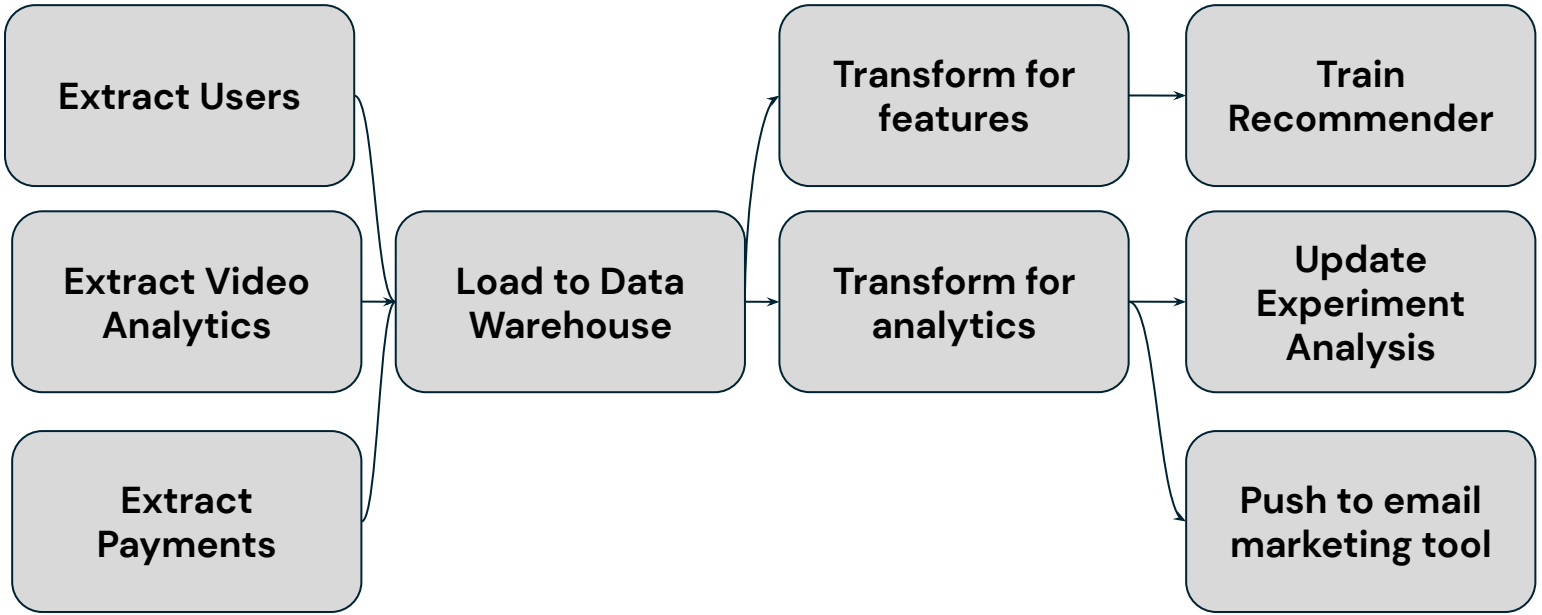
Problem

At start of project, we had some standalone pieces + lots of scripts being run on laptops and That One Server. What we needed...

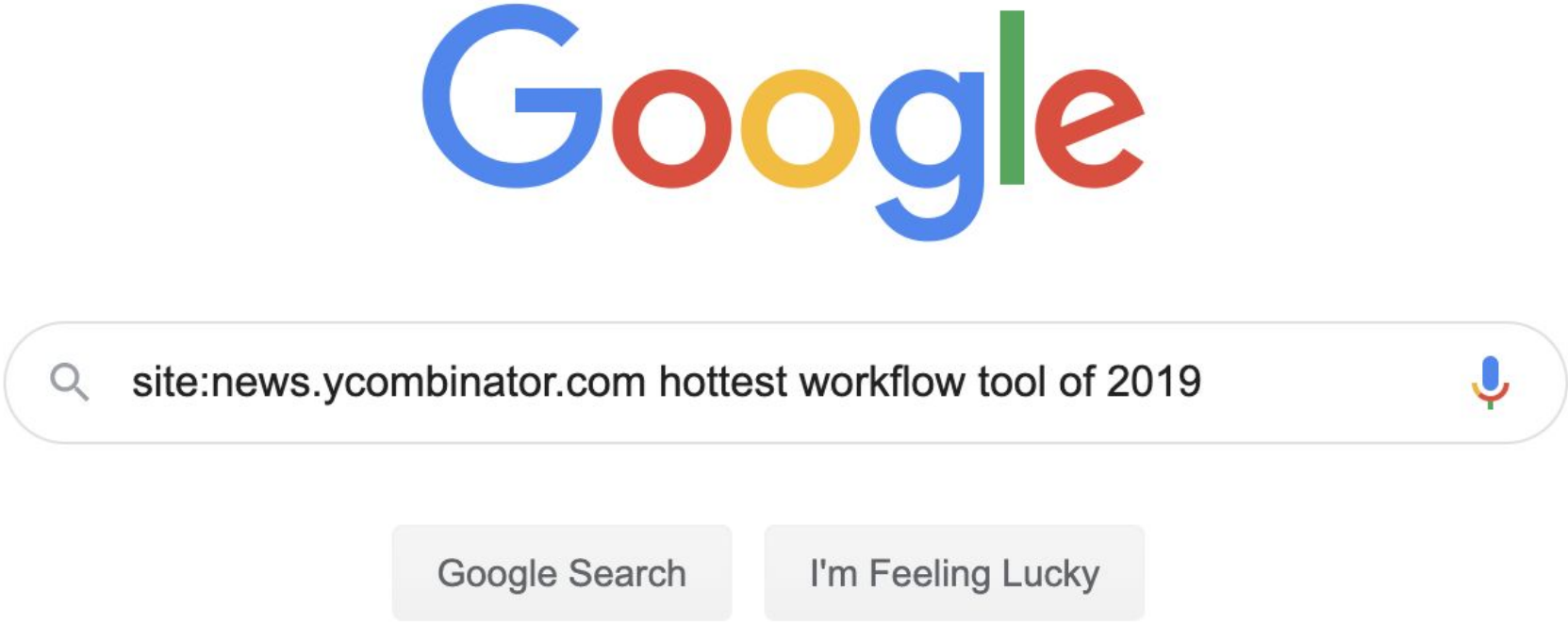
- A place to run ETL/ML/integration tasks
- A tool to orchestrate those tasks into workflows

Example Workflow



Tool Selection

How Not To Do Tool Selection



First Guiding Principle: Operability

Can less than one or more than one person operate it?

- Low conceptual complexity
- Sane development and deployment of workflows
- Logs, metrics, secrets management

Guiding Principle Two: Reliability

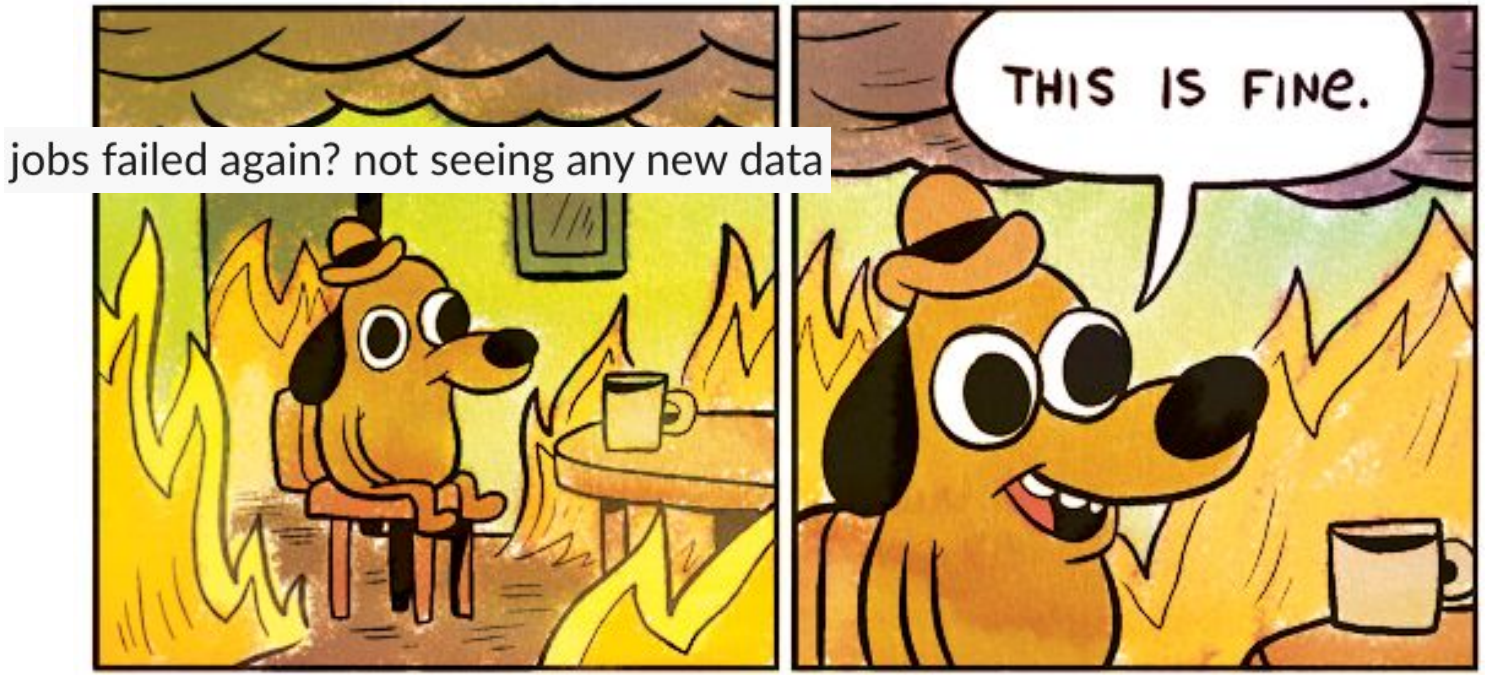
Does workflows run how they're supposed to run?

- Explicit dependencies between steps
- Decouple execution logic and orchestration logic
- Can scale as volume of data increases 10x (more customers *x* more instrumentation *x* more users)

Things We Don’t Care About (Right Now)

- Graphical creation of DAGs
- Programmatically generated DAGs
- Permissioning & security

What We Definitely Don’t Want



Build or buy?

Looked at vendor tools for ETL/ML/integrations, but...

- No single vendor did everything we wanted, and orchestrating one or more vendors + internal tools is hard to do reliably.
- Also development and deployment can be tricky
- We felt comfortable using open source or writing code for all of our workflow tasks. (Some was already written.)

Containers and Kubernetes

We built containers for a small set of initial tasks, but needed a place to run them.

SRE team is already using K8s for our application platform and offered to provision and help maintain a cluster.

Lots of benefits: scaling, secrets management, dev/QA/prod parity, integration with Datadog.

K8s Orchestration: Airflow?

Could use Airflow with KubernetesExecutor, but we weren't enthusiastic for a few reasons:

- Airflow is complex and full of footguns
- We're not going to be using most of the features
- We found a simpler alternative...

Argo

Argo: Kubernetes-Native Workflows

Open source Kubernetes workflow engine built by Intuit.

Active project. 100+ contributors, releases every couple of months.

Two components: **Argo Workflows** and **Argo Events**

argo

What Does Kubernatives-Native Mean?

Workflows and Events are defined as Custom-Resource Definitions.

Workflows can interface with other K8s resources: Secrets, ConfigMaps, Volume Mounts.

Workflows take full advantage of K8s: scheduling affinity, tolerations, resource limits.

Anatomy of a Workflow

Defined in YAML (like everything else in K8s)

“Templates” = workflow steps, Just container images

Declarative DAG. Just tell it dependencies and it does the rest.

```
kind: Workflow
spec:
  templates:
  - name: extractor
    image: extractor:v3
  - name: transformer
    image: transformer:v1
  dag:
    tasks:
     - name: get-views
       template: extractor
       parameters: [{table: views}]
     - name: get-payments
       template: extractor
       parameters: [{table: payments}]
     - name: transform
       template: transformer
       dependencies: [get-views,
get-payments]
```

Running a worfklow

```
> argo submit --watch my-worfklow

Name:                my-worfklow-57r9p
Status:              Done
Started:             Sun Nov 10 07:28:38 -0500 (12 minutes ago)
Duration:            1 minutes 35 seconds

STEP                                PODNAME                              DURATION
 ✔ my-workflow-57r9p
 ├-✔ extract-views                   my-worfklow-57r9p-3933687048  33s
 ├-✔ extract-payments                my-worfklow-57r9p-1906300422  6s
 ├-✔ transform                       my-worfklow-57r9p-1906300422  1m 2s
```

Some Advanced Features

Parameter Passing

```
container:
  args: {{tasks.extract.manifest_location}}
```

Artifacts

```
output:
 artifacts:
 - name: results
   path: /results.csv
```

```
input:
 artifacts:
 - name: training_results
   from: {{tasks.training.results}
   path: /results.csv
```

...More Advanced Features...

Memoized Resubmit

```
argo resubmit --memoized my-worfklow-57r9p
```

Suspend & Resume (Including in DAG)

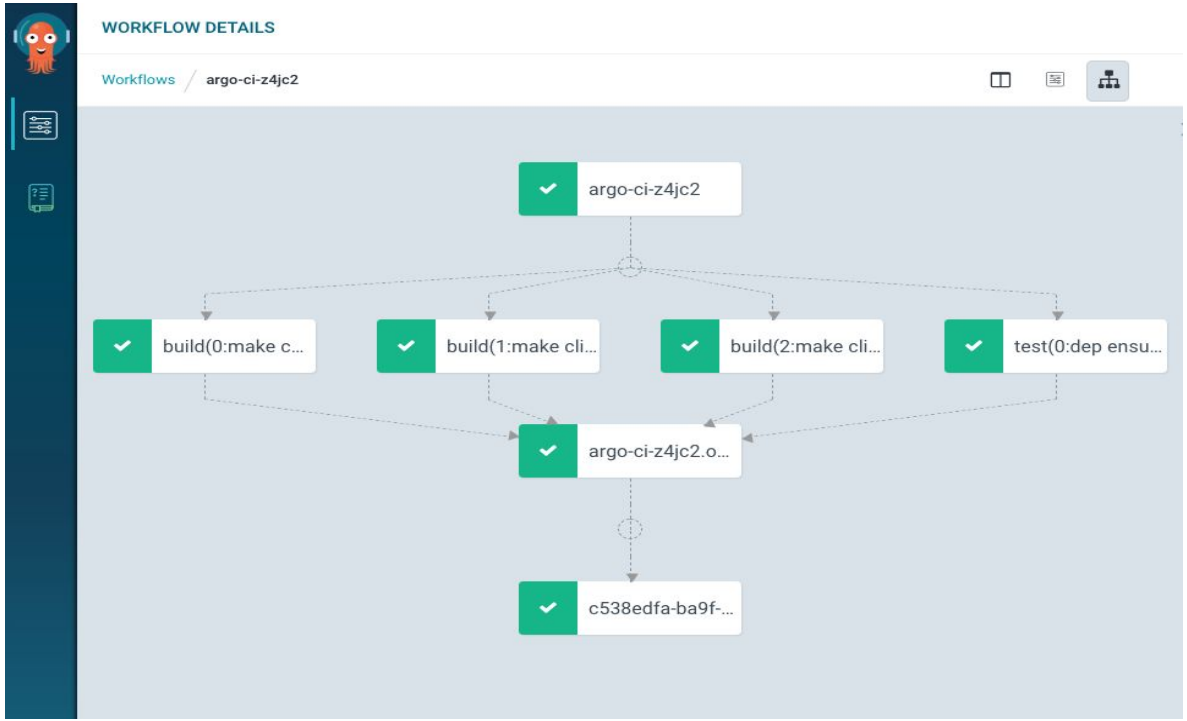
```
argo suspend my-worfklow-57r9p
argo resume my-worfklow-57r9p
```

```
task:
  suspend: {}
```

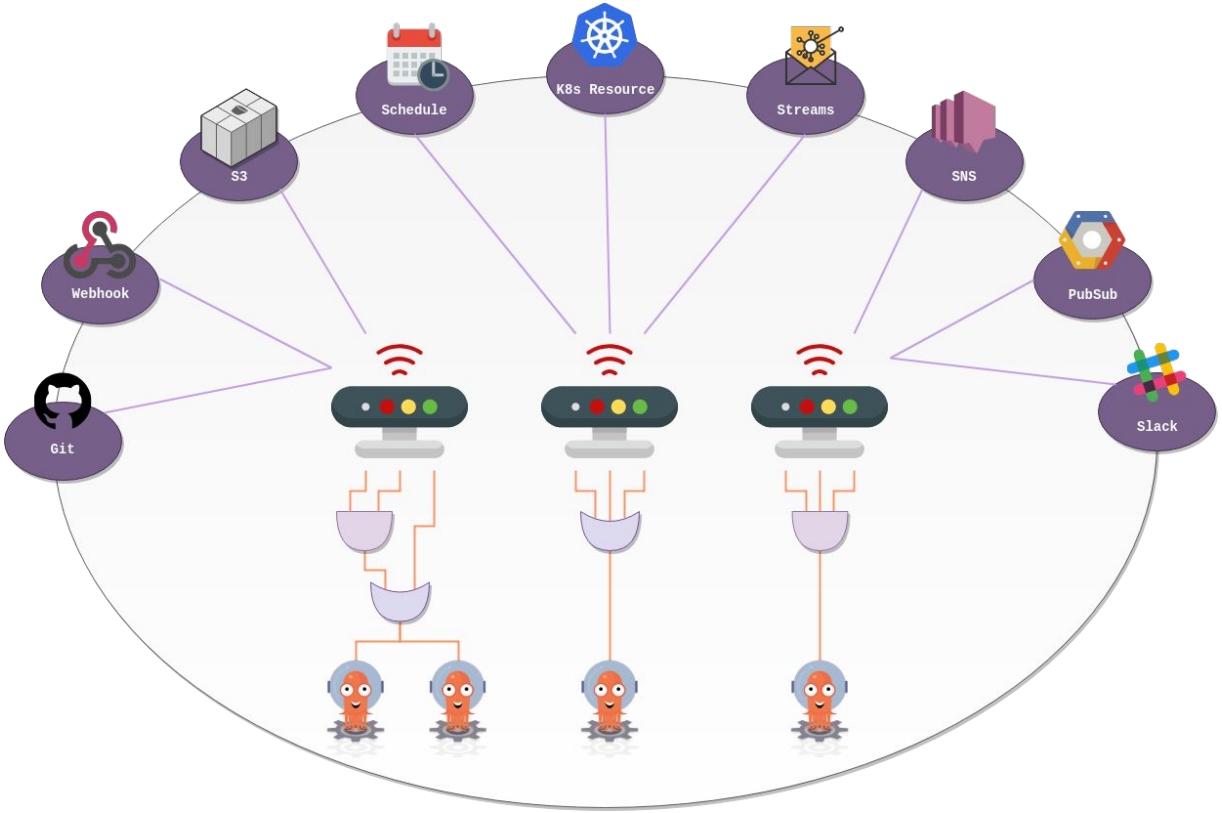
...So Many Features

- Sidecars and daemon containers
- All sorts of DAG shenanigans (conditional tasks, sub-DAGs, generated DAGs, loops, recursive DAGs)
- Post-run hooks
- Create K8s resources as a task

Argo UI



Triggering with Argo Events



Packaging & Deployment

Because Workflows and Events are K8s resources, we can use helm (package manager) to deploy and upgrade workflows:

```
> helm secrets upgrade --namespace prod -f secrets.prod.yaml .

Release "my-workflow" has been upgraded.
LAST DEPLOYED: Sun Nov 10 07:55:12 2019
NAMESPACE: prod
STATUS: DEPLOYED
```

Makes dev -> QA -> prod deployments very seamless.

Implementation Experience

Took about a week to set up Argo. (Caveat: not including Kubernetes set up time.)

Using DataDog for log aggregation.

Have been running two production DAGs with ~20 tasks for the past six months.

One production outage (fixed by a restart).

Wishlist & Future Considerations

Currently no way to limit DAG concurrency — can be an issue with time-triggered workflows.

Argo Events is a little complex (Events, Gateways, Sensors). We toughed it out but you can run workflows through an API if you want.