Robots are your friends

Keeping your operators up to date with automation

Leah Cole
Google Cloud
Agenda

- Intro to Leah + provider packages/backport packages
- Before robot friends
- With robot friends
Hi, I'm Leah!
What are the provider packages?
What are the provider packages?
Before robot friends
Process without automation

New release of providers package

Human notices update, tests locally

Tests DAGs in dev, promotes to prod
Repo Structure

dags/
    |__ example_dag.py
    |__ example_dag_test.py
    |__ requirements.txt

@leahecole
With robot friends
Step 0 - The provider packages are updated!
Step 1 - Renovate bot robot friend opens a PR
Create a requirements-composer.txt file

dags/
|  |__ example_dag.py
|  |__ example_dag_test.py
|__ requirements-composer.txt
|__ requirements.txt
Create a requirements-composer.txt file

apache-airflow-backport-providers-google==2020.11.13
Configure a robot friend

dags/
  |__ example_dag.py
  |__ example_dag_test.py
  |
  |__ requirements-composer.txt
  |__ renovate.json
  |__ requirements.txt
Configure a robot friend

{
  "extends": [
    "config:base"
  ],
  "baseBranches": ["main"],
  "masterIssue": true,
  "pip_requirements": {
    "fileMatch": ["requirements-composer.txt"]
  }
}
Renovate bot robot friend opens a PR
Step 2: Cloud Build robot friend runs presubmit tests
Create a Dockerfile

dags/
  |__ example_dag.py
  |__ example_dag_test.py
  |__ requirements-composer.txt
  |__ Dockerfile
  |__ renovate.json
  |__ requirements.txt

@leahecole
Create a Dockerfile

FROM python:3.8

# Allow statements and log messages to immediately appear in the Cloud Run logs
ENV PYTHONUNBUFFERED True

COPY requirements.txt ./
COPY requirements-composer.txt ./

RUN pip install --no-cache-dir -r requirements.txt
RUN pip install --no-cache-dir -r requirements-composer.txt

# copy dag code to container image
ENV DAGS /dags
WORKDIR $DAGS
COPY . ./
CMD ["pytest", "-s", "dags/example_dag_test.py"]
Create a cloudbuild.yaml file

dags/
    |__ example_dag.py
    |__ example_dag_test.py

    |__ requirements-composer.txt
    |__ Dockerfile
    |__ test-dags.cloudbuild.yaml
    |__ renovate.json
    |__ requirements.txt
Create a cloudbuild.yaml file

steps:
  # build the docker image
  - name: 'gcr.io/cloud-builders/docker'
    args: ['build', '-t', 'gcr.io/${PROJECT_ID}/cicd:${SHORT_SHA}', '.']  # tag docker image with commit sha
    id: 'docker build'
  # run the dag tests
  - name: 'gcr.io/cloud-builders/docker'
    args: ['run', 'gcr.io/${PROJECT_ID}/cicd:${SHORT_SHA}']
    id: 'test-dags'
Create a Cloud Build Trigger

**Name**: test-dags  
**Event**: Pull Request  
**Source** - Repository: choose your repository  
**Source** - Base branch: ^main$  
**Source** - Comment Control: not required  
**Build Configuration** - Cloud build configuration file: /test-dags.cloudbuild.yaml
Cloud Build robot friend runs presubmit tests

- All checks have passed
  1 successful check

- Test-Dags (leah-playground) Successful in 2m — Summary

- This branch has no conflicts with the base branch
  Merging can be performed automatically.

- Merge pull request
  You can also open this in GitHub Desktop or view command line instructions.
Step 3 - PR Approved (by a human) and merged to main
Step 4 - Another Cloud Build robot friend updates your dev environment
Create a cloudbuild.yaml file

dags/
 |   __ example_dag.py
 |   __ example_dag_test.py

|__ requirements-composer.txt
|__ Dockerfile
|__ test-dags.cloudbuild.yaml
|__ update-composer.cloudbuild.yaml
|__ renovate.json
|__ requirements.txt
Create a cloudbuild.yaml file

steps:
  #update the composer environment
  - name: 'gcr.io/cloud-builders/gcloud'
    args: ['composer', 'environments', 'update', '${_COMPOSER_NAME}', '--update-pypi-packages-from-file', 'requirements-composer.txt', '--location', '${_COMPOSER_REGION}']
    id: 'update-composer-env'
  timeout: 3600s #1 hour timeout accommodates the long running Composer upgrade operation
Create a Cloud Build Trigger

**Name:** update-composer-env  
**Event:** Push to a branch  
**Source** - Repository: choose your repository  
**Source** - Base branch: ^main$  
**Source** - Included files filter (glob): requirements-composer.txt  
**Build Configuration** - Cloud build configuration file:  
/update-composer.cloudbuild.yaml  

**Advanced Configuration** - Substitution variables  
_COMPOSER_NAME - the name of your composer environment  
_COMPOSER_REGION - the Compute engine region where your environment is located
Step 5 - Look at DAGs in dev environment
Step 6 - Promote your updates to your prod environment
Putting it together
Keeping DAGs Up to Date

@leahecole
What's next?

- Rollback strategy
- More system test automation
- Alternate components
  - GitHub Actions
  - Dependabot
Additional Resources

Testing Airflow workflows - Bas Harenslak - Airflow Summit 2020

Blog post summarizing the Cloud Build approach