



Custom Containers in Beam

Emily Ye

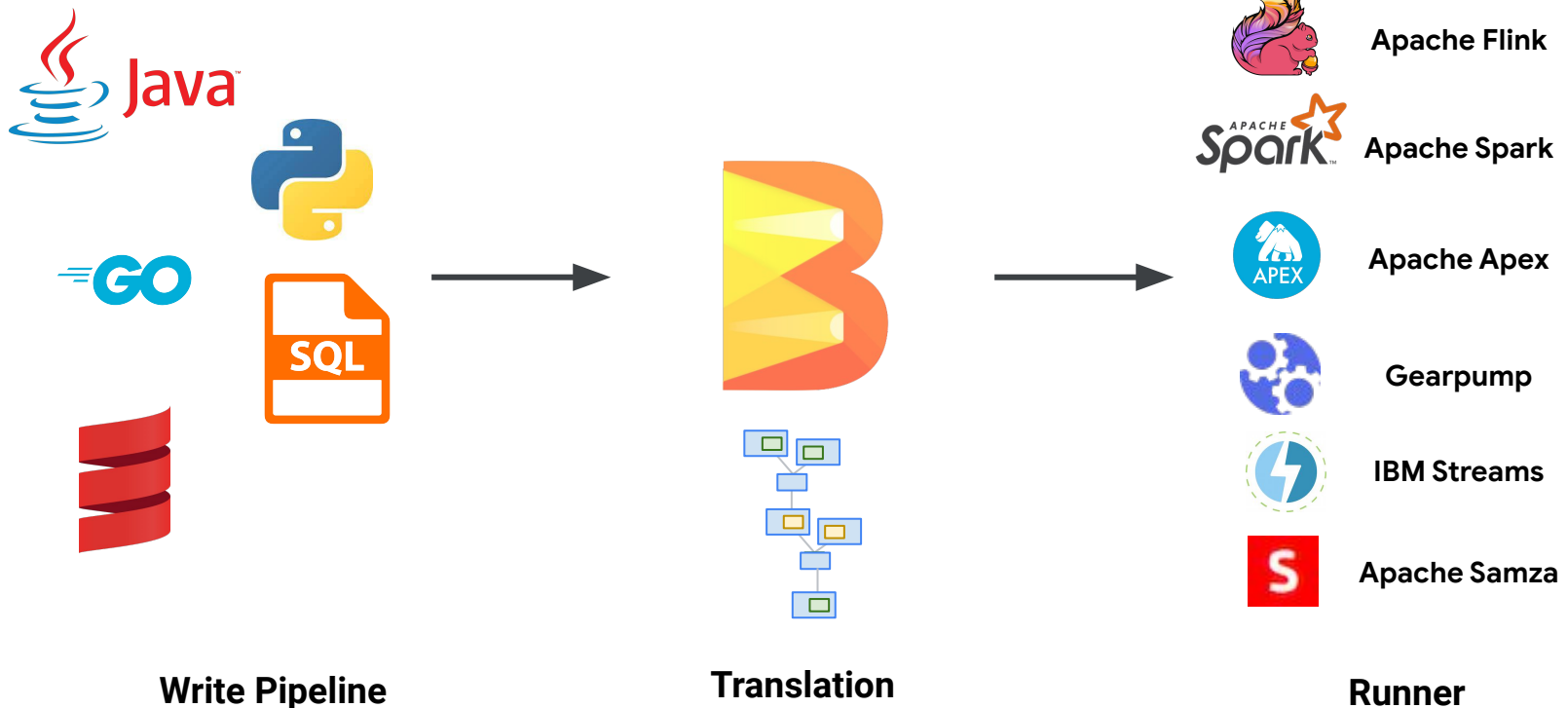
Software Engineer, Google



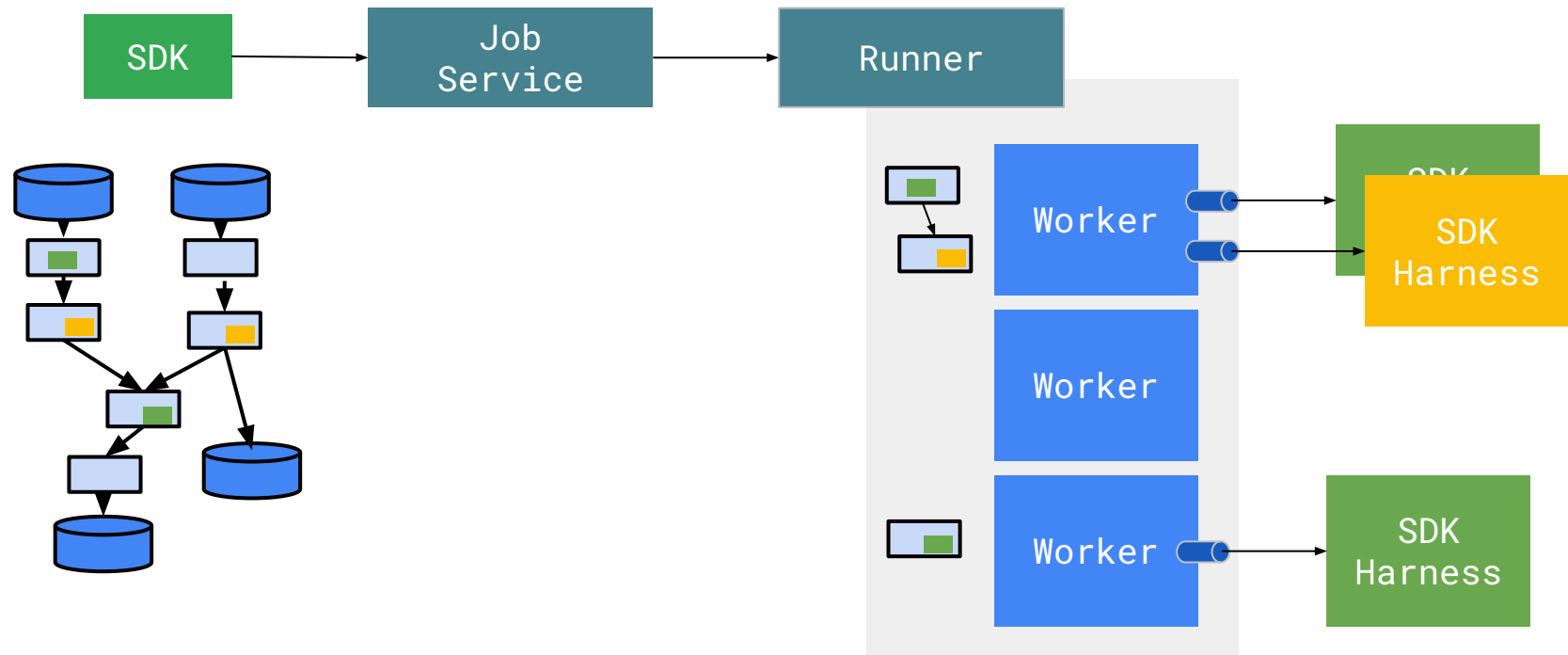
Custom Containers in Beam



Beam Portability Framework

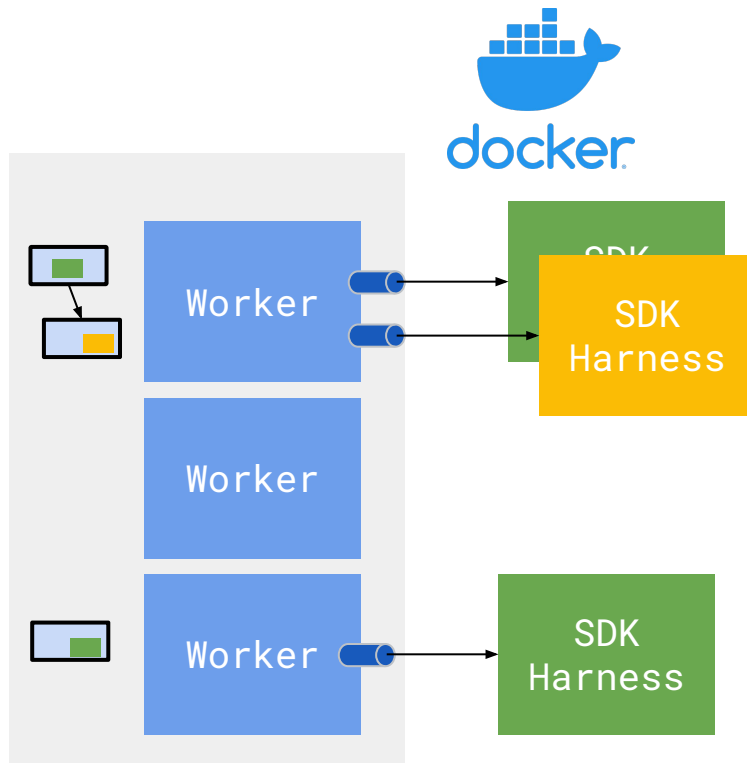


Portable Pipeline



SDK Harness

- Containerized
- Runs the user code
- Has or installs things needed to run your code
 - Language
 - SDK itself
 - Dependencies



Default SDK Harness Image

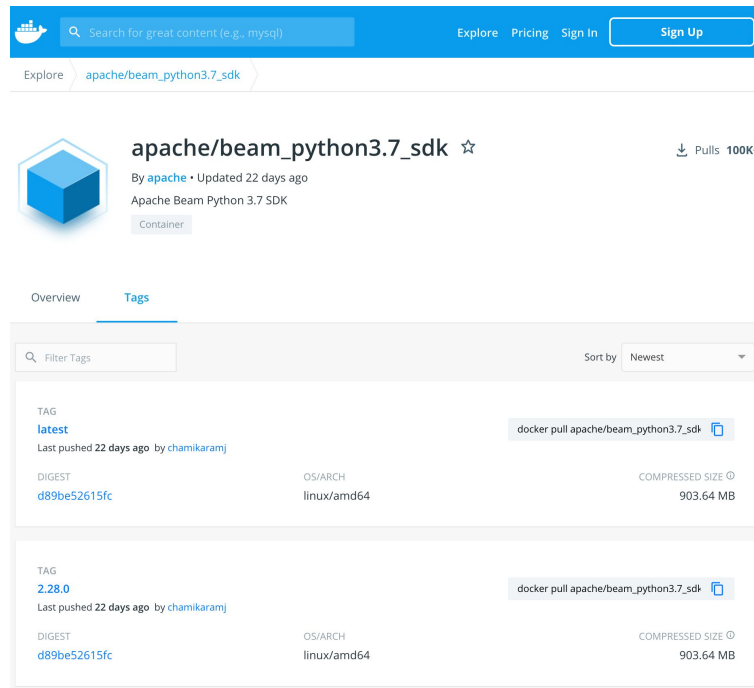
Released to [DockerHub](https://hub.docker.com/r/apache/beam_python3.7_sdk)

e.g.

https://hub.docker.com/r/apache/beam_python3.7_sdk

Also cloned to GCR

e.g. gcr.io/cloud-dataflow/v1beta3/beam_python3.7_sdk



The screenshot shows the Docker Hub interface for the `apache/beam_python3.7_sdk` image. The page header includes a search bar and navigation links like 'Explore', 'Pricing', 'Sign In', and 'Sign Up'. Below the header, the image name `apache/beam_python3.7_sdk` is displayed with a star icon and a pull count of 100K+. The description indicates it was updated 22 days ago by 'apache' and is the 'Apache Beam Python 3.7 SDK'. A 'Container' button is visible. The 'Tags' tab is selected, showing a list of tags. The first tag is `latest`, pushed 22 days ago by `chamikaramj`, with a digest of `d89be52615fc` and a compressed size of 903.64 MB. The second tag is `2.28.0`, also pushed 22 days ago by `chamikaramj`, with the same digest and size. A 'docker pull' command is provided for each tag.

TAG	OS/ARCH	COMPRESSED SIZE
<code>latest</code> Last pushed 22 days ago by <code>chamikaramj</code> DIGEST: <code>d89be52615fc</code>	linux/amd64	903.64 MB
<code>2.28.0</code> Last pushed 22 days ago by <code>chamikaramj</code> DIGEST: <code>d89be52615fc</code>	linux/amd64	903.64 MB

Custom (SDK Harness) Containers

- Build your own container image
 - Stage files/large dependencies
 - Run arbitrary setup scripts or processes
 - Environment variables
 - etc...
- Provide as a pipeline option
 - Dataflow in "preview", currently only Python portable pipelines
(`--use_runner_v2`)

Building the image



#1: Base off existing container image

```
# Dockerfile
FROM apache/beam_python37_sdk:2.27.0

COPY path/to/foo /dest/copied/foo

ENV OTHER_PROVIDER_CREDENTIALS=...
ENV LIBRARY_ENV_VAR=...

RUN apt get install ...
RUN pip install -r requirements.txt
...
```

```
$ docker build -t my_beam_python37_sdk:2.27.0-custom Dockerfile .
```

<https://docs.docker.com/engine/reference/builder/>

#1.5 - Multi-Stage Builds

```
FROM my-base-image

RUN pip install --no-cache-dir apache-beam[gcp]==2.28.0

# Copy the Apache Beam worker dependencies from the Apache Beam SDK image
COPY --from=apache/beam_python3.6_sdk:2.28.0 /opt/apache/beam /opt/apache/beam

# Set the entrypoint to Apache Beam SDK launcher.
ENTRYPOINT [ "/opt/apache/beam/boot" ]
```

#2: From source*

1. Checkout a stable branch of Beam:
2. Edit the Dockerfile.
e.g. [sdks/python/container/Dockerfile](#)
3. Build the image

```
git clone https://github.com/apache/beam.git
git fetch
git checkout release-2.27.0
...
# Edit the Dockerfile
...
./gradlew :sdks:python:container:py37:docker \
    -Pdocker-repository-root="my-repo" \
    -Pdocker-tag="2.27.0-custom"
```

Important Bits

- Language
- SDK Requirements (/opt/apache/beam)
 - SDK + required dependencies
- Boot script (/opt/apache/beam/boot)
 - Provisioning/Artifact initialization
 - Launching SDK/worker process

Usage



Prerequisites

- Runner/Workers have docker installed
- Local Runners:
 - Image loaded/loadable in docker daemon
- Remote runner/workers:
 - Pulled using docker
 - Workers may need credentials for private images
 - Cloud providers: IAM

Dataflow Runner

```
python -m apache_beam.examples.wordcount \  
  --input gs://dataflow-samples/shakespeare/kinglear.txt \  
  --output "gs://my-gcs-bucket/counts" \  
  --temp_location "gs://my-gcs-bucket/tmp/" \  
  --runner DataflowRunner \  
  --project my-gcp-project \  
  --region us-central1 \  
  --experiment=use_runner_v2 \  
  --worker_harness_container_image="my_beam_python_37_sdk:2.27.0-custom"
```

Testing Locally

```
python -m apache_beam.examples.wordcount \  
--input= /path/to/inputfile \           <-- ON CONTAINER/REMOTE  
--output /path/to/write/counts \        <-- ON CONTAINER/REMOTE  
--runner=PortableRunner \  
--job_endpoint=embed \  
--environment_type="DOCKER" \  
--environment_config="my_beam_python_37_sdk:2.27.0-custom"
```


Flink/Spark/"Portable Environment"

```
# Run a pipeline using the FlinkRunner which starts a Flink job server.
python -m apache_beam.examples.wordcount \
--input=/path/to/inputfile \           <-- ON CONTAINER/REMOTE
--output=path/to/write/counts \       <-- ON CONTAINER/REMOTE
--runner=FlinkRunner \
--environment_type="DOCKER" \
--environment_config="my_beam_python_37_sdk:2.27.0-custom"
```

Demo

<https://github.com/griscz/beam-college/tree/main/day3/custom-containers>



Keep in Mind

- Make sure:
 - SDK type and version matches (e.g. Beam 2.28.0, Python SDK)
 - Language version match (e.g. Python 3.8)
- Local runner + container behavior
 - File systems local to *container*
- "What if I don't want to install Docker?"
 - Other builders: Kaniko, Cloud Build
- DockerHub Rate limits
 - Anonymous users: 100 pulls per 6 hours

What's Next

- Add usability features for simple container builds
 - Never touch a Dockerfile or Docker
- Improvements to local container testing
- Further support/launches

Thank you!

Apache Beam: <https://beam.apache.org/community/contact-us/>

Doc Guides:

- <https://beam.apache.org/documentation/runtime/environments/>
- <https://cloud.google.com/dataflow/docs/guides/using-custom-containers>

