Developing a CD pipeline with Knative

Andrea Frittoli
Developer Advocate
andrea.frittoli@uk.ibm.com
@blackchip76

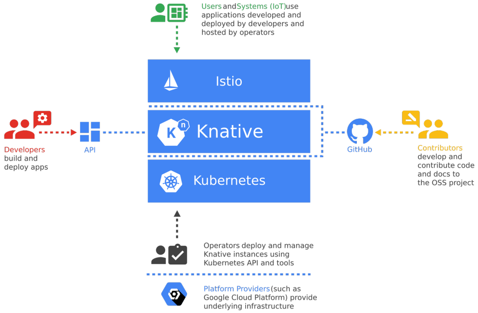
DevOps Meetup Singapore

IBM Cloud

A Bit of History

Knative

- Beginning of 2018…
- Knative:
  - Build
  - Eventing
  - Serving
- Contributors:
  - Google
  - Pivotal
  - IBM
  - RedHat
  - Cloudbees
  - …and others



Users and Systems (IoT) use applications developed and deployed by developers and hosted by operators

Istio

Knative

Kubernetes

Developers build and deploy apps

API

GitHub

Contributors develop and contribute code and docs to the OSS project

Operators deploy and manage Knative instances using Kubernetes API and tools

Platform Providers (such as Google Cloud Platform) provide underlying infrastructure

~Sept 2018: Knative Pipelines

Latest news!

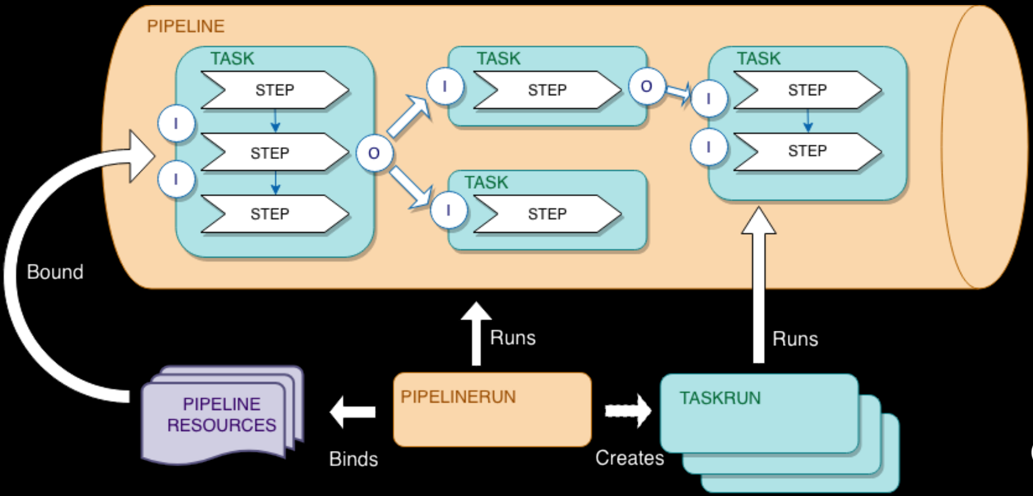
- Focus on CI/CD
- Deploy “anywhere“
- Compatible with Knative Build
- *tektoncd/pipeline*
  - New logo
  - @CD Foundation
  - Roadmap WIP
  - Alpha APIs

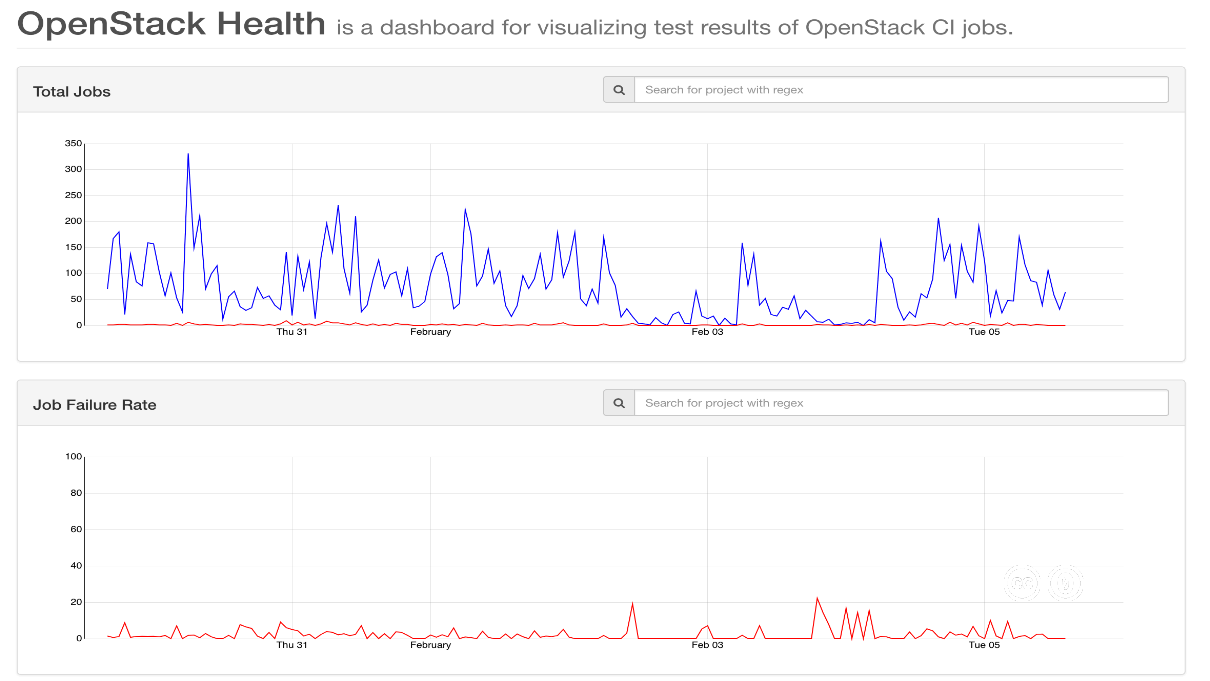
Community

- *Valid for Knative. Tekton TBD.*
- Steering Commitee (SC)
- Technical Oversight Commitee (TOC)
- Various Contribution profiles
- Design, issues: on GitHub
- Communication:
  - Weekly video meetings, recorded, Build WG
  - Asynch: Knative Users / Developers ML
  - Sync: slack.knative.dev

Tekton Pipelines

Cloud Native Pipelines





OpenStack Health is a dashboard for visualizing test results of OpenStack CI jobs.

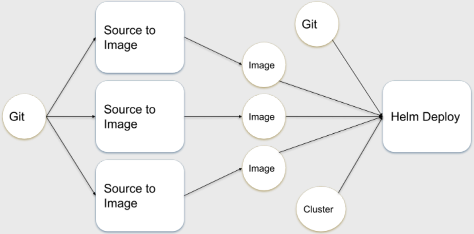
Total Jobs

Job Failure Rate

Inputs, Outputs & DAG

- Steps are sequential
- Tasks are a Directed Acyclic Graph

- Order defined by:
  - *from*: input from another task's output
  - *runAfter*: enforced task ordering



Under the Hood

Custom Resources

CRDs: Task(Run), Pipeline(Run), PipelineResource

Services in the *tekton-pipelines* namespace:

- Webhook Service: resource validation
- Controller Service:
    - Handles inputs and outputs
    - Calculates the DAG
    - Provisions pods and containers

Custom Resource Provisioning:

- Via YAML
- Via Go API
- Labels!

Pods, Entrypoints & Volumes

Steps (of a Task):

- Containers in one POD (single node)
- Any container image
- Entrypoint re-written
- Serial execution
- Resource allocation?

TaskRun:

- Provisions a POD
- Deployes entrypoint tool
- Input/output containers
- User containers (steps)

Volumes:

- EmptyDir for workspace/home
- Tools (entrypoint)
- Secrets
- Any user ConfigMap / Volume
- (Optionally) Pipeline Share

PipelineRun:

- Several PODs, different nodes
- Shared storage: PVC or GCS

Source to Image to Deploy

IBM Cloud

CD Pipeline as code

- Pipeline and Tasks in git (YAML)
- Parameters for env/run specific
- Security?

```
apiVersion: tekton.dev/v1alpha1
kind: PipelineResource
metadata:
  name: health-helm-git-knative
  labels:
    tag: agreatrelease
spec:
  type: git
  params:
    - name: revision
      value: knative
    - name: url
      value: https://github.com/afrittoli/health-helm
```

```
metadata:
  name: mycluster
spec:
  type: cluster
  params:
    - name: name
      value: mycluster
    - name: url
      value: https://mycluster.containers.cloud.ibm.com
    - name: username
      value: admin
  secrets:
    - fieldName: token
      secretKey: tokenKey
      secretName: cluster-secrets
    - fieldName: cadata
      secretKey: cadataKey
      secretName: cluster-secrets
```

```
metadata:
  name: health-api-image
spec:
  type: image
  params:
    - name: url
      value: registry.ng.bluemix.net/andreaf/health-api
```

Using Kaniko

- Features:
  - Build from Context and Dockerfile
  - Unpriviledged
  - Reproducible
  - Remote caching of layers
  - Base images caching (warmer)

- Dockefile?
  - Most common changes last
  - Careful with COPY/ADD
  - Remove what you don’t need

Tekton and Knative

Pipelines and Knative Build

CI for OpenStack Health

CI with Tekton Pipelines

Asynchronous Pipelines

Triggering and Knative Eventing

Tekton and Development

Conclusions

Shall I use Tekton Pipelines?

Roadmap

References

Q&A