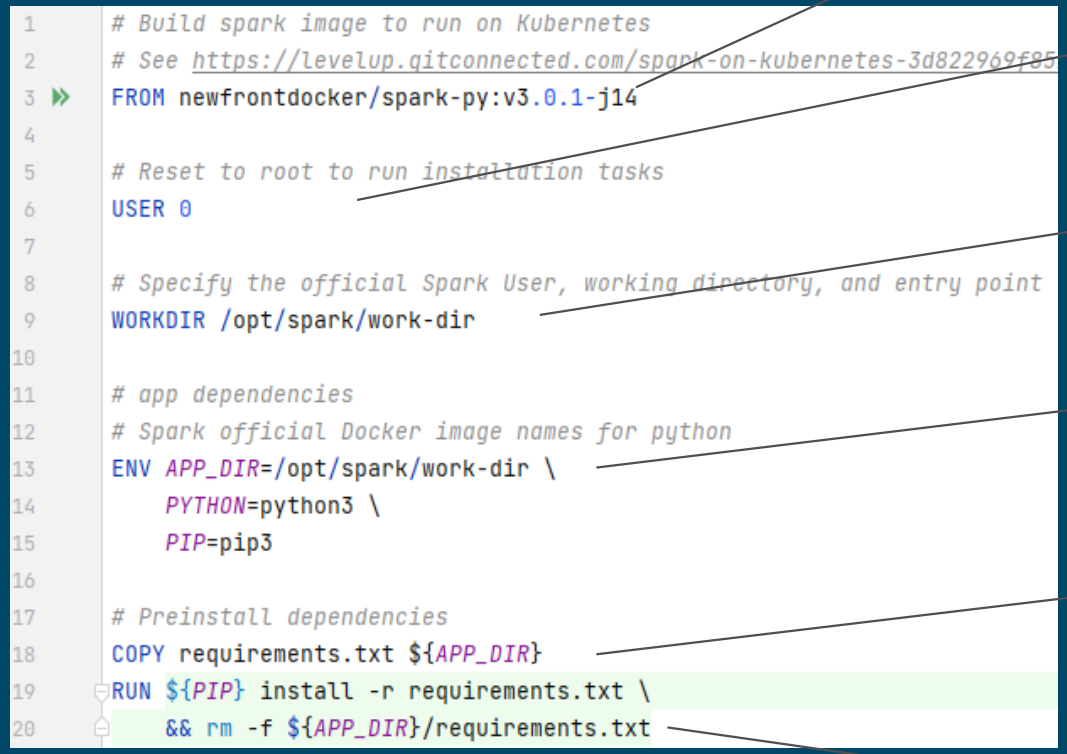
Dockerfile example

PySpark regression program as module

```dockerfile
# Build spark image to run on Kubernetes
# See https://levelup.gitconnected.com/spark-on-kubernetes-3d822969f85b
FROM newfrontdocker/spark-py:v3.0.1-j14

# Reset to root to run installation tasks
USER 0

# Specify the official Spark User, working directory, and entry point
WORKDIR /opt/spark/work-dir

# app dependencies
# Spark official Docker image names for python
ENV APP_DIR=/opt/spark/work-dir \
    PYTHON=python3 \
    PIP=pip3

# Preinstall dependencies
COPY requirements.txt ${APP_DIR}
RUN ${PIP} install -r requirements.txt \
    && rm -f ${APP_DIR}/requirements.txt

# Specify the User that the actual main process will run as
ARG spark_uid=185
# Need home directory to download Python module data (NLTK)
RUN useradd -d /home/spark -ms /bin/bash -u ${spark_uid} spark \
    && chown -R spark /opt/spark/work-dir
USER ${spark_uid}

# Just for the simple single node run in Python
ENV PYTHONPATH=${APP_DIR}/airbnb.zip:$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9-src.zip:$PYTHONPATH

# Package installed directly into this image
# (Note: package can be included in submit/sparkoperator instead of this appoach)
COPY dist/airbnb.zip ${APP_DIR}

# spark-submit local:///opt/spark/work-dir/run.py args...
# Install Python driver for modules (sleep for testing)
COPY run_env.py sleep.py ${APP_DIR}

# remove base entrypoint
#ENTRYPOINT []

# Simple single node run of program
CMD python3 -m airbnb
```



```
# Build spark image to run on Kubernetes
# See https://levelup.gitconnected.com/spark-on-kubernetes-3d822969f85
FROM newfrontdocker/spark-py:v3.0.1-j14

# Reset to root to run installation tasks
USER 0

# Specify the official Spark User, working directory, and entry point
WORKDIR /opt/spark/work-dir

# app dependencies
# Spark official Docker image names for python
ENV APP_DIR=/opt/spark/work-dir \
    PYTHON=python3 \
    PIP=pip3

# Preinstall dependencies
COPY requirements.txt ${APP_DIR}
RUN ${PIP} install -r requirements.txt \
    && rm -f ${APP_DIR}/requirements.txt
```

Begin with image containing a Spark implementation

During our image creation, work as root (0)

Docker working directory

Add some environment variables (for our runtime)

Copy local files to our image

Run the pip install command, to add modules to the image, which are required for our app

```
22  # Specify the User that the actual main process will run as
23  ARG spark_uid=185
24  # Need home directory to download Python module data (NLTK)
25  RUN useradd -d /home/spark -ms /bin/bash -u ${spark_uid} spark \
26      && chown -R spark /opt/spark/work-dir
27  USER ${spark_uid}
28
29  # Just for the simple single node run in Python
30  ENV PYTHONPATH=${APP_DIR}/airbnb.zip:$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9-src.zip:$PYTHONPATH
31
32  # Package installed directly into this image
33  # (Note: package can be included in submit/sparkoperator instead of this appoach)
34  COPY dist/airbnb.zip ${APP_DIR}
35
36  # spark-submit local:///opt/spark/work-dir/run.py args...
37  # Install Python driver for modules (sleep for testing)
38  COPY run_env.py sleep.py ${APP_DIR}
39
40  # remove base entrypoint
41  #ENTRYPOINT []
42
43  # Simple single node run of program
44  CMD python3 -m airbnb
```

Prep to run as the default ID for a spark process

Add the spark user
Spark user owns the working directory

Run as spark user

Set PYTHONPATH
It refers to our zip file (of our Python module)

Copy our Python module from local into the image

Copy our Python scripts, which help to run our program

Default way to run our program
(unless overridden to call one of our scripts)

Changes for another module: replace airbnb with your new name

- For example, airbnb becomes mynewmod

```
22  # Specify the User that the actual main process will run as
23  ARG spark_uid=185
24  # Need home directory to download Python module data (NLTK)
25  RUN useradd -d /home/spark -ms /bin/bash -u ${spark_uid} spark \
26      && chown -R spark /opt/spark/work-dir
27  USER ${spark_uid}
28
29  # Just for the simple single node run in Python
30  ENV PYTHONPATH=${APP_DIR}/airbnb.zip:$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.9-src.zip:$PYTHONPATH
31
32  # Package installed directly into this image
33  # (Note: package can be included in submit/sparkoperator instead of this appoach)
34  COPY dist/airbnb.zip ${APP_DIR}
35
36  # spark-submit local:///opt/spark/work-dir/run.py args...
37  # Install Python driver for modules (sleep for testing)
38  COPY run_env.py sleep.py ${APP_DIR}
39
40  # remove base entrypoint
41  #ENTRYPOINT []
42
43  # Simple single node run of program
44  CMD python3 -m airbnb
```

Replace airbnb

Replace airbnb

Replace airbnb

Important to remember

- Dockerfile is a script that
  - Begins with an existing image
  - Copies local files into the image
  - Installs software into the image
  - Prepares the image to run
    - E.g., environment variables, scripts