Slides & More: www.kb.gg/georodeo

Global Economic Analysis on a Budget

Kevin Booth

www.kb.gg/georodeo

Who am I?

www.kb.gg/georodeo

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

www.kb.gg/georodeo

Today’s Topic

www.kb.gg/georodeo

Building a Satellite Imagery Processing Pipeline

www.kb.gg/georodeo

A Story

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

Problems with GDP Reports

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

www.kb.gg/georodeo

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

www.kb.gg/georodeo

Software

Software

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

Hardware

www.kb.gg/georodeo

Modem and Router

www.kb.gg/georodeo

Network Storage

www.kb.gg/georodeo

Gigabit Switch

www.kb.gg/georodeo

VirtualBox Host

- K8s Master
- Docker Registry
- PostgreSQL

www.kb.gg/georodeo

K8s Node

- Airflow
- Processing Pods

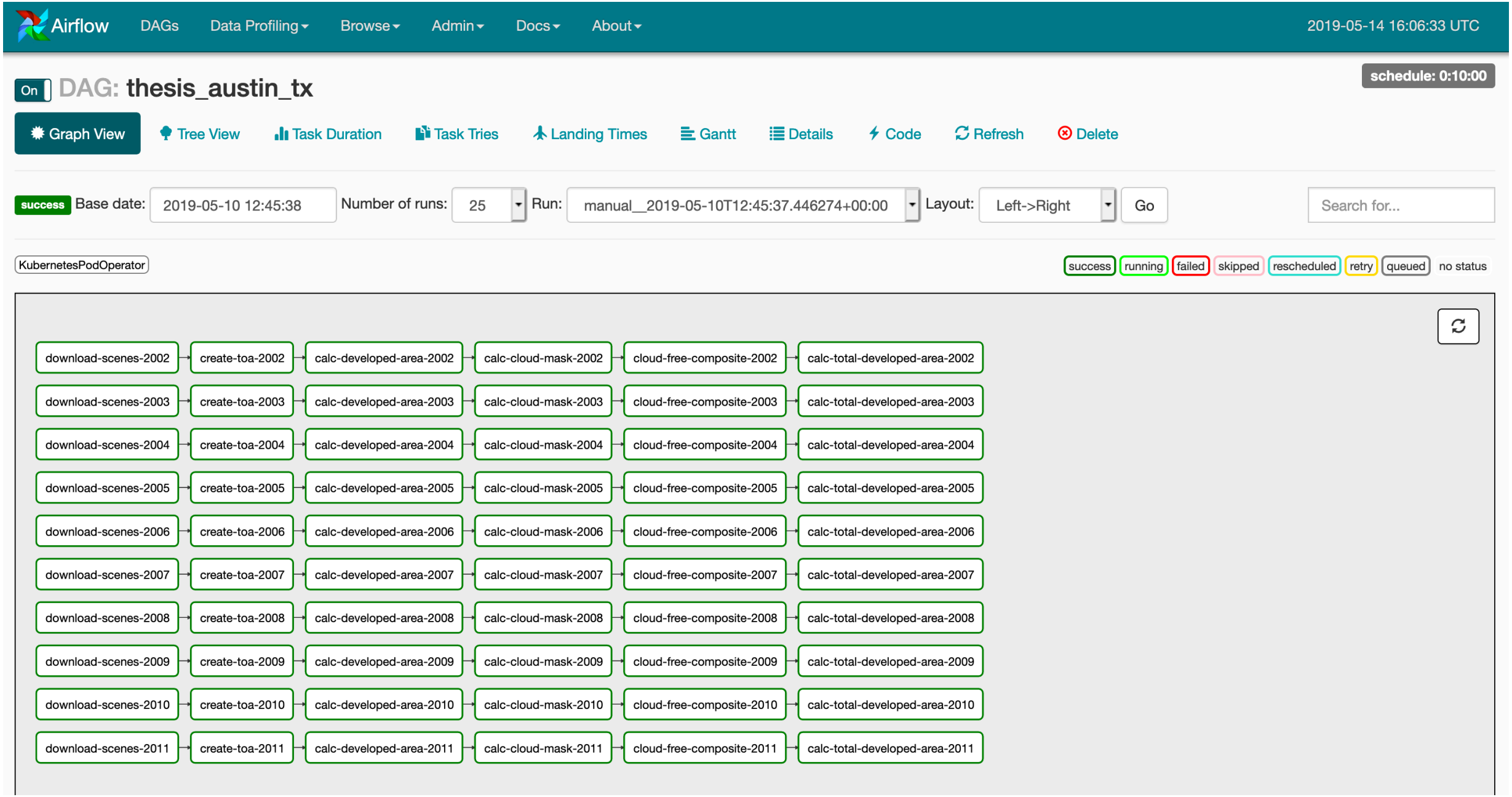
Costs

- 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

www.kb.gg/georodeo

```
default_args = {
    'owner': 'kevin',
    'depends_on_past': False,
    'start_date': datetime.utcnow(),
    'email': ['kevin@kb.gg'],
    'email_on_failure': False,
    'email_on_retry': False,
    'retries': 1,
    'retry_delay': timedelta(minutes=5)
}

secret_file = Secret('volume', '/secrets', 'thesis-db', None)
volume_mount = VolumeMount('research', mount_path='/data', sub_path=None, read_only=False)
volume_config= {'persistentVolumeClaim': {'claimName': 'research'}}
volume = Volume(name='research', configs=volume_config)

dag = DAG('thesis_austin_tx', default_args=default_args, concurrency=4)

for year in range(2002, 2012):
    download_task = KubernetesPodOperator(namespace='thesis',
                          image="registry.kb.gg:5000/thesis:latest",
                          image_pull_policy="Always",
                          image_pull_secrets="dev-regsecret",
                          cmds=["download-scenes"],
                          arguments=["2306", str(year)],
                          labels={"app": "thesis"},
                          secrets=[secret_file],
                          volumes=[volume],
                          volume_mounts=[volume_mount],
                          name="download-scenes-{}".format(year),
                          task_id="download-scenes-{}".format(year),
                          dag=dag,
                          config_file="/data/airflow/config/config")
    create_toa_task = KubernetesPodOperator(namespace='thesis',
                          image="registry.kb.gg:5000/thesis:latest",
                          image_pull_policy="Always",
                          image_pull_secrets="dev-regsecret",
                          cmds=["create-toa"],
                          arguments=["2306", str(year)],
                          labels={"app": "thesis"},
                          secrets=[secret_file],
                          volumes=[volume],
                          volume_mounts=[volume_mount],
                          name="create-toa-{}".format(year),
                          task_id="create-toa-{}".format(year),
                          dag=dag,
                          config_file="/data/airflow/config/config")

    download_task.set_downstream(create_toa_task)
```

www.kb.gg/georodeo



Results

Results

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

Unexpected Issues

Unexpected Issues

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

www.kb.gg/georodeo

Shameless Self-Promotion

www.kb.gg/georodeo

Q&A