



```
FROM continuumio/miniconda3:4.9.2
```

```
ARG port=8888
```

```
ENV NOTEBOOK_PORT $port
```

```
RUN conda config --set auto_update_conda false \  
&& conda config --set notify_outdated_conda false \  
&& conda config --prepend channels conda-forge \  
&& conda config --set channel_priority strict \  
&& conda install -Sy \  
    python==3.8.5 \  
    pip==20.2.4 \  
    notebook=6.1.4 \  
    ipywidgets=7.5.1 \  
    jupyter_contrib_nbextensions=0.5.1 \  
    tini=0.18.0 \  
    numpy=1.19.1 \  
    pandas=1.1.2 \  
    matplotlib=3.2.2 \  
    seaborn=0.11.0 \  
&& conda clean -afy
```

```
COPY jupyter_notebook_config.py /root/.jupyter/
```

```
WORKDIR "/mnt"
```

```
ENTRYPOINT ["tini", "-g", "--"]
```

```
CMD ["jupyter", "notebook"]
```

Data analysis in Docker

1. Write a *Dockerfile*

2. Build an *image*

3. Run a *container*

Donkerfileinstreuties

ARG



- Defines an argument that can be passed via the command line when building the image

• The port the dupyte notebook server will listen on

• Variable persists for the remainder of the build

FROM debian:buster-slim

LABEL maintainer="Anaconda, Inc"

ENV LANG=C.UTF-8 LC_ALL=C.UTF-8

```
RUN apt-get update -q && \
    apt-get install -q -y --no-install-recommends \
        bzip2 \
        ca-certificates \
        git \
        libglib2.0-0 \
        libsm6 \
        libxext6 \
        libxrender1 \
        mercurial \
        openssh-client \
        subversion \
        wget \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*
```

ENV PATH /opt/conda/bin:\$PATH

CMD ["/bin/bash"]

ARG CONDA_VERSION=py38_4.9.2

```
RUN set -x && \
    UNAME_M="$(uname -m)" && \
    if [ "${UNAME_M}" = "x86_64" ]; then \
        MINICONDA_URL="https://repo.anaconda.com/miniconda/Miniconda3-${CONDA_VERSION}-Linux-x86_64.sh"; \
        SHA256SUM="1314b90489f154602fd794accfc90446111514a5a72fe1f71ab83e07de9504a7"; \
    elif [ "${UNAME_M}" = "s390x" ]; then \
        MINICONDA_URL="https://repo.anaconda.com/miniconda/Miniconda3-${CONDA_VERSION}-Linux-s390x.sh"; \
        SHA256SUM="4e6ace66b732170689fd2a7d86559f674f2de0a0a0fbaefd86ef597d52b89d16"; \
    elif [ "${UNAME_M}" = "aarch64" ]; then \
        MINICONDA_URL="https://repo.anaconda.com/miniconda/Miniconda3-${CONDA_VERSION}-Linux-aarch64.sh"; \
        SHA256SUM="b6fbba97d7cef35ebec8739536752cd8b8b414f88e237146b11ebf081c44618f"; \
    elif [ "${UNAME_M}" = "ppc64le" ]; then \
        MINICONDA_URL="https://repo.anaconda.com/miniconda/Miniconda3-${CONDA_VERSION}-Linux-ppc64le.sh"; \
        SHA256SUM="2b111dab4b72a34c969188aa7a91eca927a034b14a87f725fa8d295955364e71"; \
    fi && \
    wget "${MINICONDA_URL}" -O miniconda.sh -q && \
    echo "${SHA256SUM} miniconda.sh" > shasum && \
    if [ "${CONDA_VERSION}" != "latest" ]; then sha256sum --check --status shasum; fi && \
    mkdir -p /opt && \
    sh miniconda.sh -b -p /opt/conda && \
    rm miniconda.sh shasum && \
    ln -s /opt/conda/etc/profile.d/conda.sh /etc/profile.d/conda.sh && \
    echo ". /opt/conda/etc/profile.d/conda.sh" >> ~/.bashrc && \
    echo "conda activate base" >> ~/.bashrc && \
    find /opt/conda/ -follow -type f -name '*.a' -delete && \
    find /opt/conda/ -follow -type f -name '*.js.map' -delete && \
    /opt/conda/bin/conda clean -afy
```


2. Build an *image*

3. *Run a container*

1. Write a Dockerfile

• Define an argument that can be passed via the

commanding the image

ENV

