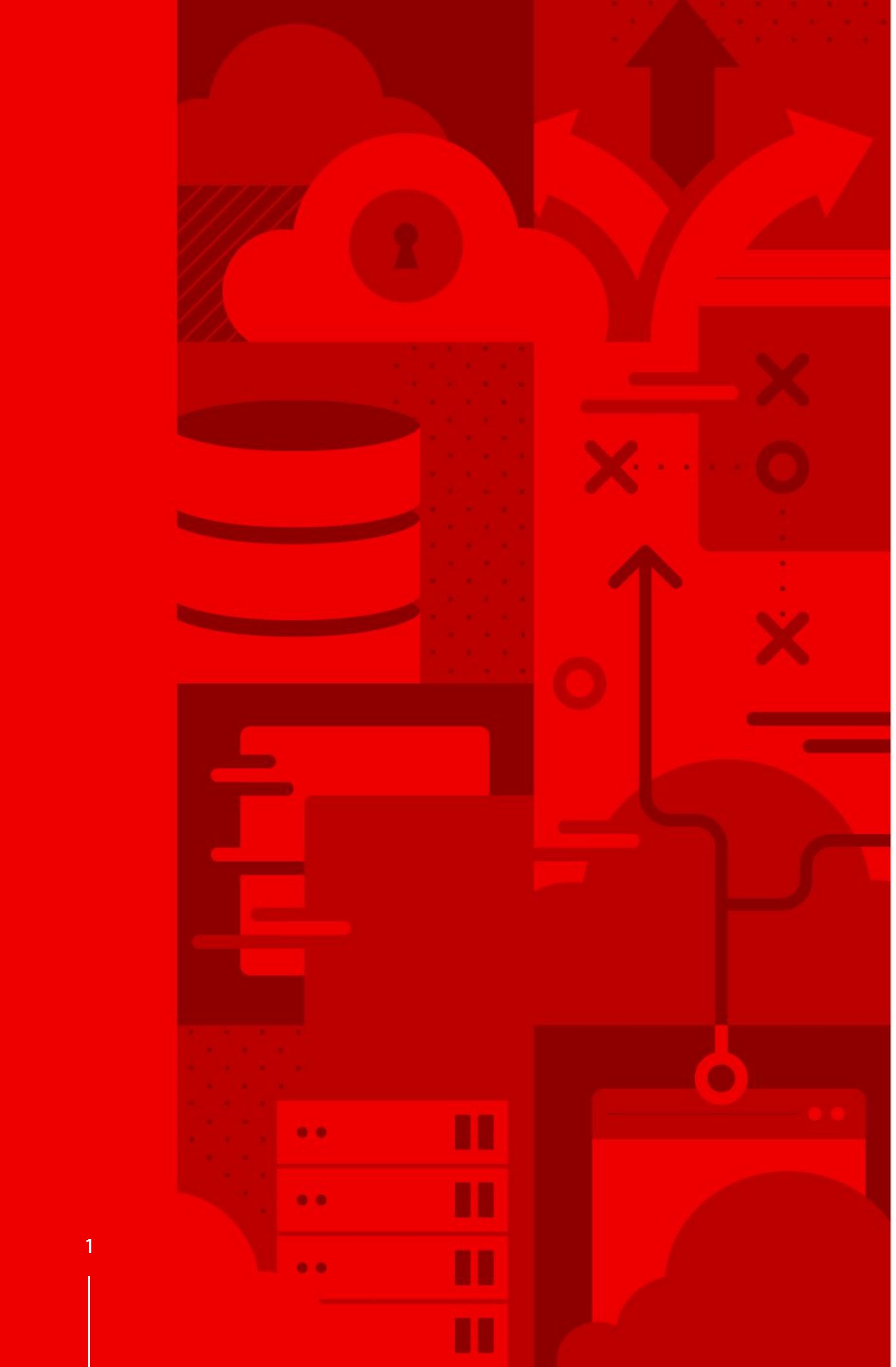


Argo Events

Introduction and demo for ODH community

Tom Coufal
Senior Software Engineer

Our story

Briefly about what we do and what we use it for

Data Ingestion

Enrich data sets based of new data availability. Either plain data transfers or with preprocessing

Workflow re-runs

Re-run an analysis when notebooks are updated or when new data is available

Events consolidation

Triggering the same workload based on different events and/or multiple events happening at the same time

Argo Events

The Event-driven Workflow Automation Framework

Argo Events - an Argo project
First release - v0.5, in July 2018

Current version is v0.14

Targeting **v1.0** release for May 2020

1 owner, 6 maintainers, 69 contributors

498 stars on GitHub

GitHub: argoproj/argo-events

Project Homepage

Argo Events

The Event-driven Workflow Automation Framework

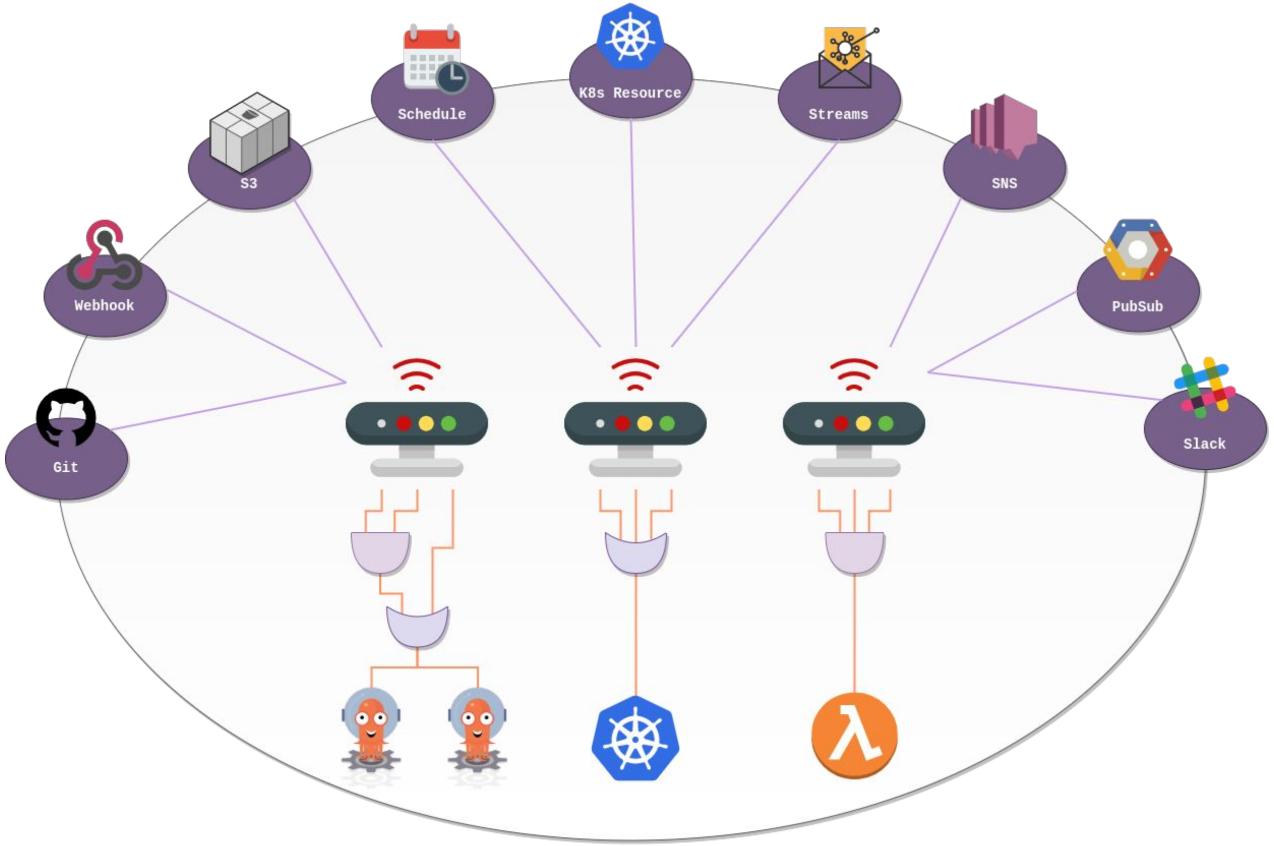
Ingest
Listens for events from various sources

Consolidate
Events are grouped and monitored via a gateway controller.

Dispatch
Triggers an Argo Workflow a circuit of Workflows or a different workload like Lambda, OpenWhisk, Kafka messages, etc.

Argo Events overview

Event sources, gateways, sensors and triggers



Event Sources

As of version 0.14

- AMQP
- AWS SNS
- AWS SQS
- Cron Schedules
- GCP PubSub
- GitHub
- GitLab
- HDFS
- File Based Events
- Kafka
- Minio
- NATS
- MQTT
- K8s Resources
- Slack
- StorageGrid
- Webhooks
- Stripe
- NSQ
- Emitter
- Redis
- Azure Events Hub

Triggers

As of version 0.14 + on the roadmap

Argo Workflows

Standard K8s Objects

HTTP Requests

AWS Lambda

NATS Messages

Kafka Messages

Slack Notifications

Argo Rollouts

Apache OpenWhisk

Custom (build your own)

AirFlow

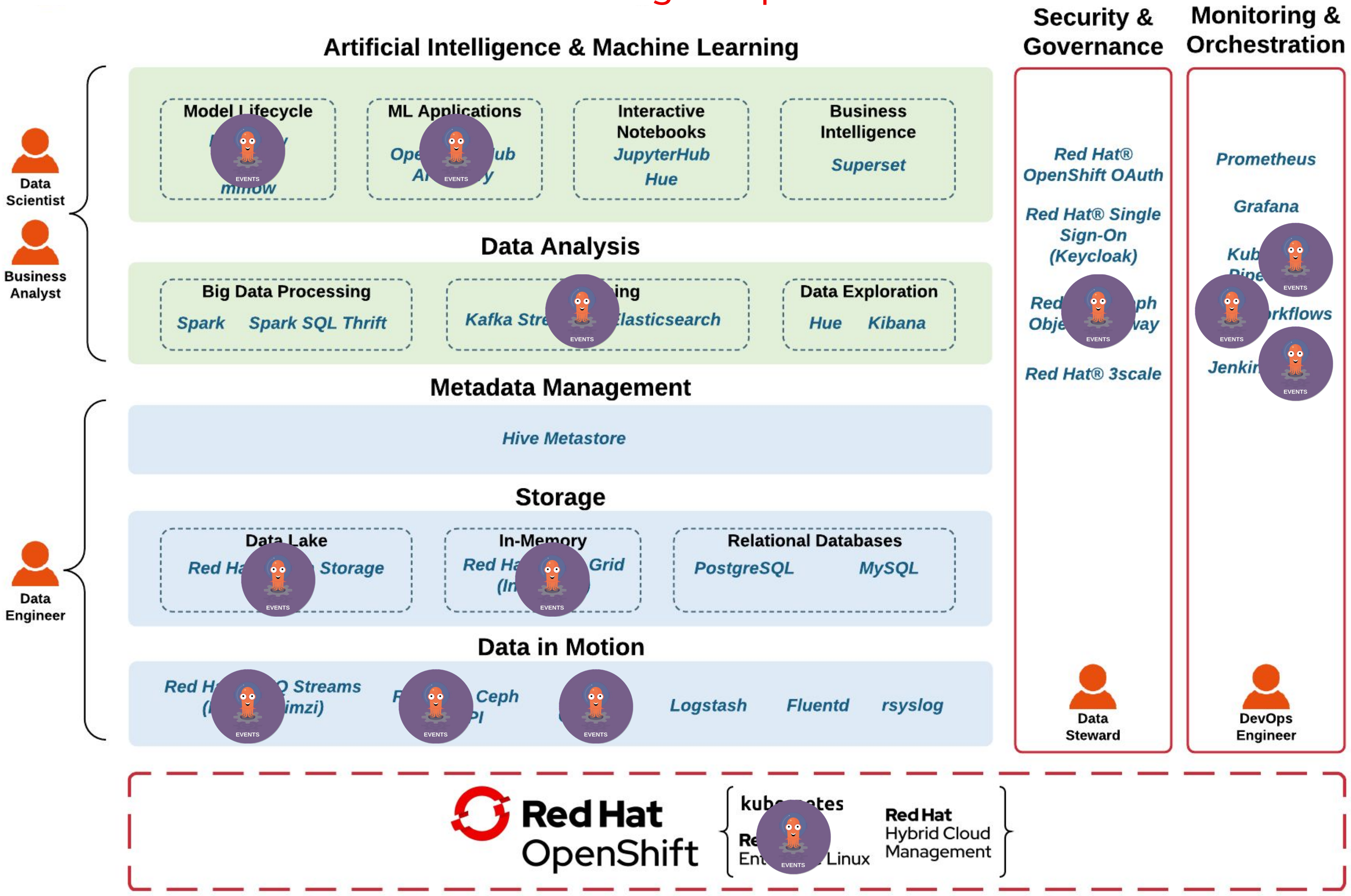
Kubeflow

TektonCD

On Roadmap

Argo Events + ODH

Fitting the picture



Artificial Intelligence & Machine Learning

Model Lifecycle
y
mlflow

ML Applications
Ope ub
Ar y

Interactive Notebooks
JupyterHub
Hue

Business Intelligence
Superset

Data Analysis

Big Data Processing
Spark Spark SQL Thrift

ing
Kafka Stre lasticsearch

Data Exploration
Hue Kibana

Metadata Management

Hive Metastore

Storage

Data Lake
Red Ha Storage

In-Memory
Red Ha Grid
(In

Relational Databases
PostgreSQL MySQL

Data in Motion

Red H Streams
(imzi)
F Ceph
PI
Logstash Fluentd rsyslog

Data Scientist

Business Analyst

Data Engineer

Security & Governance

Red Hat® OpenShift OAuth
Red Hat® Single Sign-On (Keycloak)
Red ph
Obj vay
Red Hat® 3scale

Data Steward

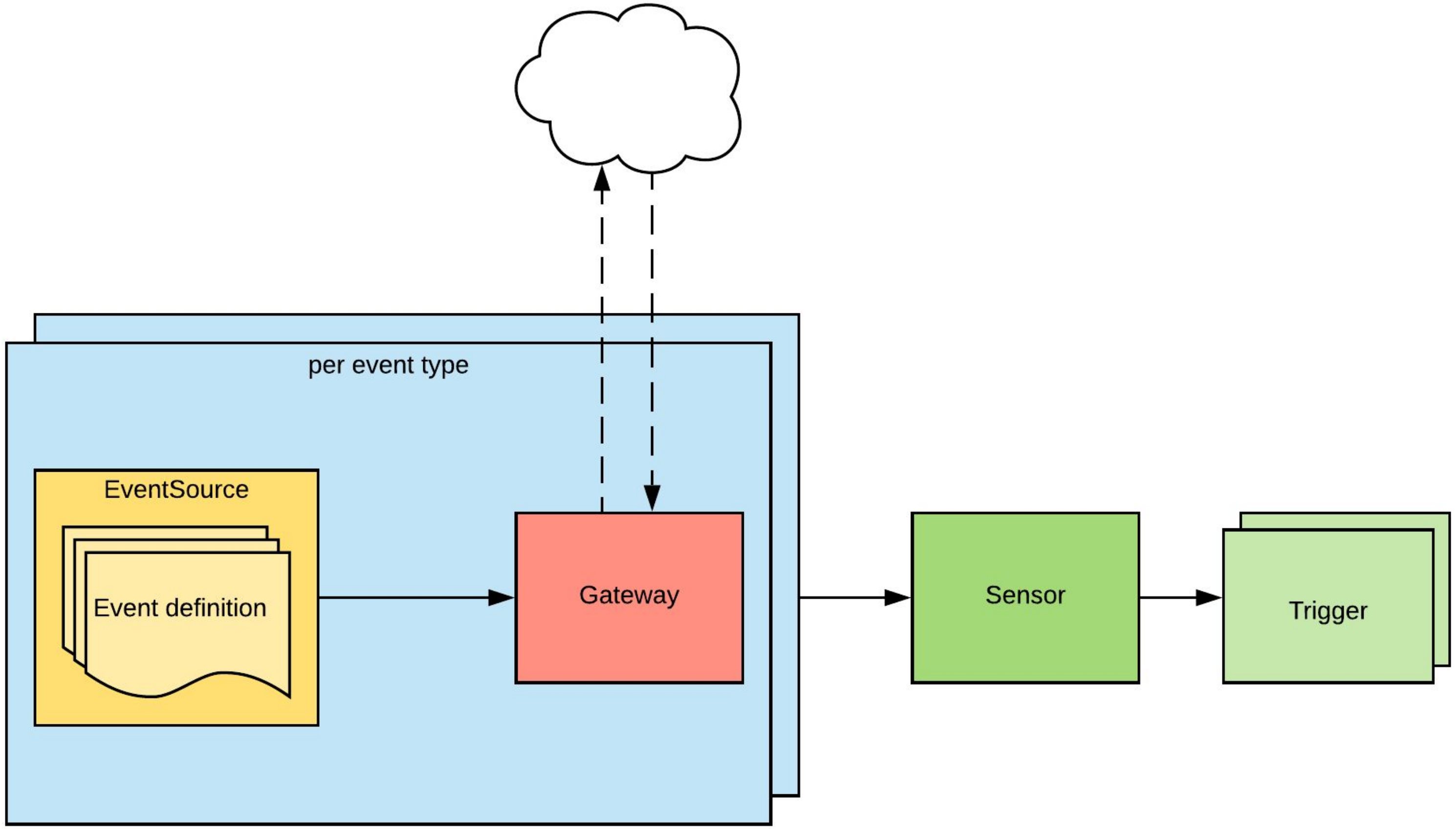
Monitoring & Orchestration

Prometheus
Grafana
Kub
Pipe
rkflows
Jenkir

DevOps Engineer

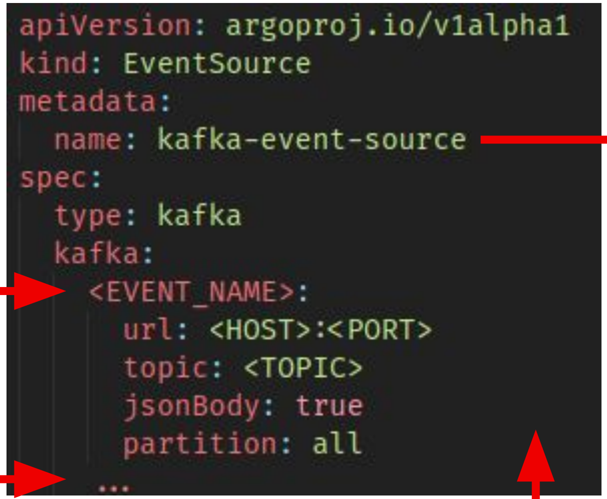
Red Hat OpenShift
kub etes
Re
Ent Linux
Red Hat Hybrid Cloud Management

Under the hood

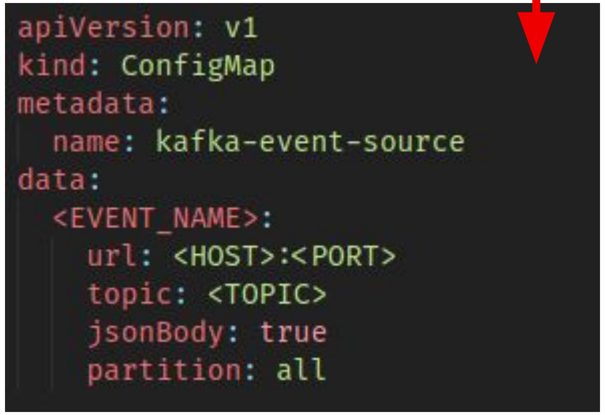


Event Source and Gateway

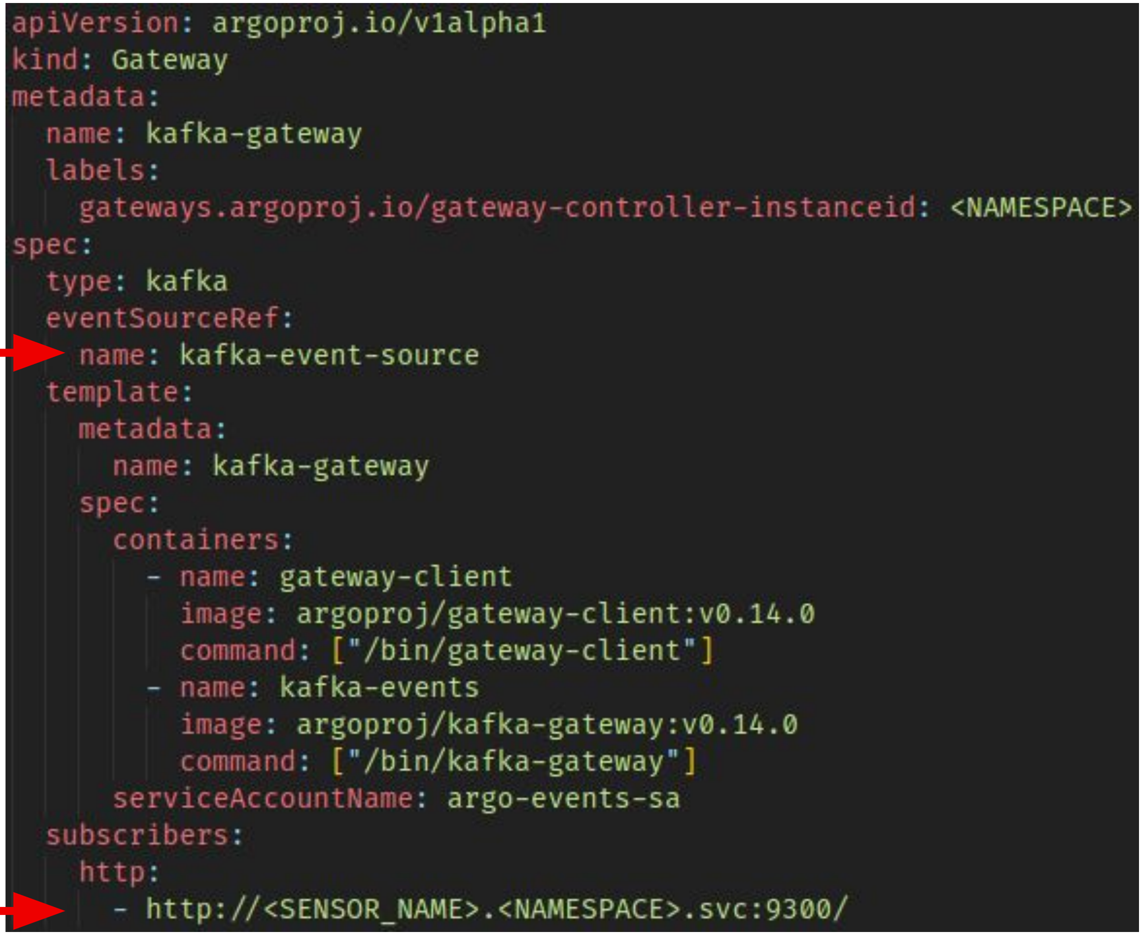
A Gateway depends on single Event Source, which can define multiple events



```yaml
apiVersion: argoproj.io/v1alpha1
kind: EventSource
metadata:
  name: kafka-event-source
spec:
  type: kafka
  kafka:
    <EVENT_NAME>:
      url: <HOST>:<PORT>
      topic: <TOPIC>
      jsonBody: true
      partition: all
    ...
```



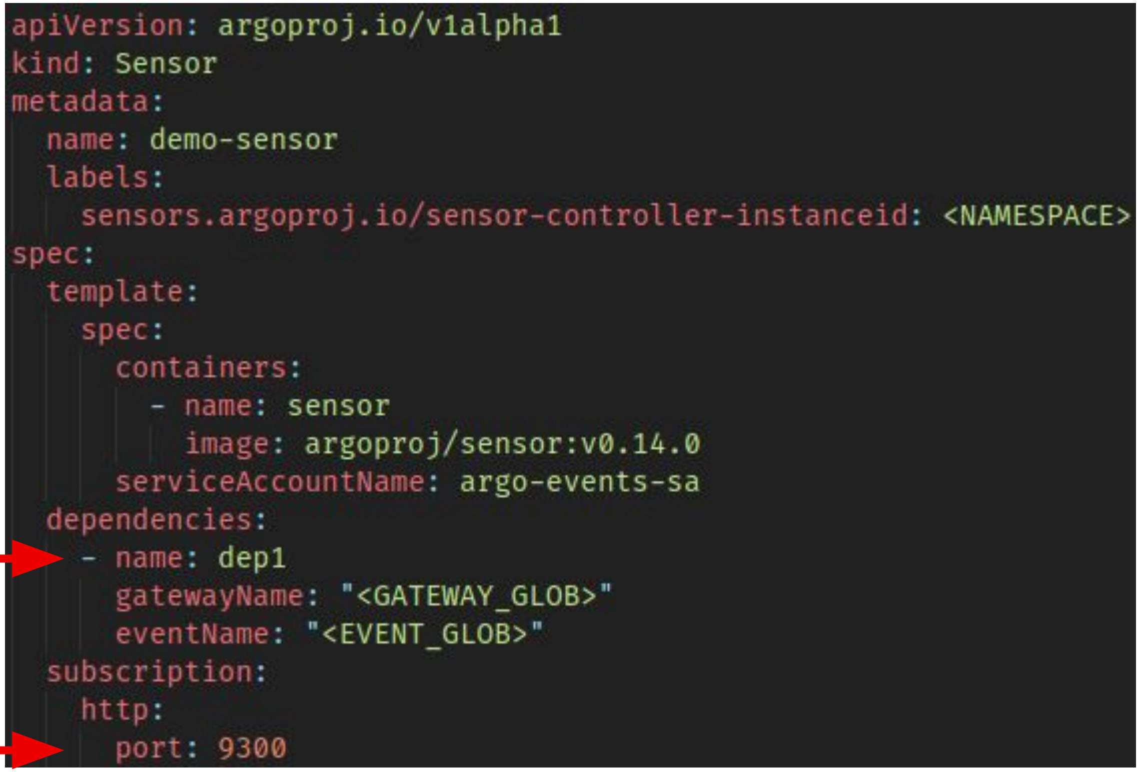
```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: kafka-event-source
data:
  <EVENT_NAME>:
    url: <HOST>:<PORT>
    topic: <TOPIC>
    jsonBody: true
    partition: all
```



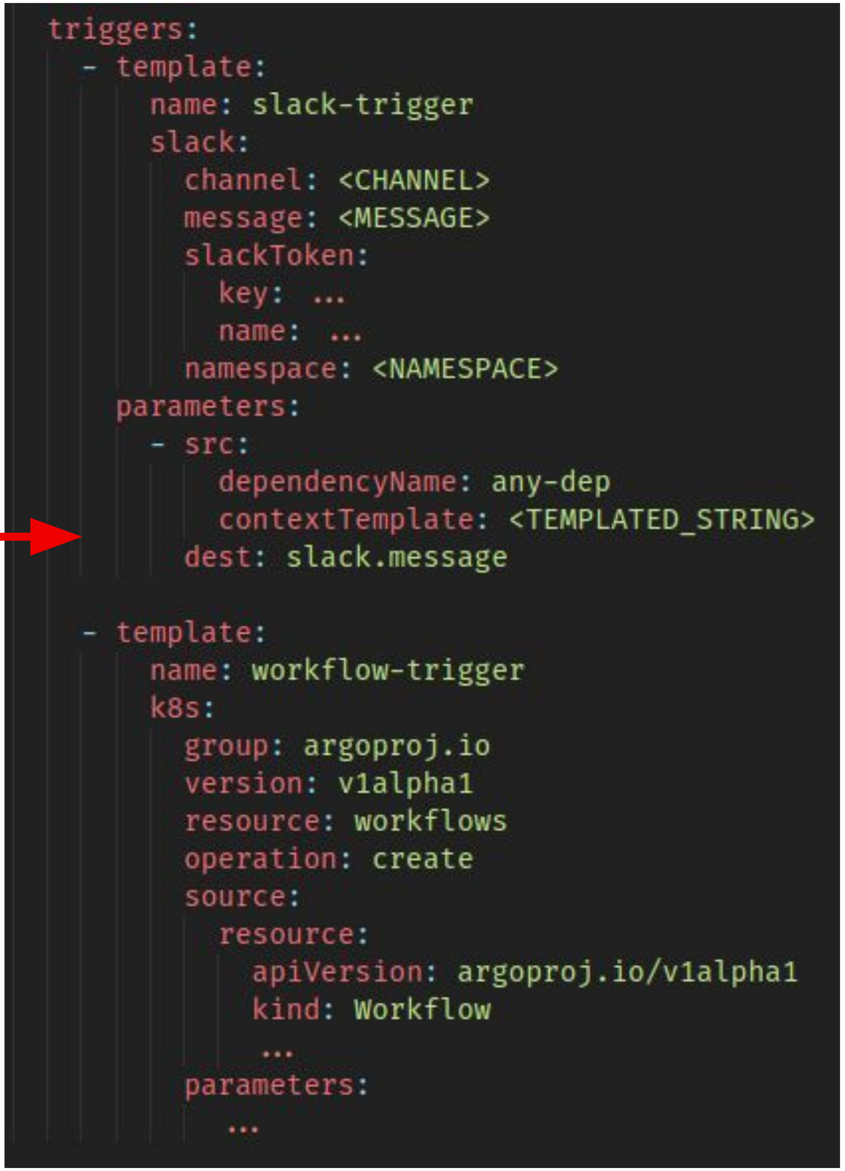
```yaml
apiVersion: argoproj.io/v1alpha1
kind: Gateway
metadata:
  name: kafka-gateway
  labels:
    gateways.argoproj.io/gateway-controller-instanceid: <NAMESPACE>
spec:
  type: kafka
  eventSourceRef:
    name: kafka-event-source
  template:
    metadata:
      name: kafka-gateway
    spec:
      containers:
        - name: gateway-client
          image: argoproj/gateway-client:v0.14.0
          command: ["/bin/gateway-client"]
        - name: kafka-events
          image: argoproj/kafka-gateway:v0.14.0
          command: ["/bin/kafka-gateway"]
      serviceAccountName: argo-events-sa
  subscribers:
    http:
      - http://<SENSOR_NAME>.<NAMESPACE>.svc:9300/
```

Sensors and Triggers

Sensor resolves dependencies, then triggers actions



```yaml
apiVersion: argoproj.io/v1alpha1
kind: Sensor
metadata:
  name: demo-sensor
  labels:
    sensors.argoproj.io/sensor-controller-instanceid: <NAMESPACE>
spec:
  template:
    spec:
      containers:
        - name: sensor
          image: argoproj/sensor:v0.14.0
      serviceAccountName: argo-events-sa
  dependencies:
    - name: dep1
      gatewayName: "<GATEWAY_GLOB>"
      eventName: "<EVENT_GLOB>"
  subscription:
    http:
      port: 9300
```



```yaml
triggers:
  - template:
      name: slack-trigger
      slack:
        channel: <CHANNEL>
        message: <MESSAGE>
        slackToken:
          key: ...
          name: ...
        namespace: <NAMESPACE>
    parameters:
      - src:
          dependencyName: any-dep
          contextTemplate: <TEMPLATED_STRING>
        dest: slack.message

  - template:
      name: workflow-trigger
      k8s:
        group: argoproj.io
        version: v1alpha1
        resource: workflows
        operation: create
        source:
          resource:
            apiVersion: argoproj.io/v1alpha1
            kind: Workflow
            ...
        parameters:
          ...
```

Why don’t you use X instead?

Why to choose Argo Events?

Kubernetes - CronJob

We’re not triggering a simple scripts, our workflows are multistage, more complex

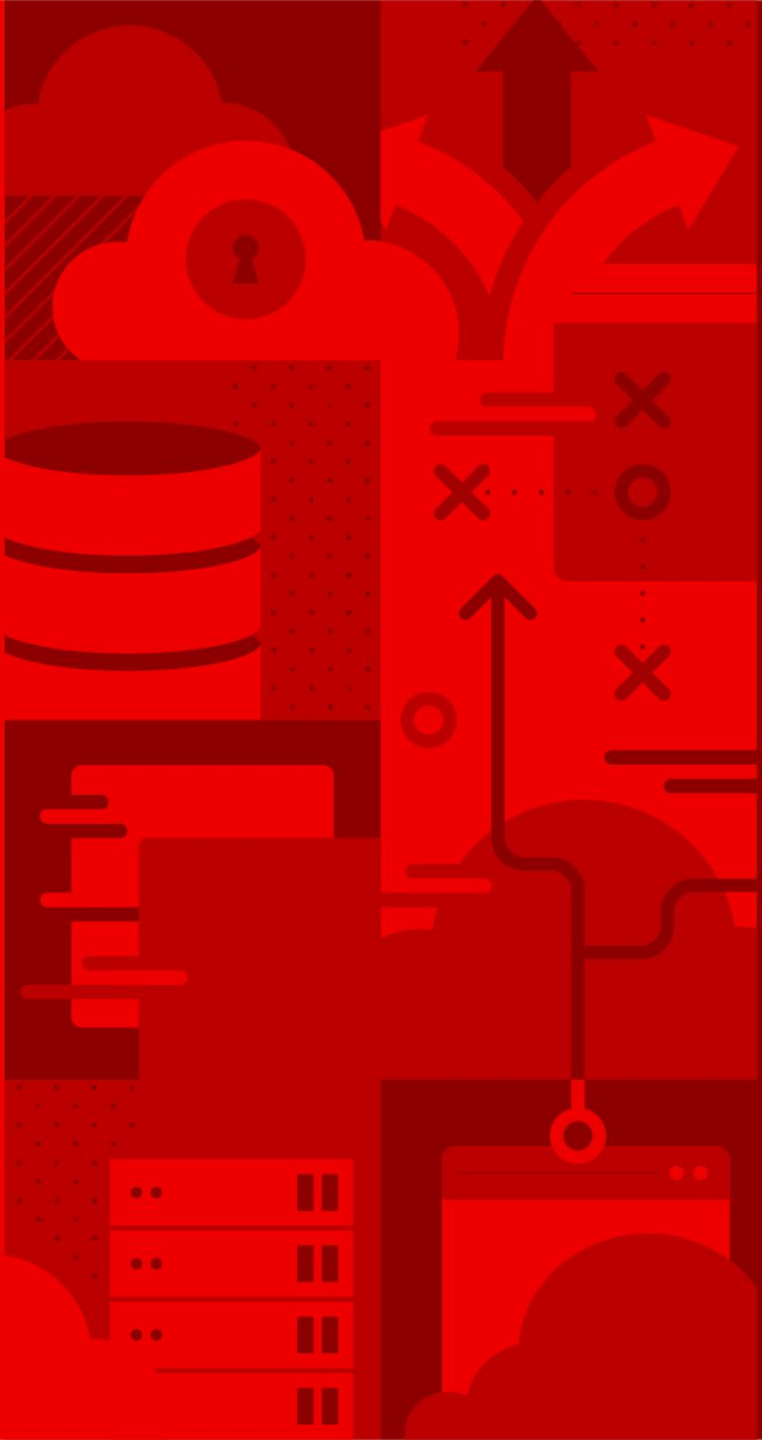
AirFlow scheduling and triggers

No event aggregation, only cron and a webhook available

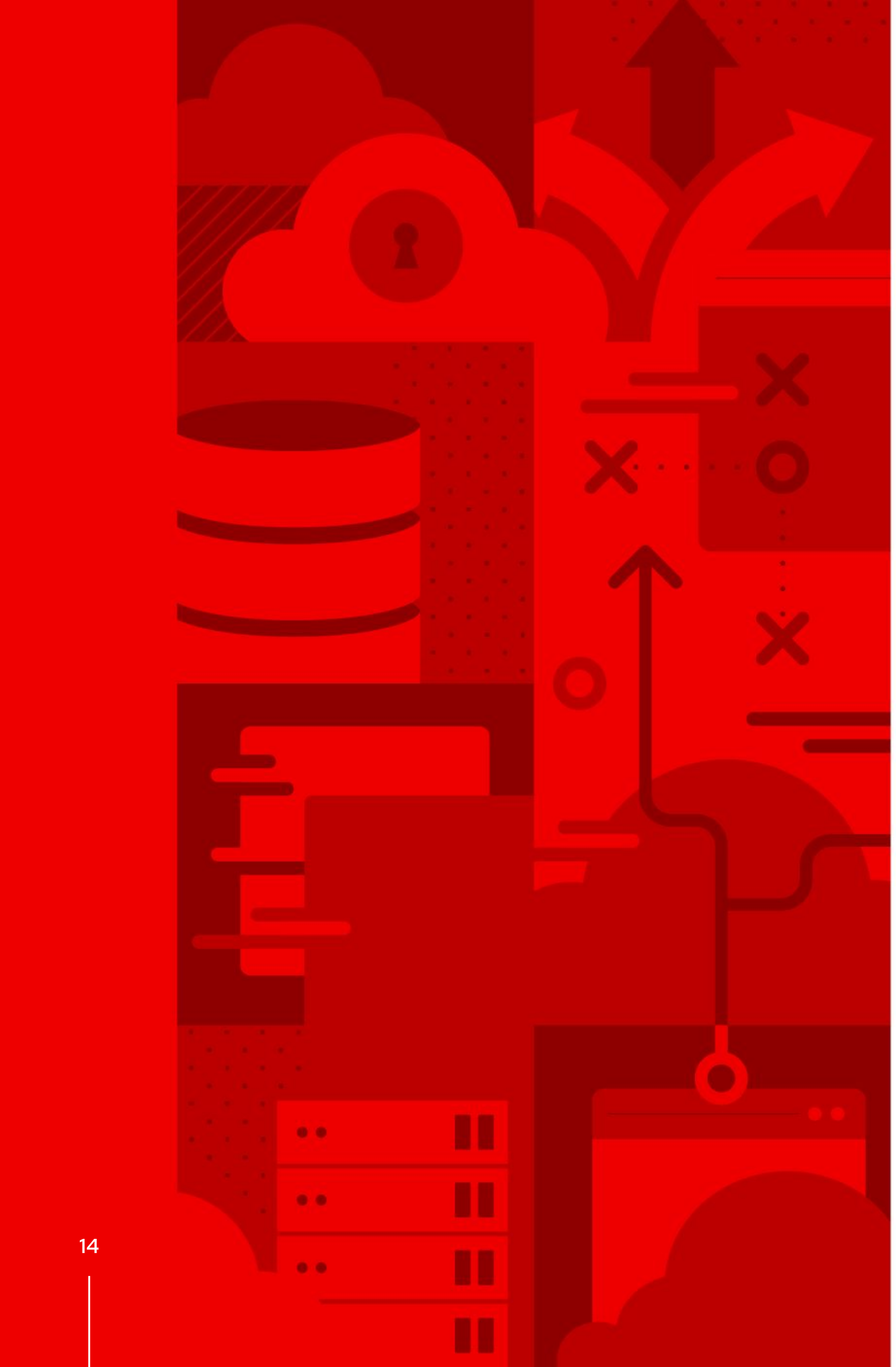
Argo Workflows - CronWorkflow

~~At the time we needed it, it wasn’t a thing in Argo yet~~

We want: different action triggers, to trigger on multiple events that happen at the same time, to trigger the same resource from different triggers, to change parameters before triggering



Demo time!



Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

f facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat