The code is inefficient



Existing Architecture

Join DataFrames (Excel/Python DataFrames + RedShift Tables)



Excel Files in DropBox Folders

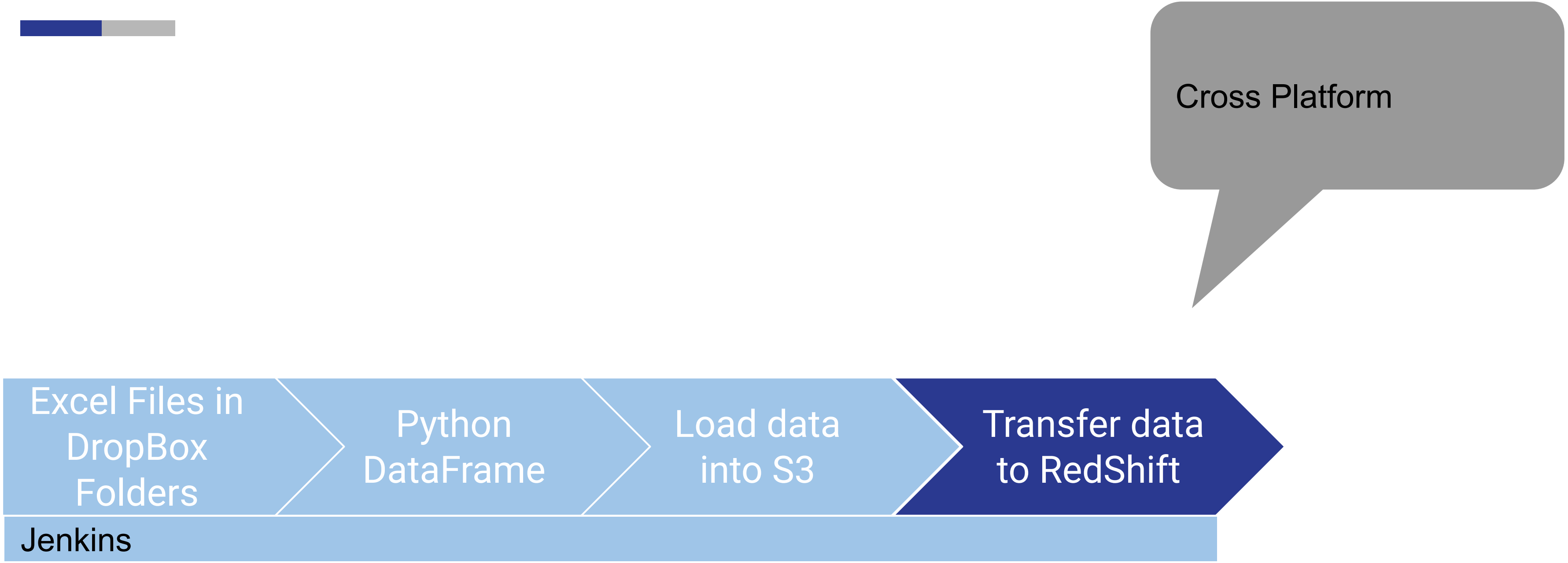
Python DataFrame

Load data into S3

Transfer data to RedShift

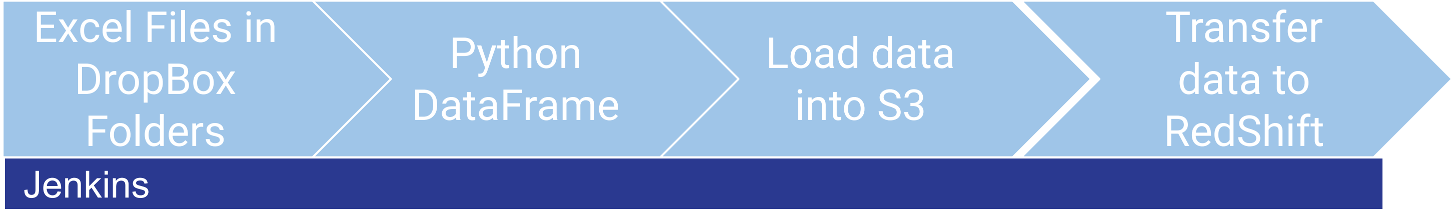
Jenkins

Existing Architecture



Existing Architecture

Code versioning is not well managed



Excel Files in DropBox Folders

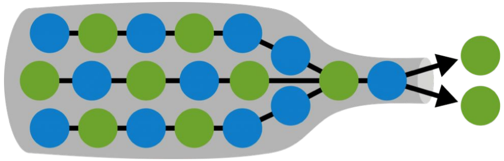
Python DataFrame

Load data into S3

Transfer data to RedShift

Jenkins

Existing Architecture



Limit the team capacity

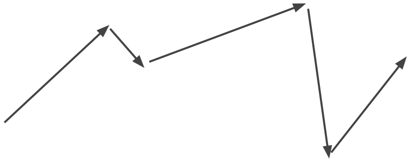
People working on Exclusive VM

Excel Files in DropBox Folders → Python DataFrame → Load data into S3 → Transfer data to RedShift

Jenkins

Why PowerShell and Azure DevOps?

What are the hooks?



- Assess the situation (devops, survey)
- Pilot test
- Deployment

Why?

~~Copycat~~

~~Best Practice~~

NOT necessary the best, but the **best suitable** solution

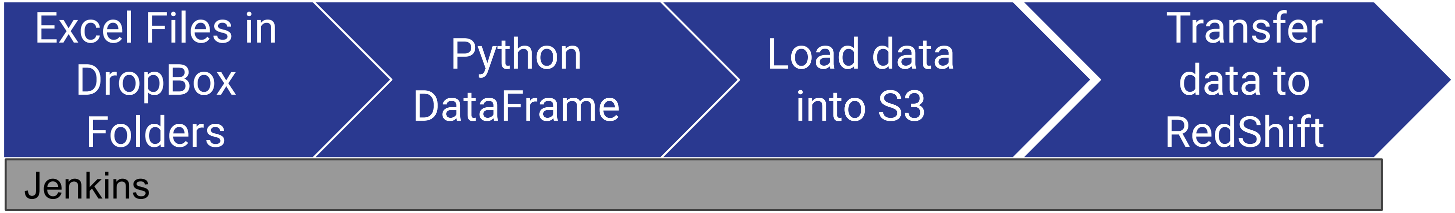
Focus on Business Value

Align architecture strategy with business needs

- M&A
    - Visibility
    - Operate up to speed
    - Scale up
        - Version Control
        - Cross Platform Management

Old

People working on Exclusive VM



Excel Files in DropBox Folders

Python DataFrame

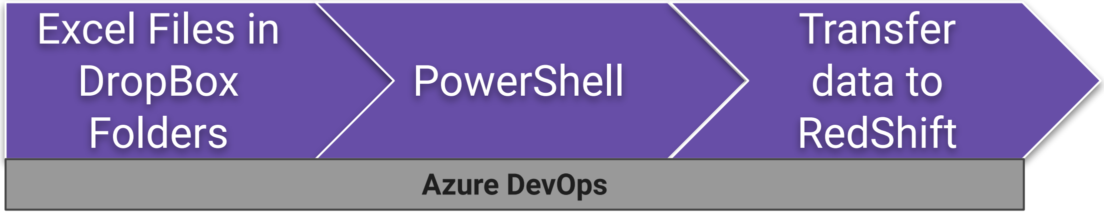
Load data into S3

Transfer data to RedShift

Jenkins

New

Cloud



Excel Files in DropBox Folders

PowerShell

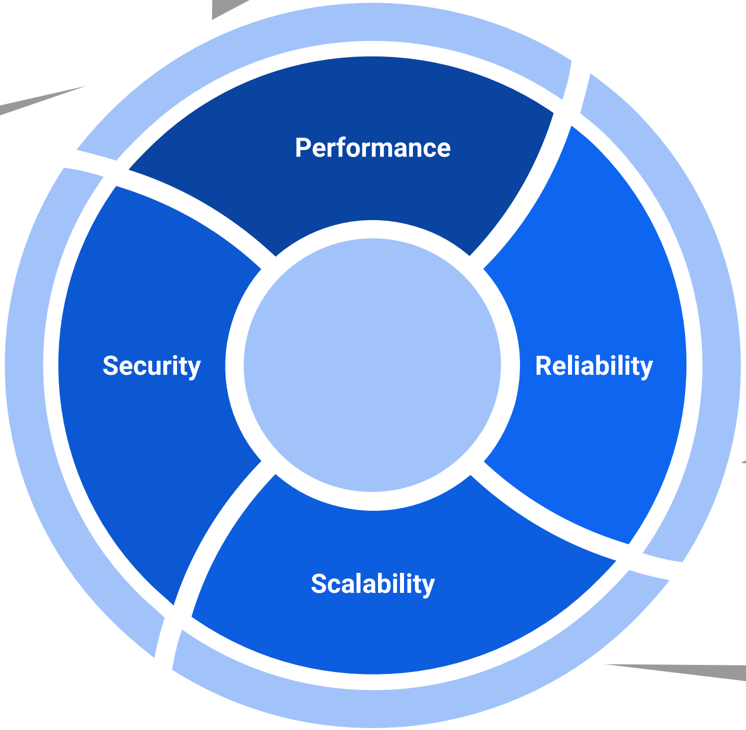
Transfer data to RedShift

Azure DevOps

Outcome

The code is inefficient

Code versioning is not well managed



Data collection is still manual

Cross Platform

Limit team capacity

Outcome

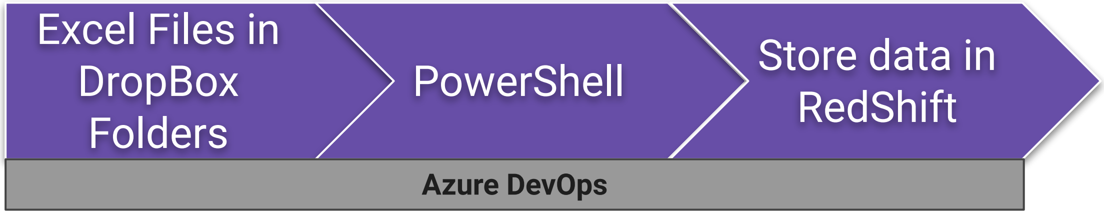
1 Enhance the visibility of business performance fast

2 Reliability

3 Scalability

New Architecture

@dfinke



Excel Files in DropBox Folders

PowerShell

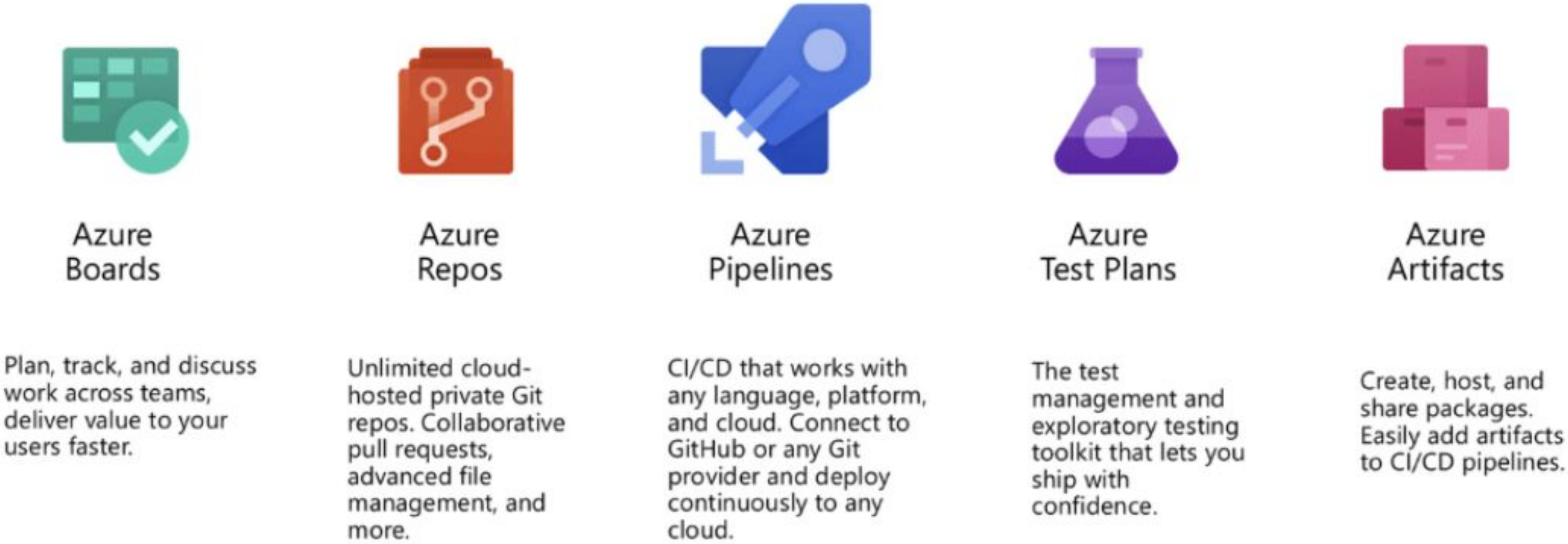
Store data in RedShift

Azure DevOps

Implementation

How?

Azure DevOps



Azure Boards

Plan, track, and discuss work across teams, deliver value to your users faster.

Azure Repos

Unlimited cloud-hosted private Git repos. Collaborative pull requests, advanced file management, and more.

Azure Pipelines

CI/CD that works with any language, platform, and cloud. Connect to GitHub or any Git provider and deploy continuously to any cloud.

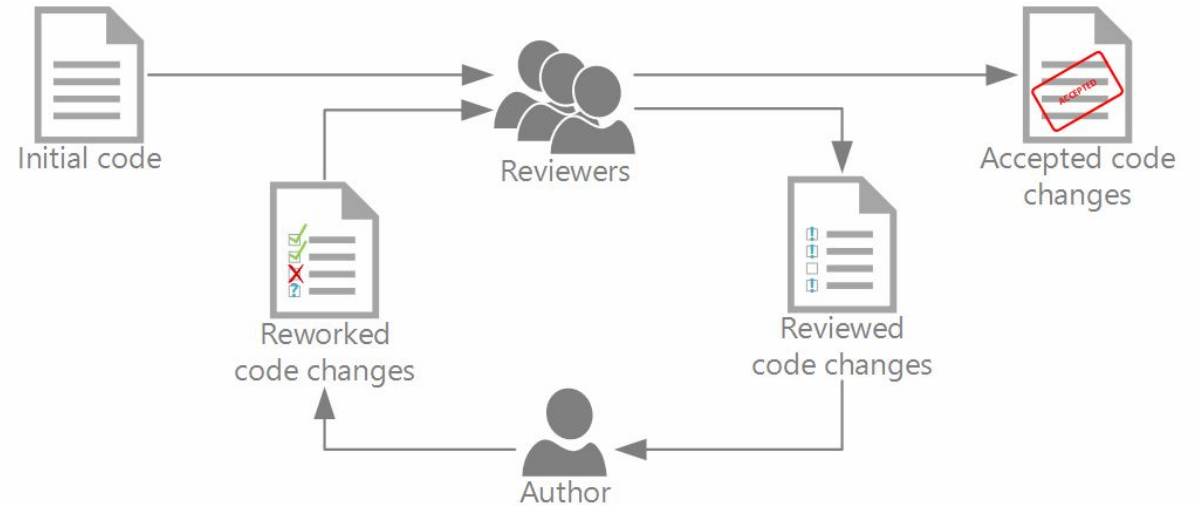
Azure Test Plans

The test management and exploratory testing toolkit that lets you ship with confidence.

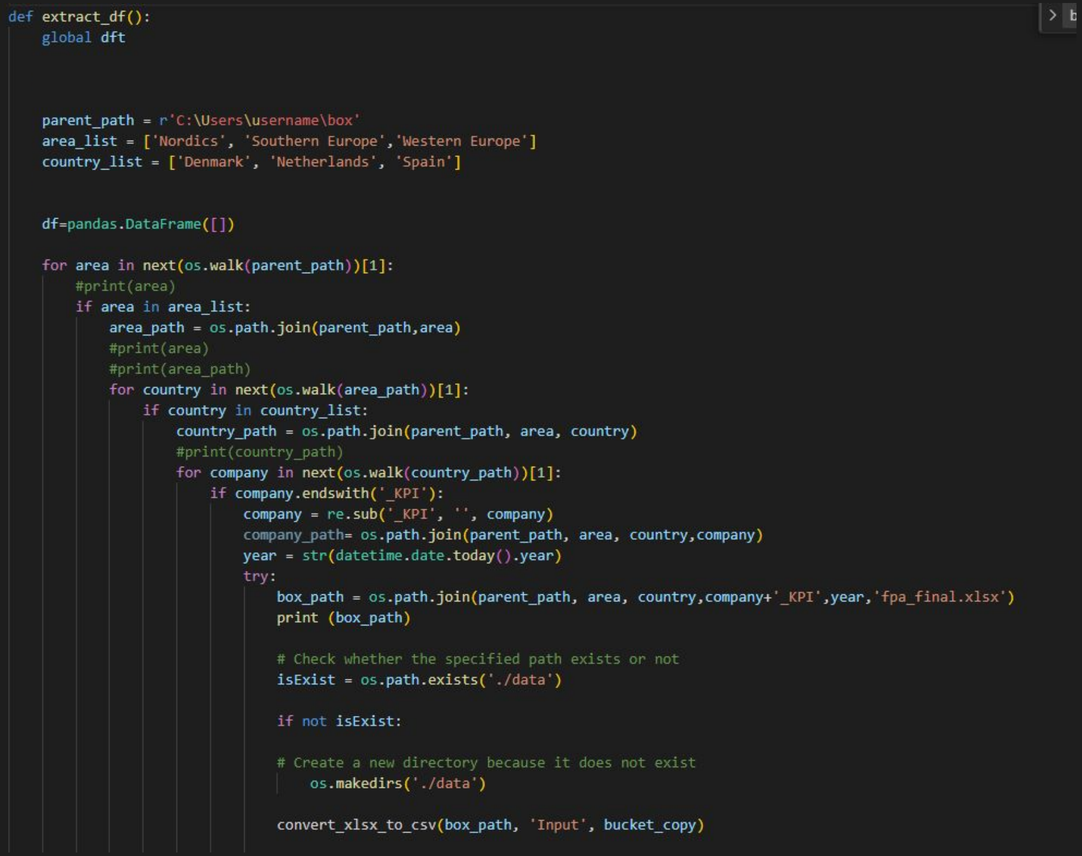
Azure Artifacts

Create, host, and share packages. Easily add artifacts to CI/CD pipelines.

How?

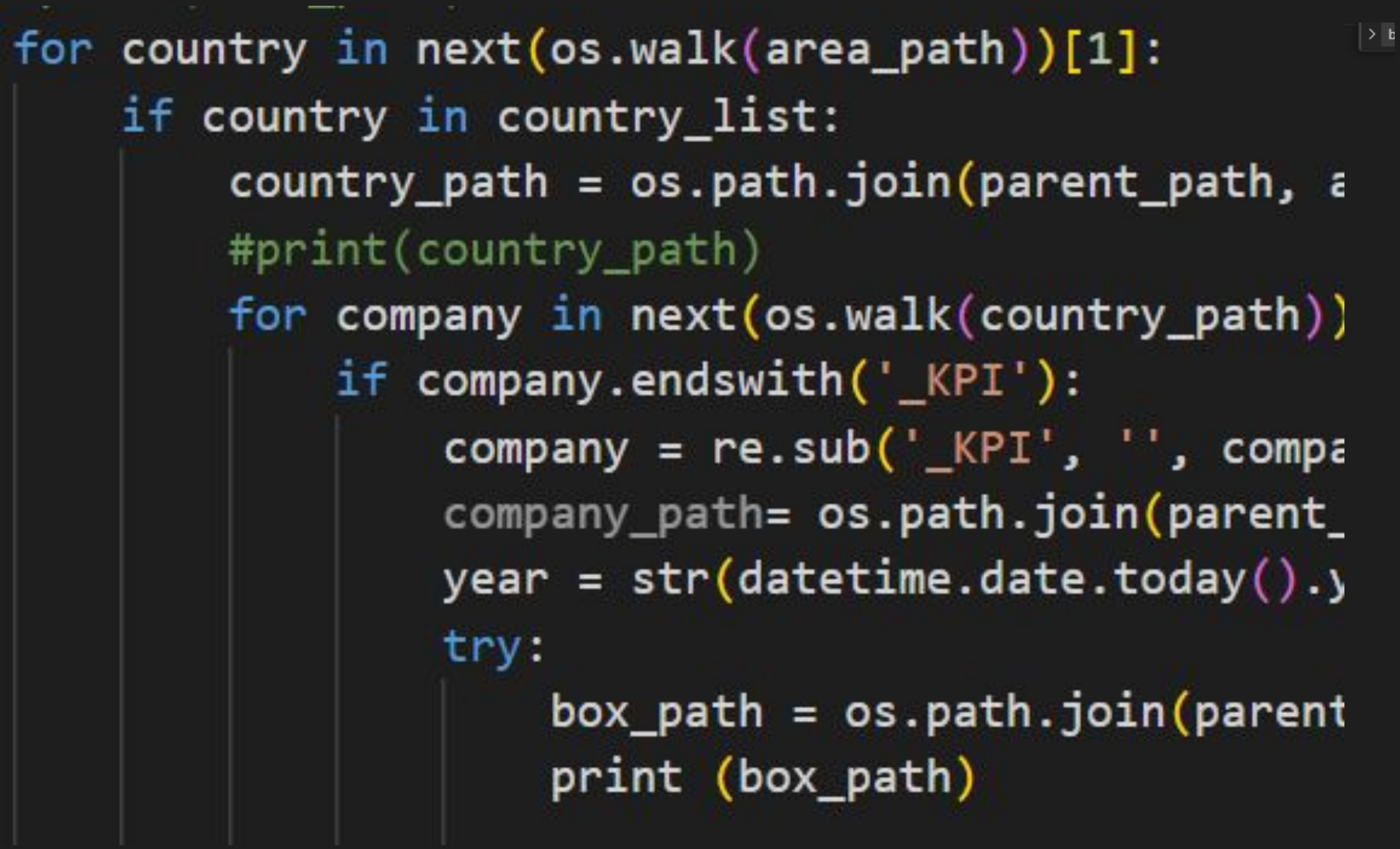

Initial code
Reviewers
Accepted code changes
ACCEPTED
Reworked code changes
Reviewed code changes
Author

Python



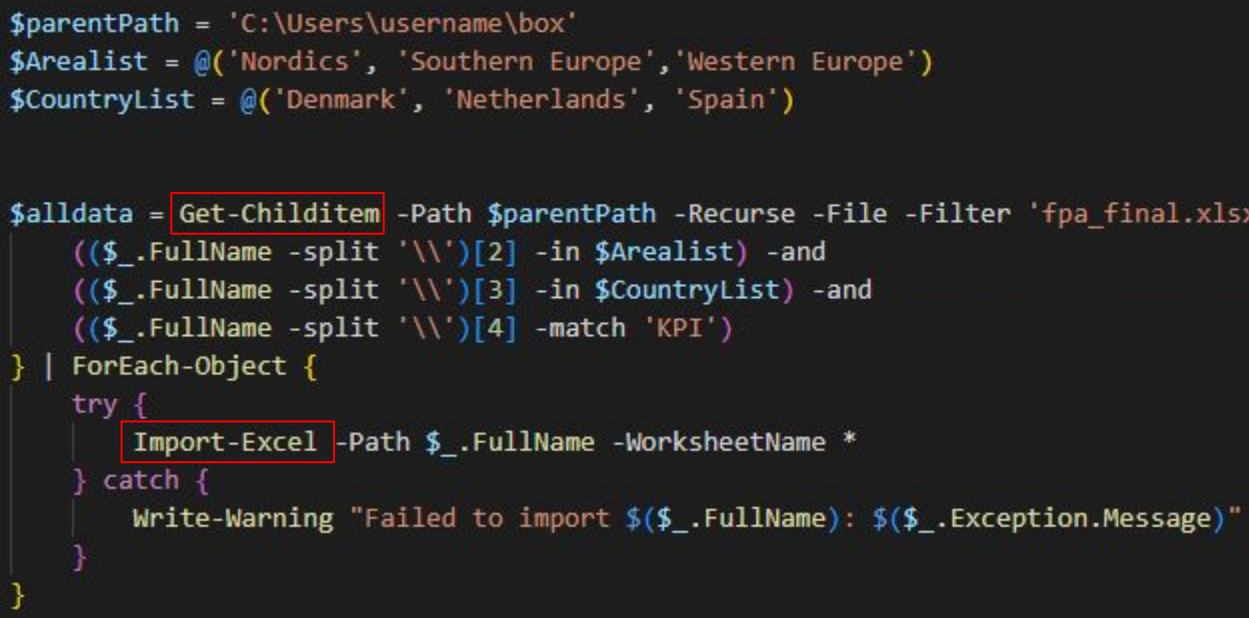
```python
def extract_df():
    global dft



    parent_path = r'C:\Users\username\box'
    area_list = ['Nordics', 'Southern Europe','Western Europe']
    country_list = ['Denmark', 'Netherlands', 'Spain']


    df=pandas.DataFrame([])

    for area in next(os.walk(parent_path))[1]:
        #print(area)
        if area in area_list:
            area_path = os.path.join(parent_path,area)
            #print(area)
            #print(area_path)
            for country in next(os.walk(area_path))[1]:
                if country in country_list:
                    country_path = os.path.join(parent_path, area, country)
                    #print(country_path)
                    for company in next(os.walk(country_path))[1]:
                        if company.endswith('_KPI'):
                            company = re.sub('_KPI', '', company)
                            company_path= os.path.join(parent_path, area, country,company)
                            year = str(datetime.date.today().year)
                            try:
                                box_path = os.path.join(parent_path, area, country,company+'_KPI',year,'fpa_final.xlsx')
                                print (box_path)

                                # Check whether the specified path exists or not
                                isExist = os.path.exists('./data')

                                if not isExist:

                                # Create a new directory because it does not exist
                                    os.makedirs('./data')

                                convert_xlsx_to_csv(box_path, 'Input', bucket_copy)
```

Python



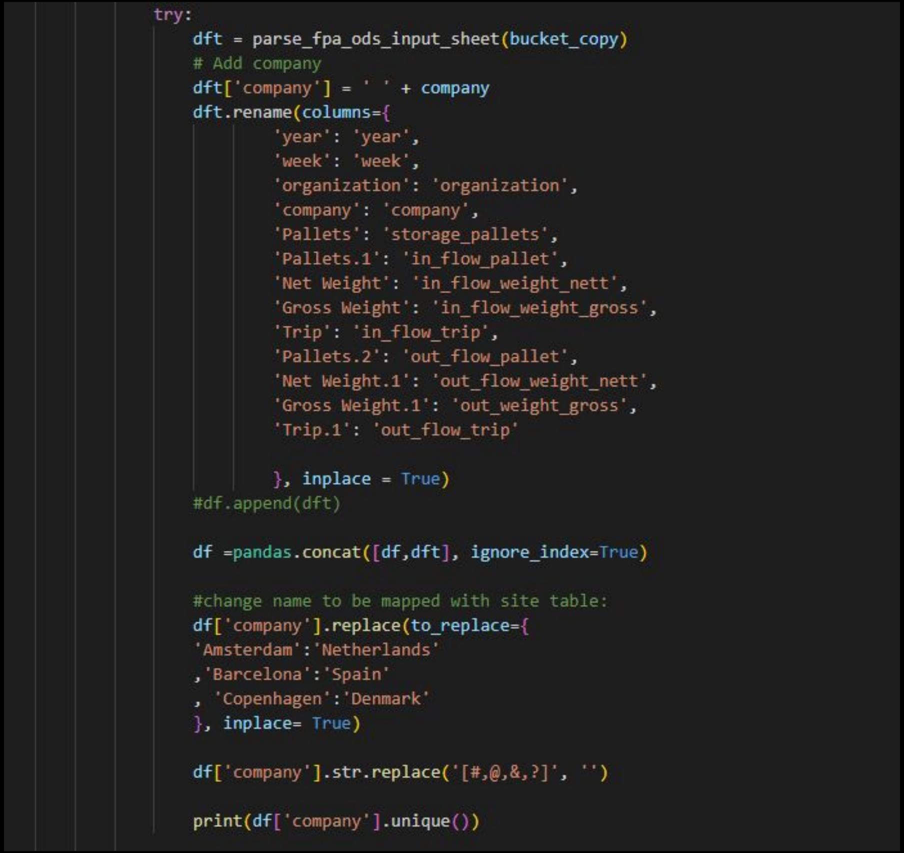
```
for country in next(os.walk(area_path))[1]:
    if country in country_list:
        country_path = os.path.join(parent_path, a
        #print(country_path)
        for company in next(os.walk(country_path))
            if company.endswith('_KPI'):
                company = re.sub('_KPI', '', compa
                company_path= os.path.join(parent_
                year = str(datetime.date.today().y
                try:
                    box_path = os.path.join(parent
                    print (box_path)
```

PowerShell



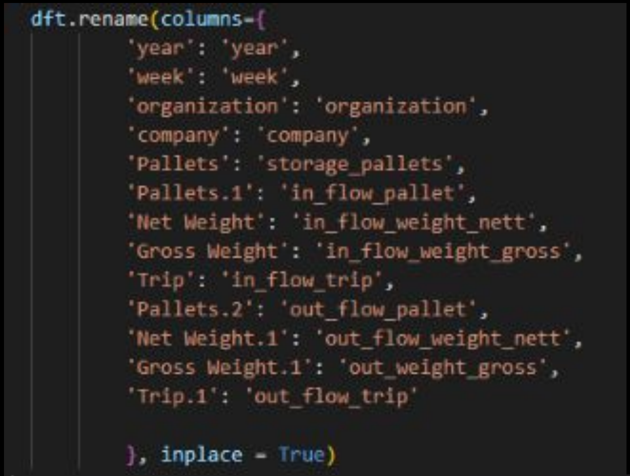
```
$parentPath = 'C:\Users\username\box'
$Arealist = @('Nordics', 'Southern Europe','Western Europe')
$CountryList = @('Denmark', 'Netherlands', 'Spain')


$alldata = Get-Childitem -Path $parentPath -Recurse -File -Filter 'fpa_final.xlsx' | Where-Object{
    (($_.FullName -split '\\')[2] -in $Arealist) -and
    (($_.FullName -split '\\')[3] -in $CountryList) -and
    (($_.FullName -split '\\')[4] -match 'KPI')
} | ForEach-Object {
    try {
        Import-Excel -Path $_.FullName -WorksheetName *
    } catch {
        Write-Warning "Failed to import $($_.FullName): $($_.Exception.Message)"
    }
}
```

Python

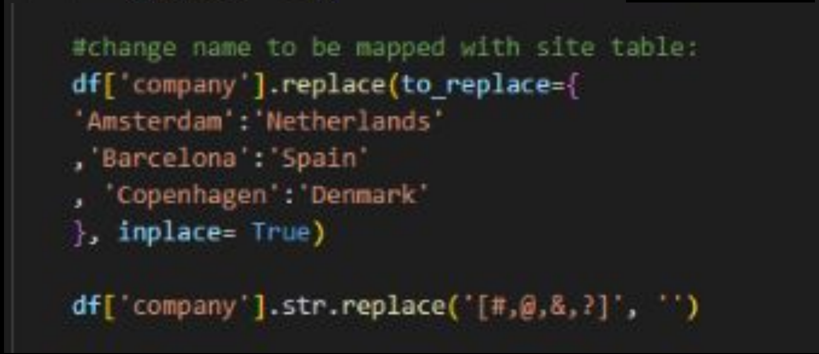


```
try:
    dft = parse_fpa_ods_input_sheet(bucket_copy)
    # Add company
    dft['company'] = ' ' + company
    dft.rename(columns={
            'year': 'year',
            'week': 'week',
            'organization': 'organization',
            'company': 'company',
            'Pallets': 'storage_pallets',
            'Pallets.1': 'in_flow_pallet',
            'Net Weight': 'in_flow_weight_nett',
            'Gross Weight': 'in_flow_weight_gross',
            'Trip': 'in_flow_trip',
            'Pallets.2': 'out_flow_pallet',
            'Net Weight.1': 'out_flow_weight_nett',
            'Gross Weight.1': 'out_weight_gross',
            'Trip.1': 'out_flow_trip'

            }, inplace = True)
    #df.append(dft)

    df =pandas.concat([df,dft], ignore_index=True)

    #change name to be mapped with site table:
    df['company'].replace(to_replace={
    'Amsterdam':'Netherlands'
    ,'Barcelona':'Spain'
    , 'Copenhagen':'Denmark'
    }, inplace= True)

    df['company'].str.replace('[#,@,&,?]', '')

    print(df['company'].unique())
```

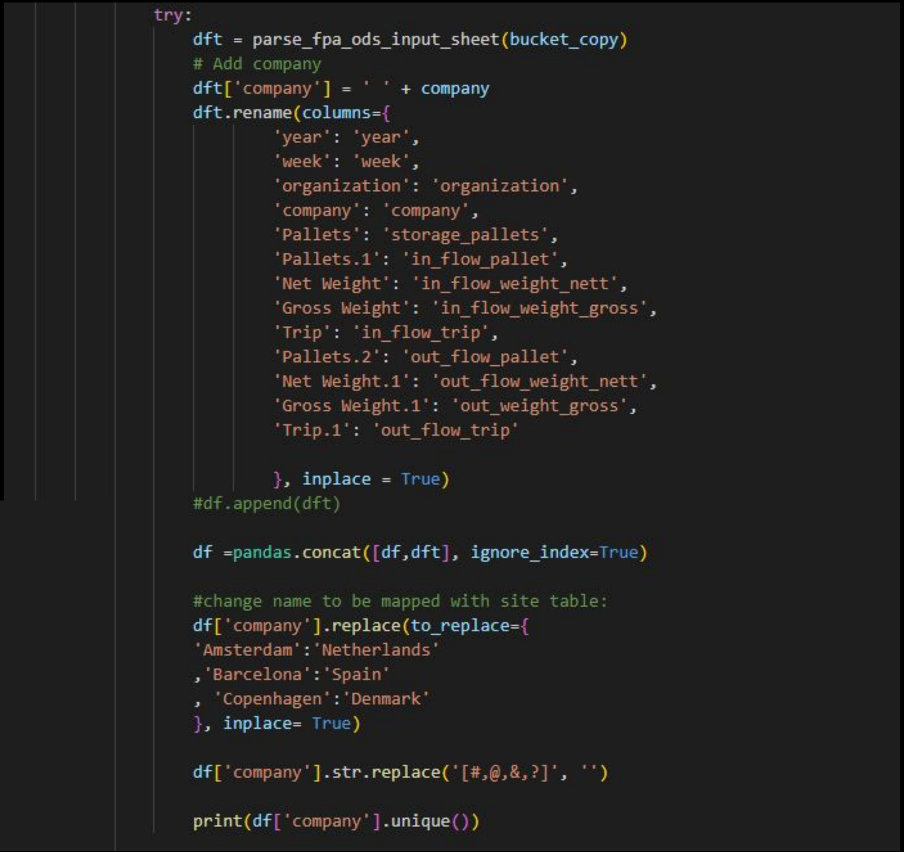
Python



```
dft.rename(columns={
          'year': 'year',
          'week': 'week',
          'organization': 'organization',
          'company': 'company',
          'Pallets': 'storage_pallets',
          'Pallets.1': 'in_flow_pallet',
          'Net Weight': 'in_flow_weight_nett',
          'Gross Weight': 'in_flow_weight_gross',
          'Trip': 'in_flow_trip',
          'Pallets.2': 'out_flow_pallet',
          'Net Weight.1': 'out_flow_weight_nett',
          'Gross Weight.1': 'out_weight_gross',
          'Trip.1': 'out_flow_trip'

          }, inplace = True)
```

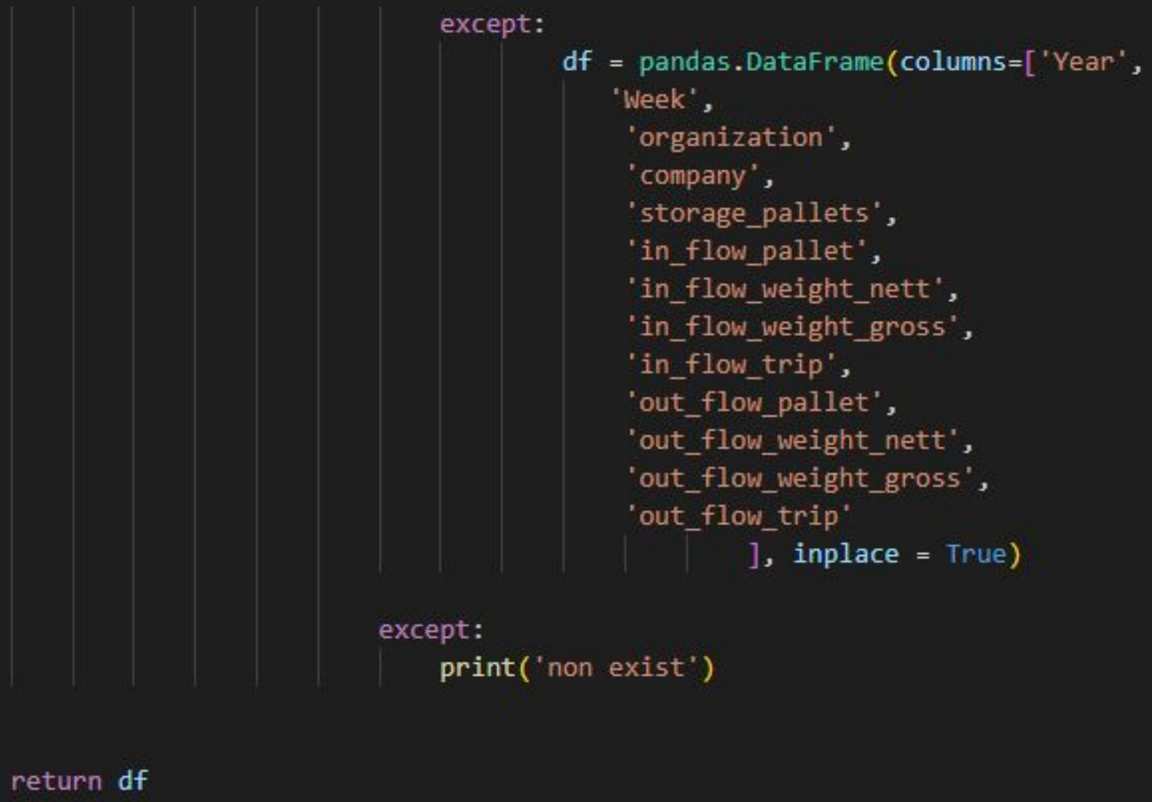


```
#change name to be mapped with site table:
df['company'].replace(to_replace={
'Amsterdam':'Netherlands'
,'Barcelona':'Spain'
, 'Copenhagen':'Denmark'
}, inplace= True)

df['company'].str.replace('[#,@,&,?]', '')
```



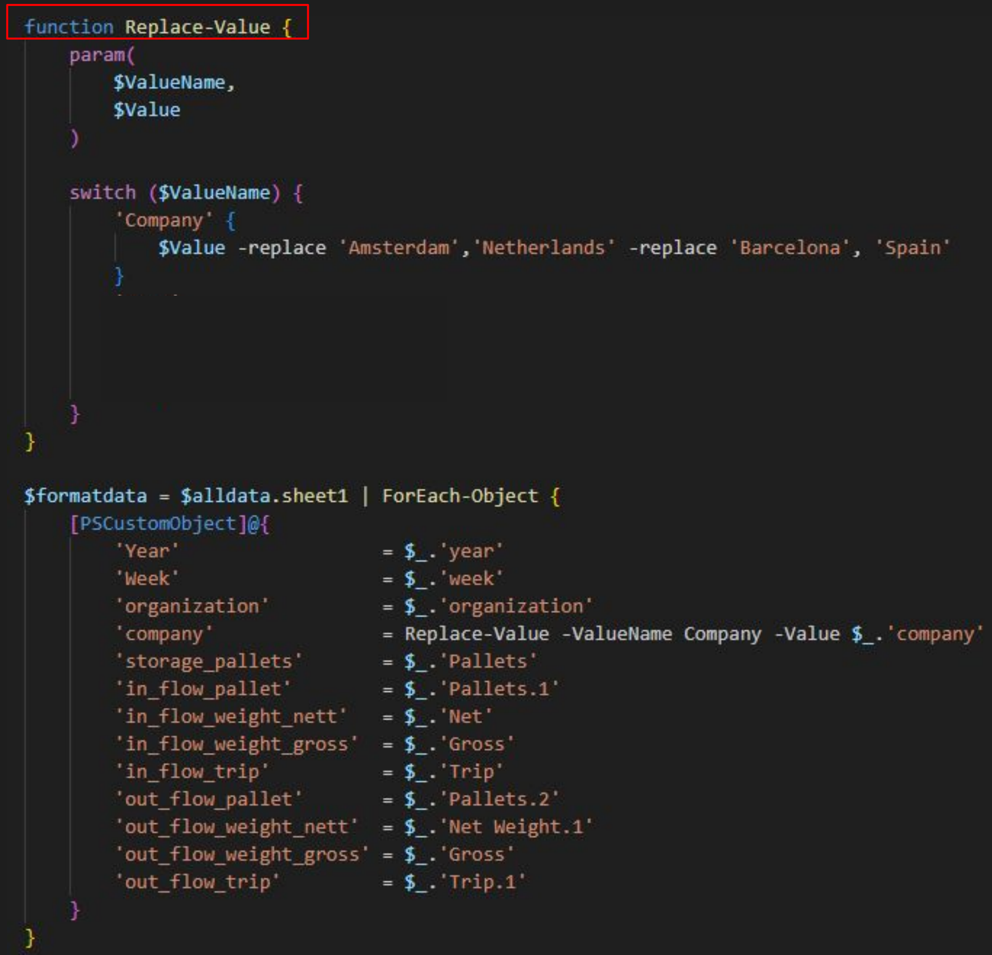
```
try:
    dft = parse_fpa_ods_input_sheet(bucket_copy)
    # Add company
    dft['company'] = ' ' + company
    dft.rename(columns={
            'year': 'year',
            'week': 'week',
            'organization': 'organization',
            'company': 'company',
            'Pallets': 'storage_pallets',
            'Pallets.1': 'in_flow_pallet',
            'Net Weight': 'in_flow_weight_nett',
            'Gross Weight': 'in_flow_weight_gross',
            'Trip': 'in_flow_trip',
            'Pallets.2': 'out_flow_pallet',
            'Net Weight.1': 'out_flow_weight_nett',
            'Gross Weight.1': 'out_weight_gross',
            'Trip.1': 'out_flow_trip'

            }, inplace = True)
    #df.append(dft)

    df =pandas.concat([df,dft], ignore_index=True)

    #change name to be mapped with site table:
    df['company'].replace(to_replace={
    'Amsterdam':'Netherlands'
    ,'Barcelona':'Spain'
    , 'Copenhagen':'Denmark'
    }, inplace= True)

    df['company'].str.replace('[#,@,&,?]', '')

    print(df['company'].unique())
```

Python



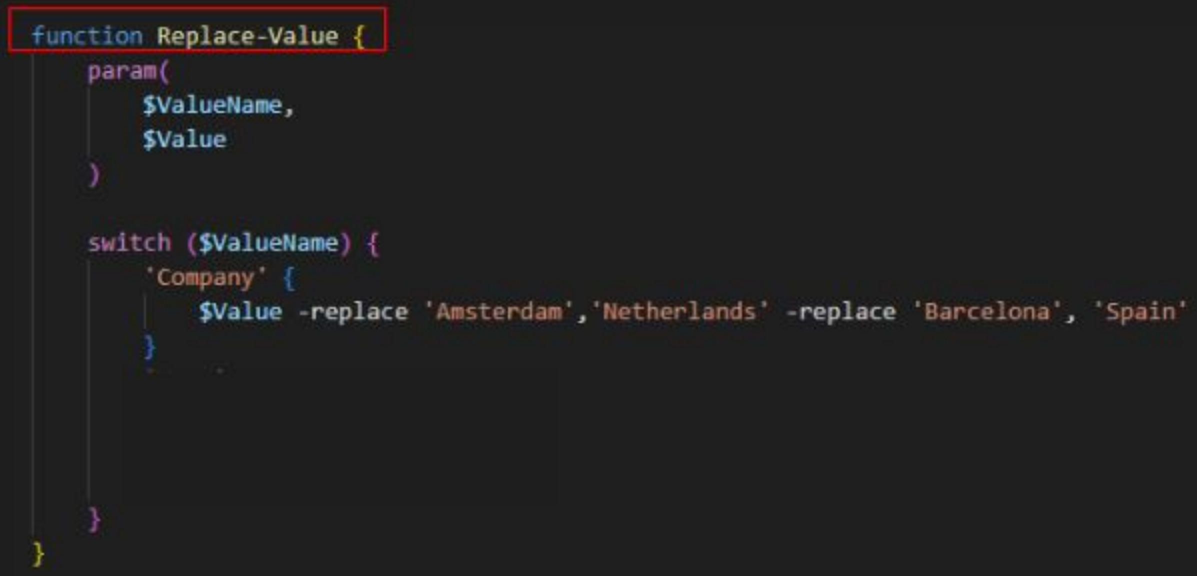
```
                            except:
                                    df = pandas.DataFrame(columns=['Year',
                                       'Week',
                                        'organization',
                                        'company',
                                        'storage_pallets',
                                        'in_flow_pallet',
                                        'in_flow_weight_nett',
                                        'in_flow_weight_gross',
                                        'in_flow_trip',
                                        'out_flow_pallet',
                                        'out_flow_weight_nett',
                                        'out_flow_weight_gross',
                                        'out_flow_trip'
                                                ], inplace = True)

                        except:
                            print('non exist')


return df
```

PowerShell



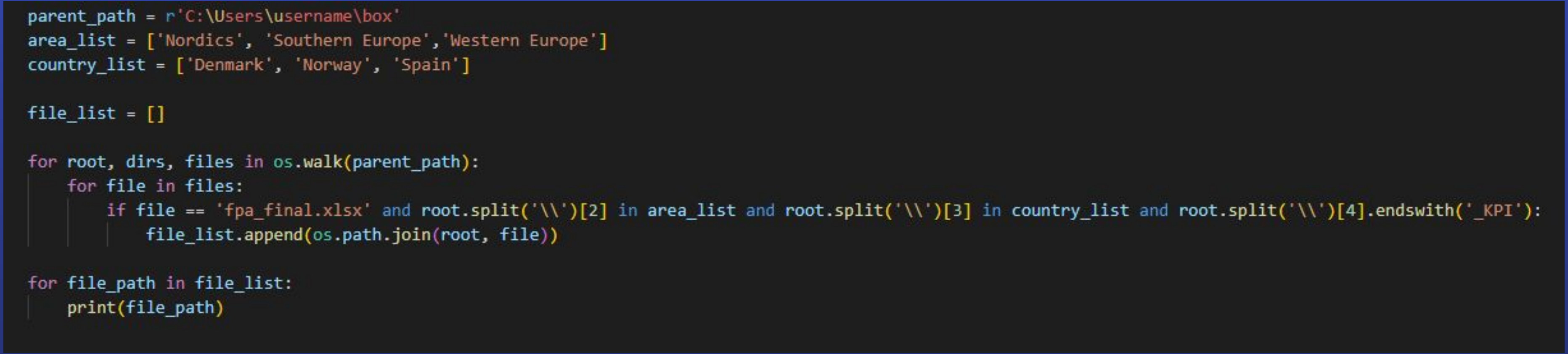
```
function Replace-Value {
    param(
        $ValueName,
        $Value
    )

    switch ($ValueName) {
        'Company' {
            $Value -replace 'Amsterdam','Netherlands' -replace 'Barcelona', 'Spain'
        }





    }
}

$formatdata = $alldata.sheet1 | ForEach-Object {
    [PSCustomObject]@{
        'Year'                  = $_.'year'
        'Week'                  = $_.'week'
        'organization'          = $_.'organization'
        'company'               = Replace-Value -ValueName Company -Value $_.'company'
        'storage_pallets'       = $_.'Pallets'
        'in_flow_pallet'        = $_.'Pallets.1'
        'in_flow_weight_nett'   = $_.'Net'
        'in_flow_weight_gross'  = $_.'Gross'
        'in_flow_trip'          = $_.'Trip'
        'out_flow_pallet'       = $_.'Pallets.2'
        'out_flow_weight_nett'  = $_.'Net Weight.1'
        'out_flow_weight_gross' = $_.'Gross'
        'out_flow_trip'         = $_.'Trip.1'
    }
}
```

PowerShell



```
function Replace-Value {
    param(
        $ValueName,
        $Value
    )

    switch ($ValueName) {
        'Company' {
            $Value -replace 'Amsterdam','Netherlands' -replace 'Barcelona', 'Spain'
        }





    }
}
```

Python



```python
parent_path = r'C:\Users\username\box'
area_list = ['Nordics', 'Southern Europe','Western Europe']
country_list = ['Denmark', 'Norway', 'Spain']

file_list = []

for root, dirs, files in os.walk(parent_path):
    for file in files:
        if file == 'fpa_final.xlsx' and root.split('\\')[2] in area_list and root.split('\\')[3] in country_list and root.split('\\')[4].endswith('_KPI'):
            file_list.append(os.path.join(root, file))

for file_path in file_list:
    print(file_path)
```

Conclusion

Limitation & Potential Enhancement

- Use DropBox API
    - Power Automate
- Integrate further with Azure DevOps
- Unit Testing

Summary

- Assess the situation to have better solution, not always the best technology is suitable, and rather make use of the current technology
- Frameworks assessment
- PowerShell can provide **intuitive** way of managing the workflow and quick learning

Thank You

Appreciate your feedback