Cloud Native CI/CD with Knative and Tekton Pipelines

—

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

—

Open Source Summit Japan 2019

July 19th 2019

A Bit of History

Knative

- Beginning of 2018...
- Knative:
  - Build
  - Eventing
  - Serving

- OpenSource
- Contributors:
  - Google
  - Pivotal
  - IBM
 - RedHat
  - Cloudbees
  - ...and others

THE LINUX FOUNDATION

Community

– Steering Commitee (SC)
– Technical Oversight Commitee (TOC)
– Dedicated Working Groups (WG)
– Various Contribution profiles
– Design, issues: on GitHub
– Communication:
    – WG periodic meetings, recorded
    – Asynch: Knative Users / Developers ML
    – Sync: slack.knative.dev

THE
LINUX
FOUNDATION

Knative Serving

THE
LINUX
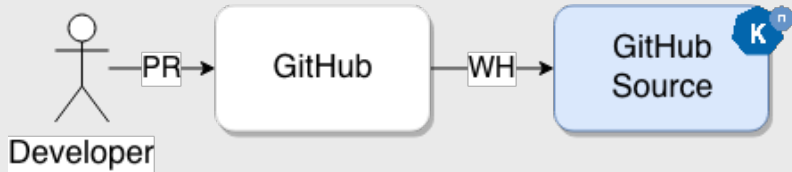FOUNDATION

Knative Serving

- Scale to Zero
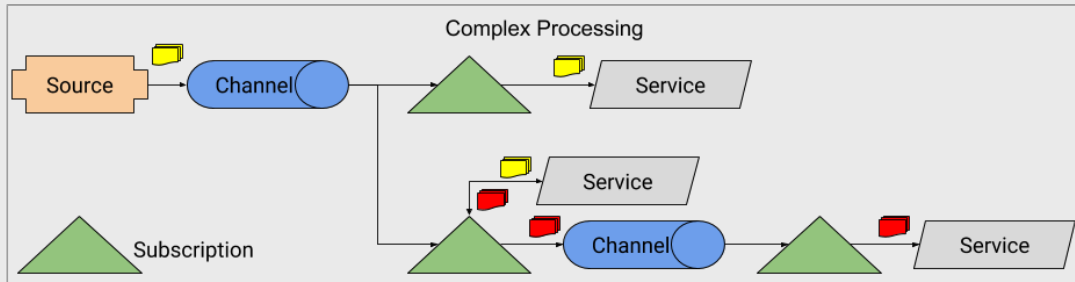- Scale up based on metrics
- Multiple revisions
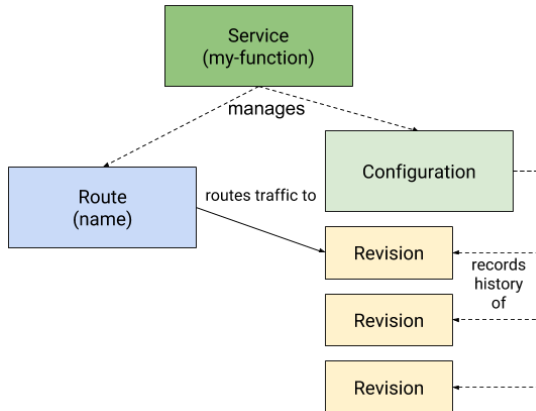
Knative Serving Demo
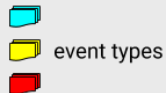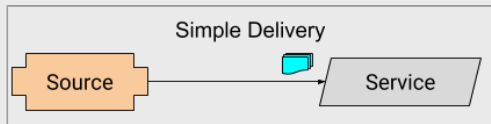
Knative Eventing

# Knative Eventing

Knative Eventing Demo

Knative Build and Pipelines

Knative Build

- – Source to image
- – Build Templates
- – How to make a new revision

```yaml
apiVersion: serving.knative.dev/v1alpha1
kind: Service
spec:
  runLatest:
    configuration:
      build:
        apiVersion: build.knative.dev/v1alpha1
        kind: Build
        spec:
          serviceAccountName: build-bot
          source:
            git:
              url: https://github.com/mc/simple-app.git
              revision: master
          template:
            name: kaniko
            arguments:
              - name: IMAGE
                value: docker.io/{USERNAME}/myapp:latest
          timeout: 10m
      revisionTemplate:
        spec:
          container:
            image: docker.io/{USERNAME}/my:latest
            imagePullPolicy: Always
```

THE
LINUX
FOUNDATION

Going further
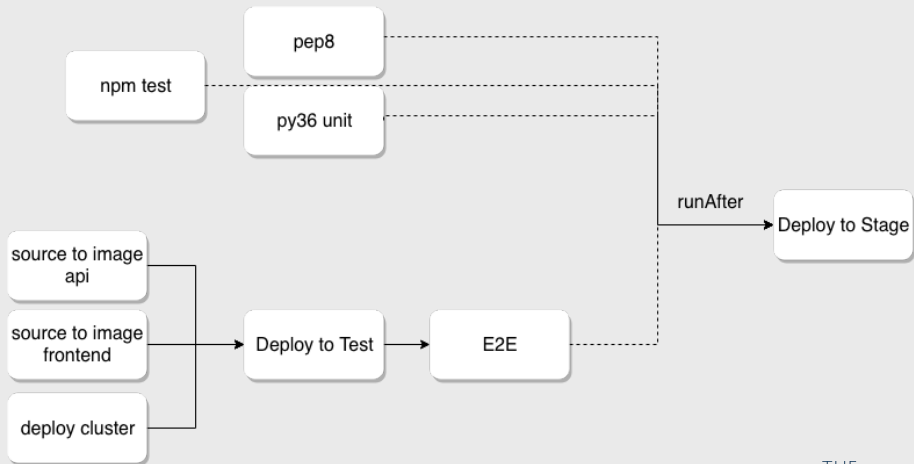
~Sept 2018: Knative Pipelines

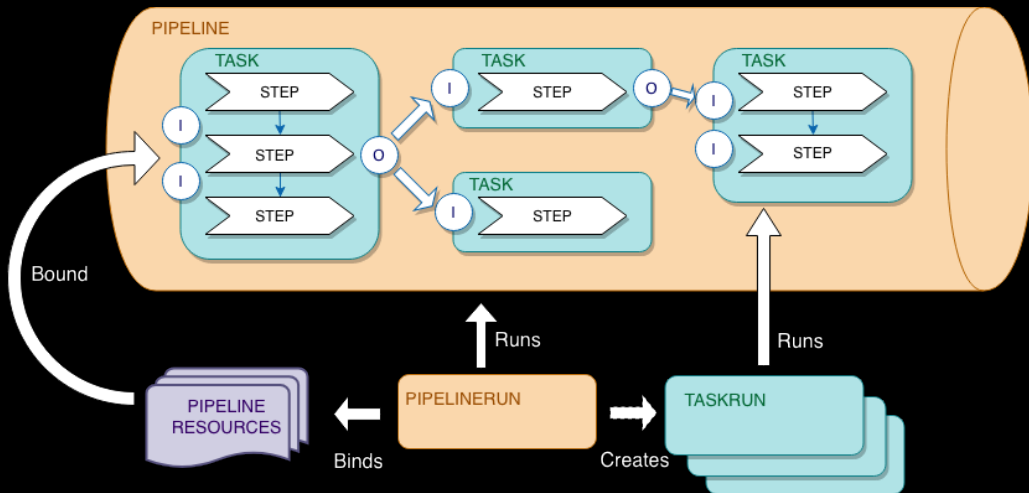# Latest news!

- Tekton pipelines (March 2019)
- Focus on CI/CD
- @CD Foundation (March 2019)
- Deploy "anywhere"
- Replaces Knative Build (June 2019)
- Bootstrap governance

Tekton Pipelines
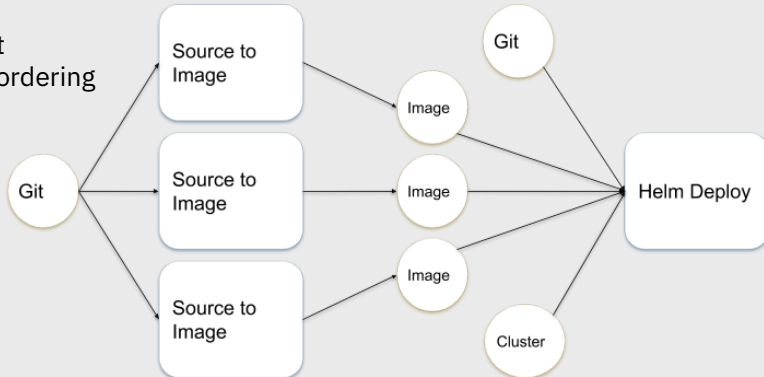
THE
LINUX
FOUNDATION

Cloud Native Pipelines

Sample pipeline

- Steps are sequential
- Tasks are a Directed Acyclic Graph
- Order defined by:
  - *from*: input / output
  - *runAfter*: enforced ordering

Sample Task: Source to Image

Source to Image (spec only):

```
inputs:
  resources:
    - name: workspace
      type: git
  params:
    - name: pathToDockerFile
      default: Dockerfile
    - name: pathToContext
      default: .
    - name: useImageCache
      default: "true"
    - name: imageTag
      default: "default"
outputs:
  resources:
    - name: builtImage
      type: image
volumes:
  - name: kaniko-base-image-cache
    persistentVolumeClaim:
      claimName: kaniko-base-image-cache
steps:
  - name: build-and-push
    image: gcr.io/kaniko-project/executor
    command:
      - /kaniko/executor
```

```
    args:
      - --cache=${inputs.params.useImageCache}
      - --cache-dir=/cache
      - --dockerfile=${inputs.params.pathToDockerFile}
      - --reproducible
      - --destination=${outputs.resources.builtImage.url}:
${inputs.params.imageTag}
      - --context=/workspace/workspace/${inputs.params.
pathToContext}
    volumeMounts:
      - name: kaniko-base-image-cache
        mountPath: /cache
```

THE
LINUX
FOUNDATION

Pipeline as code

- Pipeline and Tasks:
  - Static definition, stored in git
- PipelineRuns and TaskRuns:
  - Specific to one execution
  - Generated programmatically
- Parameters and Resources:
  - Env / run specific

```yaml
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
```

```yaml
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
```

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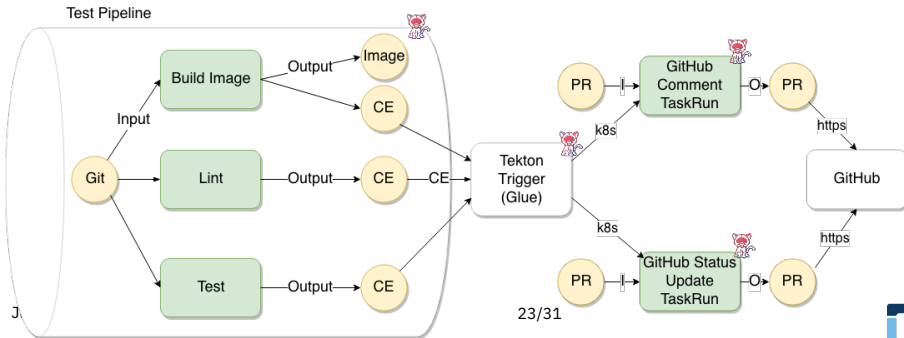
THE
LINUX
FOUNDATION

Tekton Projects

THE
LINUX
FOUNDATION
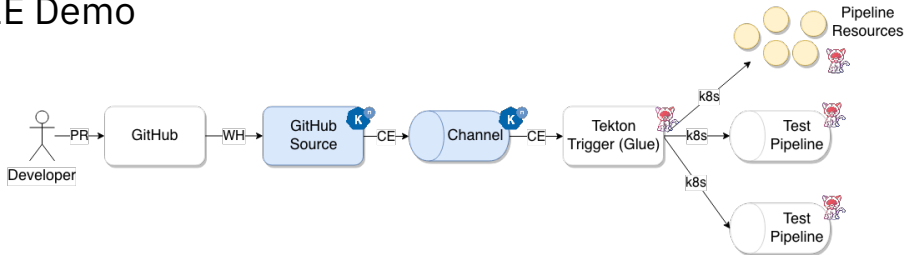
All Tekton projects

- Pipeline
- Dashboard
- CLI
- Catalog
- Trigger
- Plumbing

- Operator
- Community
- Friends
- Website (https://tekton.dev/)

THE
LINUX
FOUNDATION

Full Demo: Knative, Tekton and Triggers

THE
LINUX
FOUNDATION

E2E Demo

23/31

Tekton Pipeline for Knative Serving

THE
LINUX
FOUNDATION

Tekton in local development
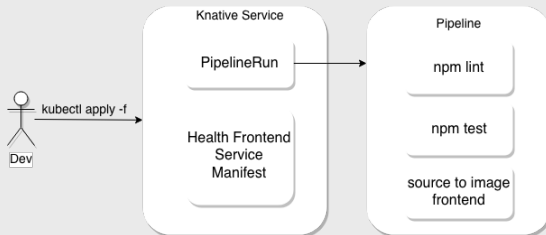
KService Example

```yaml
apiVersion: serving.knative.dev/v1alpha1
kind: Service
metadata:
  name: health-frontend
  labels:
    app: health
    component: frontend
    tag: "__TAG__"
spec:
  runLatest:
    configuration:
      build:
        apiVersion: tekton.dev/v1alpha1
        kind: PipelineRun
        metadata:
          labels:
            app: health
            component: frontend
            tag: "__TAG__"
        spec:
          pipelineRef:
            name: dev-test-build-frontend
          params:
            - name: imageTag
              value: "__TAG__"
            - name: nodeTestImage
              value: __NODE_IMAGE_NAME__
          trigger:
            type: manual
          resources:
            - name: src
              resourceRef:
                name: __GIT_RESOURCE_NAME__
            - name: builtImage
              resourceRef:
                name: __IMAGE_RESOURCE_NAME__
      revisionTemplate:  # template for building Revision
        spec:
          container:
            image: us.icr.io/andreaf/health-frontend:__TAG__
            imagePullPolicy: Always
            env:
              - name: API_URL
                value: http://health-api.containers.domain
            ports:
              - name: http1
                containerPort: 80
                protocol: TCP
            livenessProbe:
              httpGet:
                path: /
            readinessProbe:
              httpGet:
                path: /
```

THE LINUX FOUNDATION

Conclusions

Dev, CI and CD

Development Workflow:
- + Local or in-cluster
- + Reproducibility
- + Parallel execution
- - No support for local source

CI / CD:
- + Integrates with CI systems
  - Jenkins X
  - Prow
  - RedHat OpenShift
  - IBM Cloud Pipelines
  - Kabanero (https://kabanero.io/)
  - ...and more

CI / CD:
- + Pipeline readability
- + Re-usable blocks
- + Best practices
- + Small Footprint
- + Tested on iteself (dogfooding - WIP)
- - Depends on k8s
- - Security and observability are WIP

Roadmap

- Workflow features
  - Conditional Execution
  - Notifications
  - Build Results and logs
  - Pluggable Tasks & Triggers
- Community Library
  - Include via URI
  - Dashboard / CLI integration

- Beta APIs
- Extendibility
- Security
- Observability
- Dogfooding
- UI: Dashboard and CLI

THE
LINUX
FOUNDATION

References

- This Talk: https://github.com/afrittoli/tekton_pipelines_knative_intro/tree/devoxx-fr
- Tekton Links:
    - https://tekton.dev/, https://cd.foundation/
    - https://github.com/tektoncd/pipeline
    - https://github.com/tektoncd/pipeline
    - https://github.com/tektoncd/pipeline/blob/master/api_compatibility_policy.md
    - https://github.com/tektoncd/pipeline/blob/master/roadmap-2019.md
- Knative Community: https://github.com/knative/docs/tree/master/community
- Kaniko: https://github.com/GoogleContainerTools/kaniko
- Source code and my blog:
    - https://github.com/afrittoli/health-helm/tree/knative
    - https://github.com/afrittoli/openstack-health/tree/knative-eventing
    - https://github.com/afrittoli/github_tekton_glue
    - https://andreafrittoli.me
- IBM Cloud: https://cloud.ibm.com

THE
LINUX
FOUNDATION

Q&A

THE
LINUX
FOUNDATION