PySpark Development Made Simple

Using VS Code, Jupyter Notebooks, and Docker

Jason Clarke · Follow

Published in Better Programming
4 min read · Sep 30, 2022

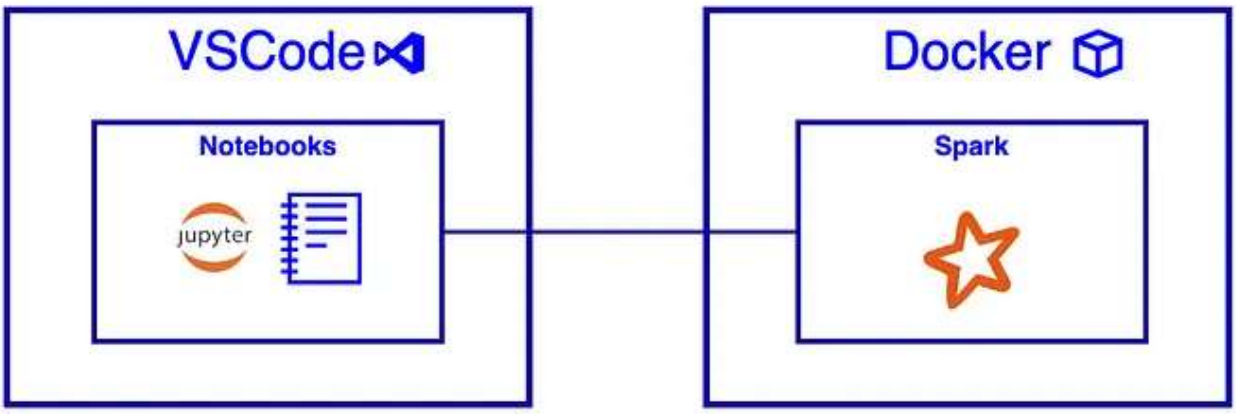


image by author

A few weeks back, I was searching for that holy grail of a tutorial describing how to use VS Code with Jupyter Notebooks and PySpark… on a Mac. And surprisingly, I couldn't find any. Well, none that passed my "explain-it-like-I'm-five" litmus test.

This article is the result of an agonising Saturday afternoon.

The Path of Least Resistance: REPLs to the Rescue

These days I have very little, if any, free time for playing around with new tech. When I do, I want it to be as painless as possible. And most importantly, I want it to be fun — otherwise, why bother?

Moreover, nothing is worse than wasting hours of your free time configuring a development environment. It's just painful.

VS Code with Jupyter Notebooks

I'm a big fan of REPLs for rapid development — for example, evaluating a new framework, analysing data, data fixes, etc.

In these situations, I don't want to configure a new project and get bogged down with trivial set-up complexities. I simply need a scratchpad to thrash out some code.

Jupyter Notebooks are a REPL-based system designed to analyse, visualise, and collaborate on data. They are also great as a scratchpad.

What is a REPL?

> *A read–eval–print loop (REPL), also termed an interactive top level or language shell, is a simple interactive computer programming environment that takes single user inputs, executes them, and returns the result to the user; a program written in a REPL environment is executed piecewise.*
> *Wikipedia*

Visual Studio code has native support for Notebooks, including Jupyter.

Setup

Prerequisites

- Install Docker
  If you're using a Mac and cannot install Docker Desktop due to licensing restrictions, check out Colima.

- Install VS Code

VS Code Development Container

1. Create a new directory for your project.

2. Create a Docker file within the root of the project directory using the code below. At the time of writing this, the current PySpark version is 3.3.0. I would check here to ensure you're using the latest version.

```
ARG IMAGE_VARIANT=slim-buster
ARG OPENJDK_VERSION=8
ARG PYTHON_VERSION=3.9.8

FROM python:${PYTHON_VERSION}-${IMAGE_VARIANT} AS py3
FROM openjdk:${OPENJDK_VERSION}-${IMAGE_VARIANT}

COPY --from=py3 / /

ARG PYSPARK_VERSION=3.3.0

RUN pip --no-cache-dir install pyspark==${PYSPARK_VERSION}
RUN pip --no-cache-dir install pandas
RUN pip --no-cache-dir install ipykernel

ENTRYPOINT ["bash"]
```

Dockerfile hosted with ❤ by GitHub view raw

3. Create a directory with the name `.devcontainer`.

4. Within the `.devcontainer` directory, add the following JSON configuration.

```
{
    "name": "Dockerfile",
    "context": "../",
    "dockerFile": "../Dockerfile",
    "extensions": ["ms-python.python", "ms-toolsai.jupyter"],
    "settings": {
        "terminal.integrated.shell.linux": null
    },
    "forwardPorts": [4050]
}
```

devcontainer.json hosted with ❤ by GitHub view raw

5. On the bottom left corner of VS Code, click the Open Remote Window button → Open In Container.

Click here to learn more about remote development within VS Code.

VS Code will restart the IDE and connect to the VS Code development container — instantiated from the Docker image defined in step 2.

That's it for the setup.

Developing Your First PySpark Application

Creating a notebook

1. Create a new file within your project directory with the extension .ipynb.
2. Open the file — you should see the VS Code notebook experience.

Test data

1. Within the root directory, add a new folder called data.
2. Within the data directory, create a new CSV file called `users.csv` and add the data below:

```
name,age,gender
jon,45,male
sarah,32,female
jane,65,female
jim,70,male
joe,22,male
```

users.txt hosted with ❤ by GitHub view raw

Example: Spark application

This section assumes you've installed Docker, configured a VS Code development container, and created an empty notebook.

```
In [ ]:
from pyspark.sql import *
import pandas as pd
```

```
In [ ]:
spark = SparkSession\
            .builder\
            .appName("test-app")\
            .getOrCreate()
```

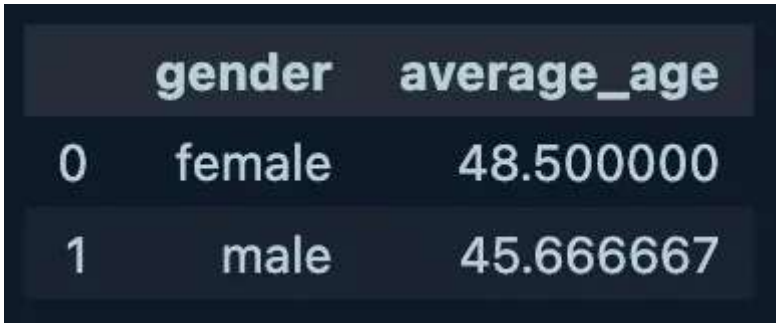
```
In [ ]:
df = spark.read.csv("./data/users.csv",
                            header="true",
                            inferSchema="true"

df.createOrReplaceTempView("users")
```

```
In [ ]:
sql = """
SELECT gender, AVG(age) as average_age
FROM users
GROUP BY gender
"""
query = spark.sql(sql)
query.toPandas()
```

pyspark-users.ipynb hosted with ❤ by GitHub view raw

OK, let's break this down cell by cell.

1. Import Libraries: The first cell imports the PySpark and Pandas Python libraries.
2. Connection to Spark: The second cell is where we define the connection to Spark. As we're running in local mode, we don't need to worry about a connection string.
3. Reading CSV into a Temp View: In the third cell, we ingest a CSV file from the local file system into Spark — the CSV contains test data.
   The second step creates a temporary view called 'users' — this allows us to query the table using plain old SQL.
4. Query: In the last cell, we define a SQL query that will return the average age of all users by gender. The function call toPandas(), converts the Spark dataframe to Panda's dataframe — allowing us to use VS Code's dataframe rendering.



	gender	average_age
0	female	48.500000
1	male	45.666667

5. Click Run All at the top to execute all cells within the notebook. If it works, you should see a two-row dataframe — as depicted in the image above.

Final Thoughts

Using Visual Studio code with Jupyter notebooks and Docker is a simple way to get started with PySpark.

If you have any tips for improving the development workflow outlined above, please let me know in the comments.

I hope you found this interesting.

The Yam Yam Architect.

If you enjoy reading stories like these and want to support me as a writer, consider signing up to become a Medium member. It's $5 a month, giving you unlimited access to stories on Medium. If you sign up using my link, I'll earn a small commission.

Join Medium with my referral link — yam yam architect

Read every story from yam yam architect (and thousands of other writers on Medium). Your membership fee directly...

medium.com

Programming Docker Data Science Software Development Software Engineering

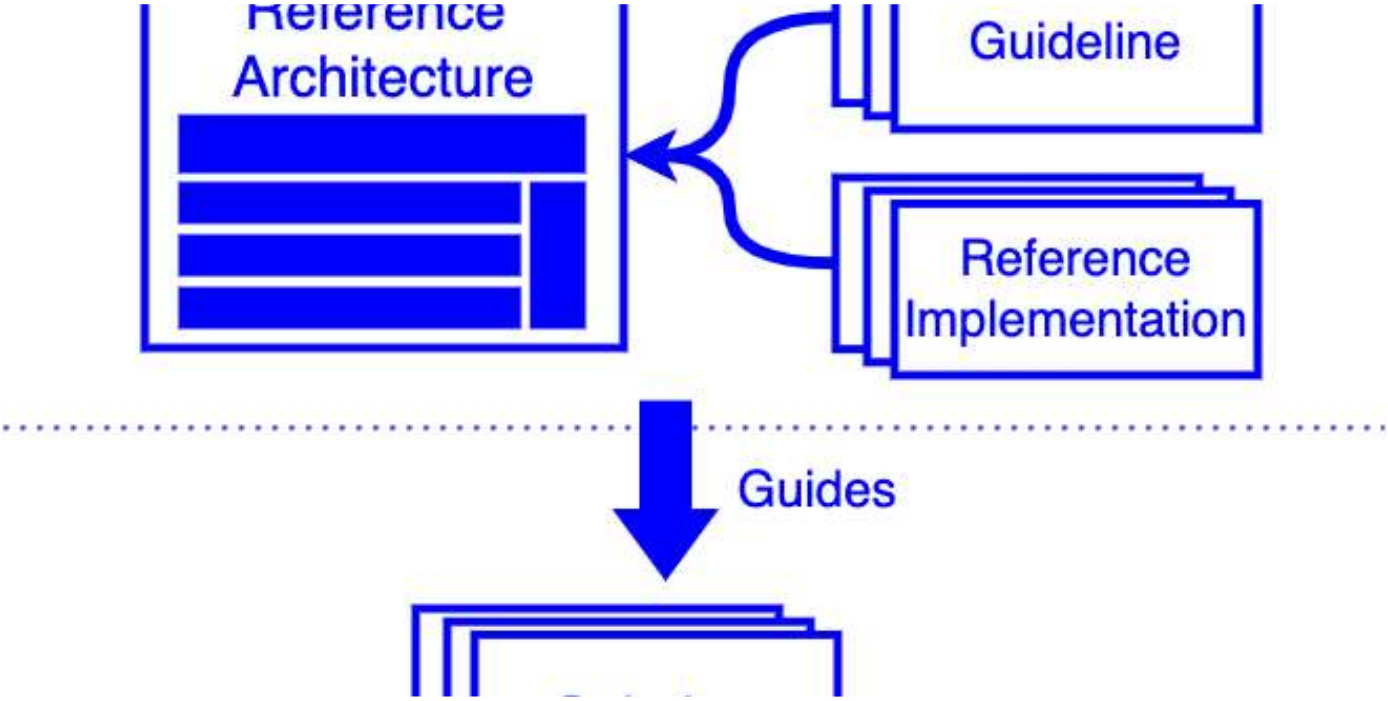
Follow

Written by Jason Clarke

870 Followers · Writer for Better Programming

Boxes and lines artist

More from Jason Clarke and Better Programming



Jason Clarke in Geek Culture

Reference Architectures

Don't reinvent the wheel—use a tried and tested approach

✨ · 6 min read · Nov 8, 2022

195 1

Timothy Mugayi in Better Programming

How To Build Your Own Custom ChatGPT With Custom Knowledge Base

Feed your ChatGPT bot with custom data sources

✨ · 11 min read · Apr 7

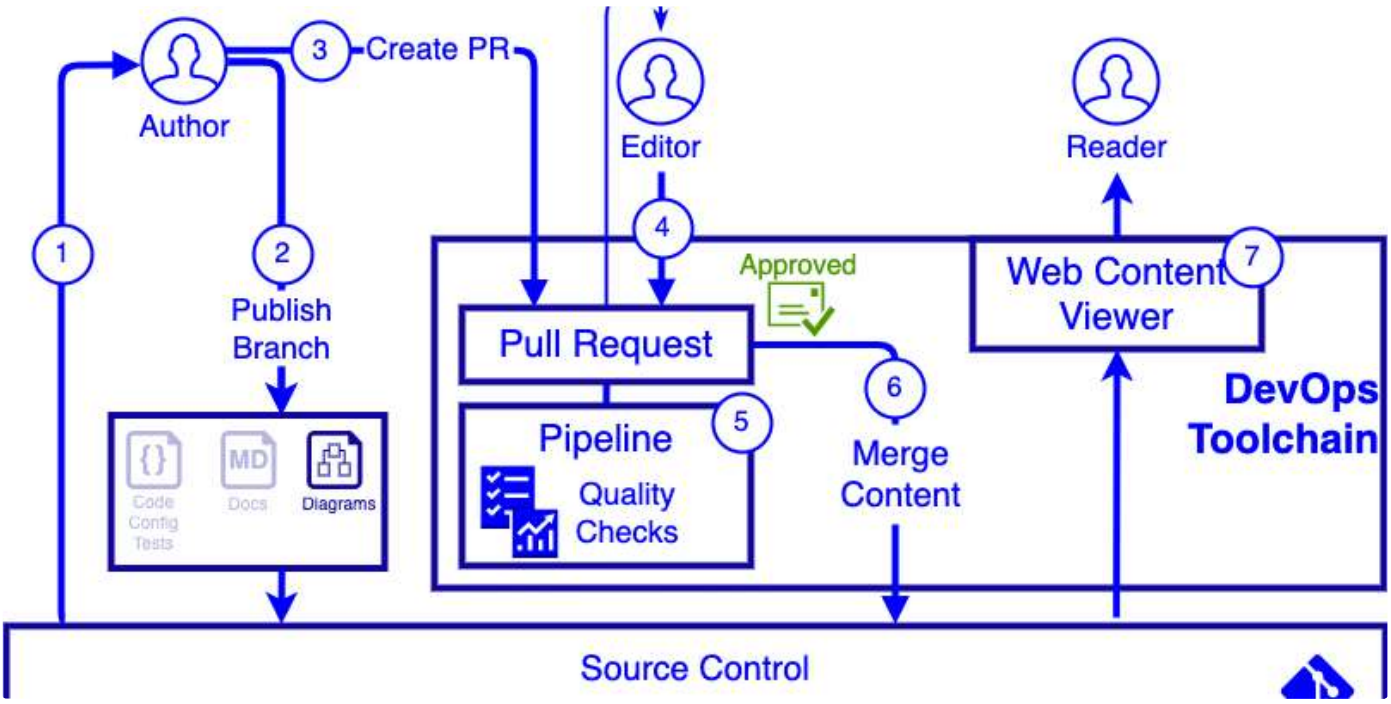
3.9K 91

Vinita in Better Programming

How to Build Credibility at Work

Building credibility requires more than just competence and knowledge.

· 8 min read · Apr 3



Jason Clarke in Geek Culture

Streamlining Diagram Creation With Draw.io and GIT

And how to manage controlled artefacts

✨ · 3 min read · Dec 24, 2022

111 1

See all from Jason Clarke

See all from Better Programming

Recommended from Medium

Luís Oliveira in Level Up Coding

How to Run Spark With Docker

Tutorial with Pyspak

✦ · 6 min read · Dec 27, 2022

187 1

Edwin Tan in Towards Data Science

How to Test PySpark ETL Data Pipeline

Validate big data pipeline with Great Expectations

✦ · 6 min read · Dec 6, 2022

113

Julian West in Towards Data Science

Unit testing PySpark code using Pytest

When it comes to writing unit-tests for PySpark pipelines, writing focussed, fast, isolated and concise tests can be a challenge.

 · 10 min read · Jan 16

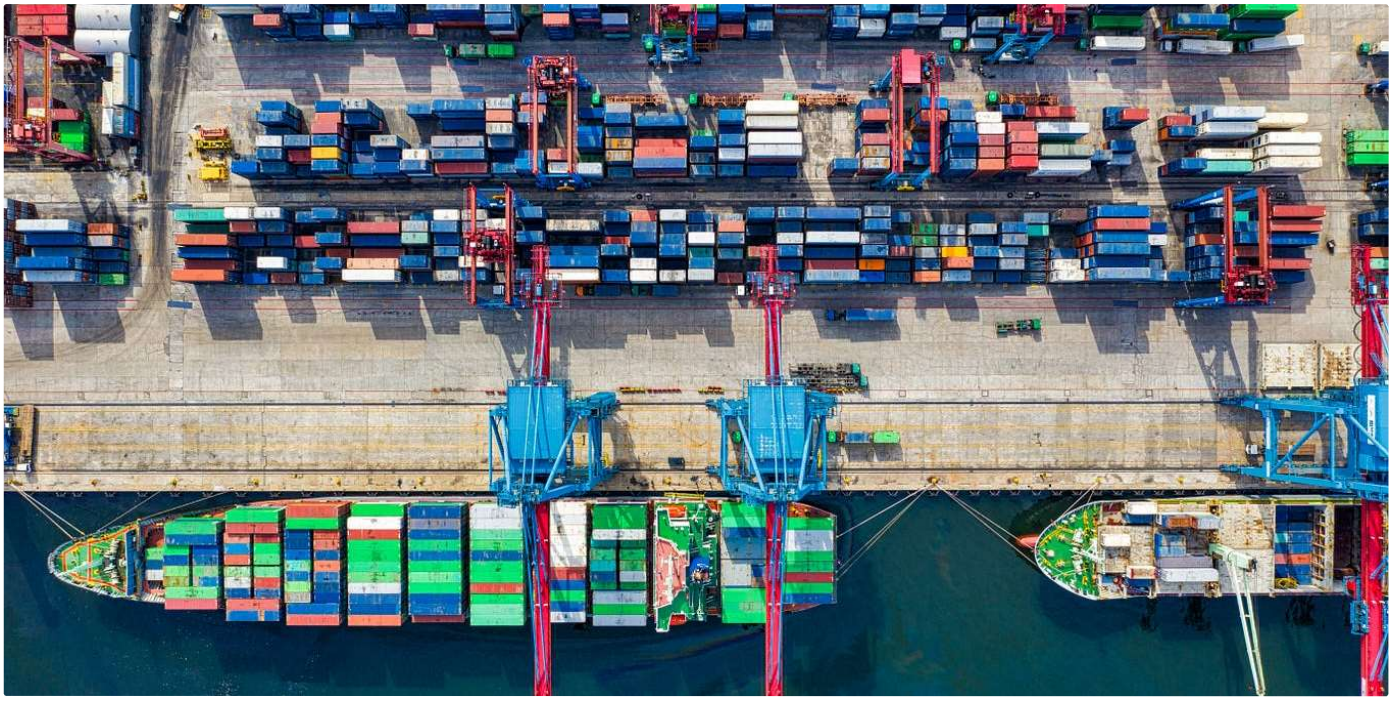
 115  1 

Park Sehun

All I can do with withColumn in Spark

If you are a spark engineer, you will know to use withColumn usefully and frequently. The `withColumn()` function in Spark is a powerful...

· 3 min read · Apr 22



Antonello Benedetto in Towards Data Science

Dockerizing Apache Zeppelin and Apache Spark for Easy Deployment

Learn How To Build a Portable and Scalable Data Analysis Environment with Docker-Compose And Volumes.

· 9 min read · Jan 24

278 4