CI CD for Dataflow with Flex Template

Mazlum TOSUN
Head of Data and Cloud Group
GroupBees
https://twitter.com/MazlumTosun3
About me

Mazlum TOSUN

❖ Google Cloud Evangelist, Data Architect, functional programming, Devops, Serverless...
❖ Fan

https://www.youtube.com/@GCPLearning-ce9bg
https://github.com/tosun-si
https://twitter.com/MazlumTosun3
https://www.linkedin.com/in/mazlum-tosun-900b1812/
https://medium.com/@mazlum.tosun
https://stackoverflow.com/users/9261558/mazlum-tosun
Dataflow Templates Types

Flex Templates

- Standardisation of Dataflow Template deployment based on a Docker image
- Standardisation whatever the language and the sdk
- All the dependencies can be pre installed in the container
- Template spec from GCS
Advantages of Flex Templates over classic template

- Docker image offers more flexibility
- Classic templates require the ValueProvider interface for input parameters
- Flex templates don't require the ValueProvider
- Classic templates have a static job graph
- Flex templates can dynamically construct the job graph
- Example, select a different I/O connector based on input parameters
CI CD Flex Templates with Cloud Build for Java SDK

Dataflow CI CD with Flex Template and Cloud Build (with Dockerfile)

Artifact Management
- Image Storage
- Artifact Registry

Continuous Integration
- Cloud Build
- Automatic Trigger

Continuous Delivery
- Cloud Build
- Maven

Source Code Repository
- Git Source Repository

DevOps
Commit PR
Developer

flex-template command
Create spec file GCS
Build and publish image
Deploy the Flex Template spec file GCS
Run the Flex Template Dataflow job
CI CD Flex Templates with Cloud Build for Java SDK

Dataflow CI CD with Flex Template and Cloud Build (without Dockerfile)

Artifact Management

Continuous Integration

Source Code Repository

Continuous Delivery

Artifact Registry

Cloud Build

Commit PR

Developer

Launch Unit Test

maven

Java

maven

Run the Flex Template Dataflow job

Deploy the Flex Template spec file GCS

Build Fat Jar

DevOps

DevOps

Manual Triggers

Automatic Trigger

flex-template command

Generate Docker image

Create spec file GCS

DevOps

Cloud Build

Cloud Build

maven
Cloud Build

- Run unit tests with Maven
- Build image with a Dockerfile and all the dependencies in the container
- Create spec file in Cloud Storage
CI CD Flex Templates with Cloud Build for Java SDK: without Dockerfile

Cloud Build

- Run unit tests with Maven
- Build the fat jar
- Generate the image and spec file in GCS with flex-template command
CI CD Flex Templates with Cloud Build for Python SDK

- Run unit tests with PyTest
- Build image with a Dockerfile and all the dependencies
- Create spec file in Cloud Storage
CI CD Flex Templates with Gitlab CI for Java SDK

- Run unit tests with Maven
- Build image with a Dockerfile and all the dependencies using Kaniko
- Create spec file in a Cloud Storage
CI CD Flex Templates with Gitlab CI for Python SDK

- Run unit tests with PyTest
- Build image with a Dockerfile and all the dependencies using Kaniko
- Create spec in Cloud Storage bucket
CI CD Flex Templates with Dagger for Java and Python SDK

- Pipeline As Code with Go SDK
- Build image with a Dockerfile and all the dependencies using Kaniko
- Create spec in Cloud Storage bucket
Cloud Build

- Serverless
- No need a token key for authentication
- Not an interactive CI and no orchestration for the pipeline
- Natively can’t share YAML templates
CI CD Flex Templates: difference between approaches

- Graphic and interactive CI CD pipeline
- Manual and automatic jobs
- Dependencies between jobs / can share YAML templates
- Can require a token key if installed outside of Google Cloud
- If installed on GCP, need having a VM or a GKE cluster
- Can use Workload Identity Federation if installed on GKE

Gitlab CI
CI CD Flex Templates: difference between approaches

- **Dagger**
  - Pipeline As Code
  - Different SDKs: Go, Node, Python
  - More easy to add some code logics than YAML (example: error handling)
  - Depends only on Docker
  - Need to handle the authentication with ADC or a token key
  - No orchestration for the pipeline (manual and automatic steps)
Links to the projects used for this talk

https://github.com/tosun-si/dataflow-java-ci-cd

https://github.com/tosun-si/dataflow-python-ci-cd

https://medium.com/google-cloud/ci-cd-for-dataflow-java-with-flex-templates-and-cloud-build-e3c584b8e564
Thank you :)

QUESTIONS?