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
A Bit of History
Knative

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

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

- Steering Committee (SC)
- Technical Oversight Committee (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

July 19th 2019, #ossummit
Knative Serving
Knative Serving

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

Simple Delivery

Source -> Service

Complex Processing

Source -> Channel
   -> Service
   --> Subscription
   --> Channel
   --> Service

event types

July 19th 2019, #ossummit
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
```
Going further

- npm test
- pep8
- py36 unit
- source to image api
- source to image frontend
- deploy cluster
- Deploy to Test
- E2E
- Deploy to Stage

runAfter
~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
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-projectexecutor
    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
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
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
```

July 19th 2019, #ossommit
Tekton Projects
All Tekton projects

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

- Operator
- Community
- Friends
- Website (https://tekton.dev/)
Full Demo: Knative, Tekton and Triggers
E2E Demo

[Diagram of the E2E Demo process involving GitHub, Tekton, and Kubernetes (k8s) with various stages such as PR, GitHub Source, Channel, Tekton Trigger (Glue), and Test Pipeline.]
Tekton Pipeline for Knative Serving
Tekton in local development

Knative Service

PipelineRun

Health Frontend Service Manifest

Pipeline

npm lint

npm test

source to image frontend

kubectl apply -f

Dev
KService Example

```yaml
apiVersion: serving.knative.dev/v1alpha1
kind: Service
metadata:
  name: health-front-end
  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-front-end
    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-front-end:__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: /
```

July 19th 2019, #oss summit
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
- + Pipeline readability
- + Re-usable blocks
- + Best practices
- + Small FOOTprint
- + Tested on itself (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
References

- This Talk:
- Tekton Links:
  - https://github.com/tektoncd/pipeline
  - https://github.com/tektoncd/pipeline
  - https://github.com/tektoncd/pipeline/blob/master/api_compatibility_policy.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