**Document 1:** Creating a Docker Image for the Integrated Python Course

This document outlines the steps to create a Docker image with Windows as the base OS, including all necessary software and dependencies for the course.

**Prerequisites**

* Docker Desktop installed on a Windows host machine (download from [docker.com](https://www.docker.com/products/docker-desktop)).
* Windows 10/11 Pro, Enterprise, or Education edition (required for Windows containers).
* Administrative privileges to build and run Docker images.

**Software and Dependencies**

* **Python 3.11**: Core language for the course.
* **VS Code IDE**: Lightweight code editor with extensions.
* **PyCharm IDE**: Professional Python IDE for development.
* **Git**: Version control system.
* **boto3**: AWS SDK for Python (file operations with S3).
* **azure-storage-blob**: Azure SDK for Python (file operations with Azure).
* **pandas, numpy**: Data analytics and manipulation.
* **matplotlib, seaborn, plotly**: Data visualization.
* **scikit-learn**: Machine learning library.
* **torch (PyTorch)**: Deep learning framework.
* **fastapi, uvicorn**: REST API development.
* **django, flask**: Web development frameworks.
* **requests**: For API and AI integration.
* **scipy, statsmodels**: Statistical analysis.
* **beautifulsoup4, requests-html**: Web scraping.
* **rpy2** (optional): R integration for statistical analysis.
* **Additional Tools**: pip (package manager), virtualenv, shutil (file operations).

**Steps to Create the Docker Image**

1. **Install Docker Desktop**
   * Download and install Docker Desktop for Windows.
   * Enable the "Use Windows containers" option (switch to Windows containers via the Docker Desktop tray icon).
2. **Create a Dockerfile**
   * Create a new directory (e.g., integrated\_python\_course\_docker) and add a file named Dockerfile with the following content:

dockerfile

*# Use Windows Server Core as the base image*

FROM mcr.microsoft.com/windows/servercore:ltsc2019

*# Set environment variables*

ENV PYTHON\_VERSION=3.11

ENV PATH="C:\\Python311;C:\\Python311\\Scripts;%PATH%"

*# Install Chocolatey (Windows package manager)*

RUN powershell -Command "Set-ExecutionPolicy Bypass -Scope CurrentUser -Force; iwr https://chocolatey.org/install.ps1 -UseBasicParsing | iex"

*# Install Python and basic tools*

RUN powershell -Command "choco install python --version=%PYTHON\_VERSION% -y"

RUN powershell -Command "choco install git -y"

RUN powershell -Command "choco install vscode -y"

RUN powershell -Command "choco install pycharm-community -y"

*# Install Python dependencies*

RUN powershell -Command "python -m pip install --upgrade pip"

RUN powershell -Command "pip install boto3 azure-storage-blob pandas numpy matplotlib seaborn plotly scikit-learn torch fastapi uvicorn django flask requests scipy statsmodels beautifulsoup4 requests-html rpy2 virtualenv"

*# Set working directory*

WORKDIR /app

*# Copy any additional course files (e.g., sample data) if needed*

*# COPY . /app*

*# Command to keep container running*

CMD ["powershell.exe", "-Command", "Start-Sleep -Seconds 3600"]

1. **Build the Docker Image**
   * Open a terminal in the directory containing the Dockerfile.
   * Run the following command to build the image:

text

docker build -t integrated-python-course:latest .

* + This may take time due to the size of Windows containers and software installation.

1. **Test the Docker Image**
   * Run the image to verify:

text

docker run -it integrated-python-course:latest

* + Inside the container, test Python (python --version), VS Code, PyCharm, and key libraries (e.g., python -c "import pandas; print(pandas.\_\_version\_\_)").

1. **Export the Docker Image**
   * Save the image as a tar file for distribution:

text

docker save -o integrated-python-course.tar integrated-python-course:latest

* + Share the .tar file with students via a secure link or drive.

**Notes**

* The image uses Windows Server Core to keep it manageable, but it’s larger than Linux-based images (~10GB+).
* Ensure students have Docker Desktop installed with Windows container support.
* Update the PYTHON\_VERSION or package versions in the Dockerfile as needed.