

Global Economic Analysis on a Budget

Kevin Booth

Who am I?

Who am I?

- Graduate Student at Texas State University

Who am I?

- Graduate Student at Texas State University
- Software Developer
 - NASA Johnson Space Center (Intern)
 - Swipejobs (Backend Developer)
 - Chaotic Moon Studios (Android Developer)

Who am I?

- Graduate Student at Texas State University
- Software Developer
 - NASA Johnson Space Center (Intern)
 - Swipejobs (Backend Developer)
 - Chaotic Moon Studios (Android Developer)
- Dog Dad



Today's Topic

Building a Satellite Imagery Processing Pipeline

A History Lesson

GDP Reports are Important

- Measures the pulse of our economy

GDP Reports are Important

- Measures the pulse of our economy
 - Federal Reserve uses it to determine monetary policy
 - Investors use it to efficiently allocate resources

GDP Reports are Important

- Measures the pulse of our economy
 - Federal Reserve uses it to determine monetary policy
 - Investors use it to efficiently allocate resources
- Valuing Ecosystem Services

GDP Reports are Important

- Measures the pulse of our economy
 - Federal Reserve uses it to determine monetary policy
 - Investors use it to efficiently allocate resources
- Valuing Ecosystem Services
 - Fertile Land

GDP Reports are Important

- Measures the pulse of our economy
 - Federal Reserve uses it to determine monetary policy
 - Investors use it to efficiently allocate resources
- Valuing Ecosystem Services
 - Fertile Land
 - Mangroves

Problems with GDP Reports

Problems with GDP Reports

- Delayed by Government Shutdowns

Problems with GDP Reports

- Delayed by Government Shutdowns
- Political Incentive to Under/Over Report

Problems with GDP Reports

- Delayed by Government Shutdowns
- Political Incentive to Under/Over Report
- Reported for Administrative Units

Thesis

Thesis

- Estimating GDP using Satellite Imagery

Thesis

- Estimating GDP using Satellite Imagery
 - We previously used night-time imagery

Thesis

- Estimating GDP using Satellite Imagery
 - We previously used night-time imagery
 - Switching to SAR and Multispectral

Thesis

- Estimating GDP using Satellite Imagery
 - We previously used night-time imagery
 - Switching to SAR and Multispectral
- Landsat Imagery

Thesis

- Estimating GDP using Satellite Imagery
 - We previously used night-time imagery
 - Switching to SAR and Multispectral
- Landsat Imagery
 - Measuring effectiveness of a technique

Thesis

- Austin, TX

Thesis

- Austin, TX
- Calculate Developed Area for 2002
 - Around 50 Landsat Scenes

Thesis

- Austin, TX
- Calculate Developed Area for 2002
 - Around 50 Landsat Scenes
- Repeat for every year until 2012

Thesis

- Austin, TX
- Calculate Developed Area for 2002
 - Around 50 Landsat Scenes
- Repeat for every year until 2012
- Repeat for 199 more study areas
 - More than 100,000 scenes

Pipeline Steps

Pipeline Steps

- Download Landsat Scenes

Pipeline Steps

- Download Landsat Scenes
- Convert to ToA Reflectance

Pipeline Steps

- Download Landsat Scenes
- Convert to ToA Reflectance
- Calculate Developed Area Index

Pipeline Steps

- Download Landsat Scenes
- Convert to ToA Reflectance
- Calculate Developed Area Index
- Calculate Cloud Mask

Pipeline Steps

- Download Landsat Scenes
- Convert to ToA Reflectance
- Calculate Developed Area Index
- Calculate Cloud Mask
- Create Cloud Free Composite

Pipeline Steps

- Download Landsat Scenes
- Convert to ToA Reflectance
- Calculate Developed Area Index
- Calculate Cloud Mask
- Create Cloud Free Composite
- Calculate Total Developed Area

Software

Software

- Python
 - rasterio
 - python-fmask

Software

- Python
 - rasterio
 - python-fmask
- Kubernetes

Software

- Python
 - rasterio
 - python-fmask
- Kubernetes
- PostgreSQL

Software

- Python
 - rasterio
 - python-fmask
- Kubernetes
- PostgreSQL
- Airflow

```
1 default_args = {
2     'owner': 'kevin',
3     'depends_on_past': False,
4     'start_date': datetime.utcnow(),
5     'email': ['kevin@kb.gg'],
6     'email_on_failure': False,
7     'email_on_retry': False,
8     'retries': 1,
9     'retry_delay': timedelta(minutes=5)
10 }
11
12 secret_file = Secret('volume', '/secrets', 'thesis-db', None)
13 volume_mount = VolumeMount('research', mount_path='/data', sub_path=None, read_only=False)
14 volume_config= {'persistentVolumeClaim': {'claimName': 'research'}}
15 volume = Volume(name='research', configs=volume_config)
16
17 dag = DAG('thesis_austin_tx', default_args=default_args, concurrency=4)
18
19 for year in range(2002, 2012):
20     download_task = KubernetesPodOperator(namespace='thesis',
21                                           image="registry.kb.gg:5000/thesis:latest",
22                                           image_pull_policy="Always",
23                                           image_pull_secrets="dev-regsecret",
24                                           cmds=["download-scenes"],
25                                           arguments=["2306", str(year)],
26                                           labels={"app": "thesis"},
27                                           secrets=[secret_file],
28                                           volumes=[volume],
29                                           volume_mounts=[volume_mount],
30                                           name="download-scenes-{}".format(year),
31                                           task_id="download-scenes-{}".format(year),
32                                           dag=dag,
33                                           config_file="/data/airflow/config/config")
34     create_toa_task = KubernetesPodOperator(namespace='thesis',
35                                             image="registry.kb.gg:5000/thesis:latest",
36                                             image_pull_policy="Always",
37                                             image_pull_secrets="dev-regsecret",
38                                             cmds=["create-toa"],
39                                             arguments=["2306", str(year)],
40                                             labels={"app": "thesis"},
41                                             secrets=[secret_file],
42                                             volumes=[volume],
43                                             volume_mounts=[volume_mount],
44                                             name="create-toa-{}".format(year),
45                                             task_id="create-toa-{}".format(year),
46                                             dag=dag,
47                                             config_file="/data/airflow/config/config")
48
49     download_task.set_downstream(create_toa_task)
```


On  **DAG: thesis_austin_tx**

schedule: 0:10:00

 Graph View

 Tree View

 Task Duration

 Task Tries

 Landing Times

 Gantt

 Details

 Code

 Refresh

 Delete

success Base date: 2019-05-10 12:45:38

Number of runs: 25 ▾

Run: manual__2019-05-10T12:45:37.446274+00:00 ▾

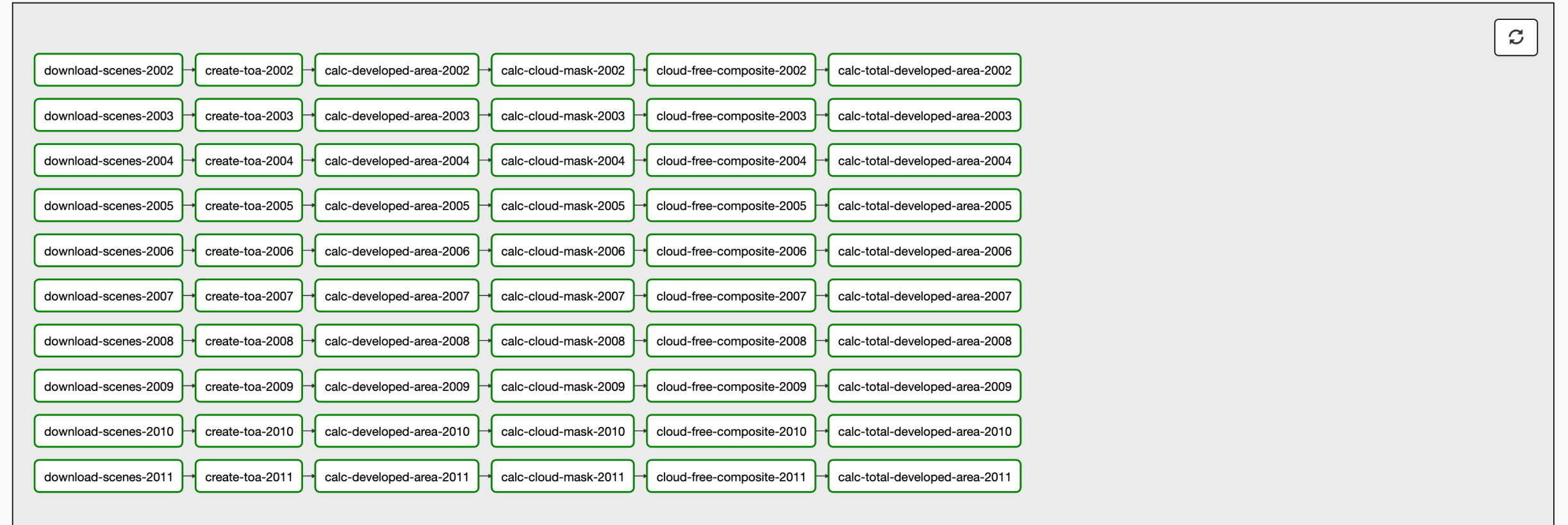
Layout: Left->Right ▾

Go

Search for...

KubernetesPodOperator

success **running** **failed** **skipped** **rescheduled** **retry** **queued** no status



Hardware

Modem and Router





Network Storage



Gigabit Switch

VirtualBox Host

- K8s Master
- Docker Registry
- PostgreSQL





K8s Node

- Airflow
- Processing Pods

Costs

- Modem and Router, Network Storage, and K8s Master
 - Already Had

Costs

- Modem and Router, Network Storage, and K8s Master
 - Already Had
- Gigabit Switch
 - \$35

Costs

- Modem and Router, Network Storage, and K8s Master
 - Already Had
- Gigabit Switch
 - \$35
- K8s Node
 - \$40
 - \$45 – 16GB Memory
 - \$30 – 240 GB SSD

Costs

- Modem and Router, Network Storage, and K8s Master
 - Already Had
- Gigabit Switch
 - \$35
- K8s Node
 - \$40
 - \$45 – 16GB Memory
 - \$30 – 240 GB SSD
- Total: \$150

Results

Results

- 1 Scene Processed in about 5.3 Minutes

Results

- 1 Scene Processed in about 5.3 Minutes
 - 4 scenes processed concurrently

Results

- 1 Scene Processed in about 5.3 Minutes
 - 4 scenes processed concurrently
 - 1 scene every 1.325 minutes

Results

- 1 Scene Processed in about 5.3 Minutes
 - 4 scenes processed concurrently
 - 1 scene every 1.325 minutes
 - 1087 scenes per day

Results

- 1 Scene Processed in about 5.3 Minutes
 - 4 scenes processed concurrently
 - 1 scene every 1.325 minutes
 - 1087 scenes per day
- Landsat Scenes Captured per Day?

Results

- 1 Scene Processed in about 5.3 Minutes
 - 4 scenes processed concurrently
 - 1 scene every 1.325 minutes
 - 1087 scenes per day
- Landsat Scenes Captured per Day?
 - Around 1000

Unexpected Issues

Unexpected Issues

- Managed Switch

Unexpected Issues

- Managed Switch
 - Loud

Unexpected Issues

- Managed Switch
 - Loud
 - Power hungry

Unexpected Issues

- Managed Switch
 - Loud
 - Power hungry
- DIY Pipeline Orchestration with Message Queuing

Unexpected Issues

- Managed Switch
 - Loud
 - Power hungry
- DIY Pipeline Orchestration with Message Queuing
 - Difficult to debug

Unexpected Issues

- Managed Switch
 - Loud
 - Power hungry
- DIY Pipeline Orchestration with Message Queuing
 - Difficult to debug
 - More code written for orchestrating than processing

Shameless Self-Promotion

Q&A