TEKTON

Intro to Tekton

Chris Baumbauer

iam.cab/tekton-talk

Who am I?

- Chris Baumbauer (cab)
  - cab@cabnetworks.net
    @cab105
    github.com/cab105
- Software Developer
  - Kompose, Kmachine, Aktion
- DevOps
  - Kubernetes, CI/CD
- Tinkerer

Who are you?

- Build Engineer
- DevOps
- Developer
- Curious about CI/CD
- Curious about k8s or Serverless

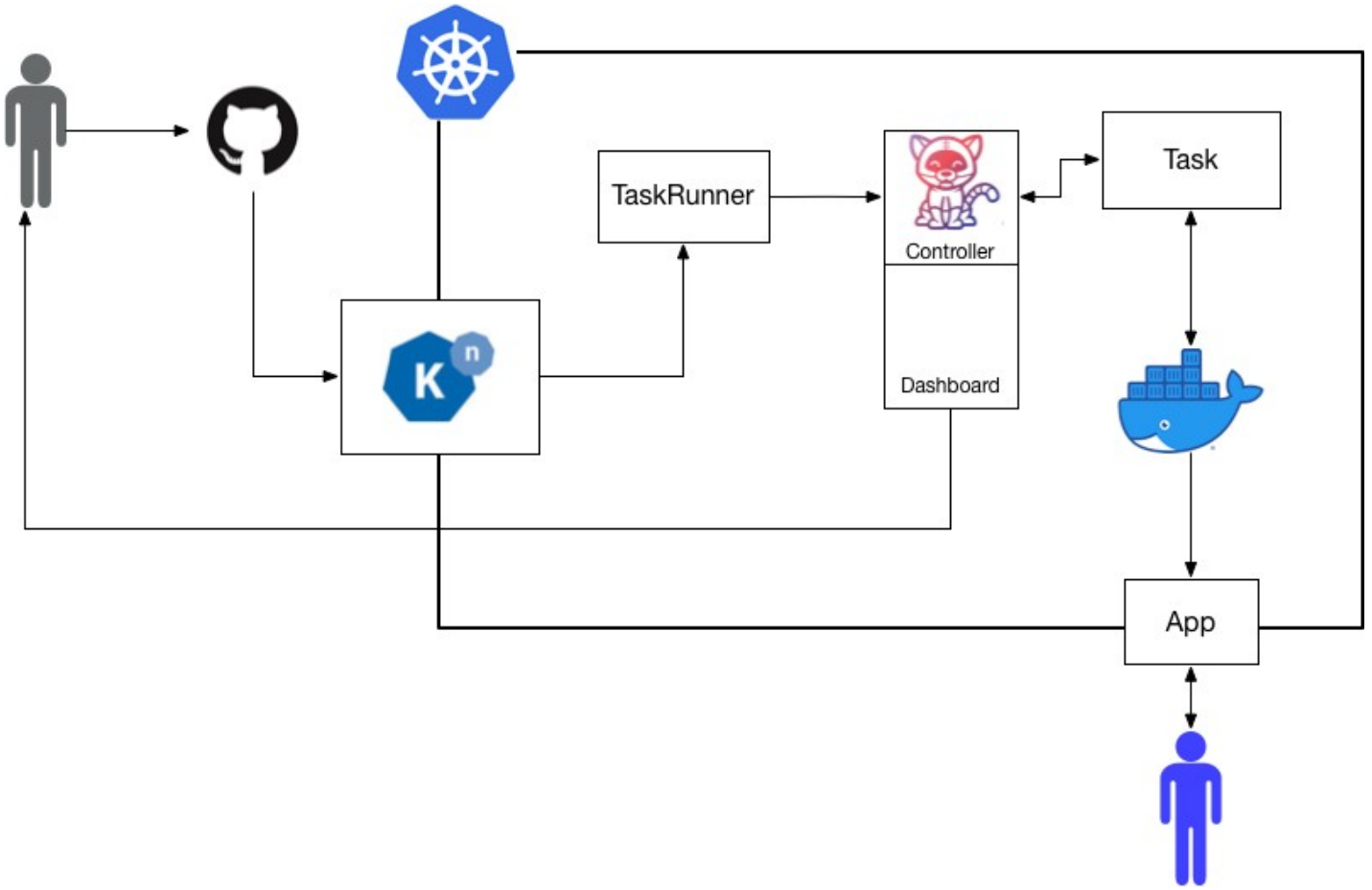
Origin of Tekton: Serverless

- In the beginning there was Kubernetes
  - Kubeless
  - AWS Lambda
  - Knative
- Knative's various components
  - Serving
  - Eventing
  - Building

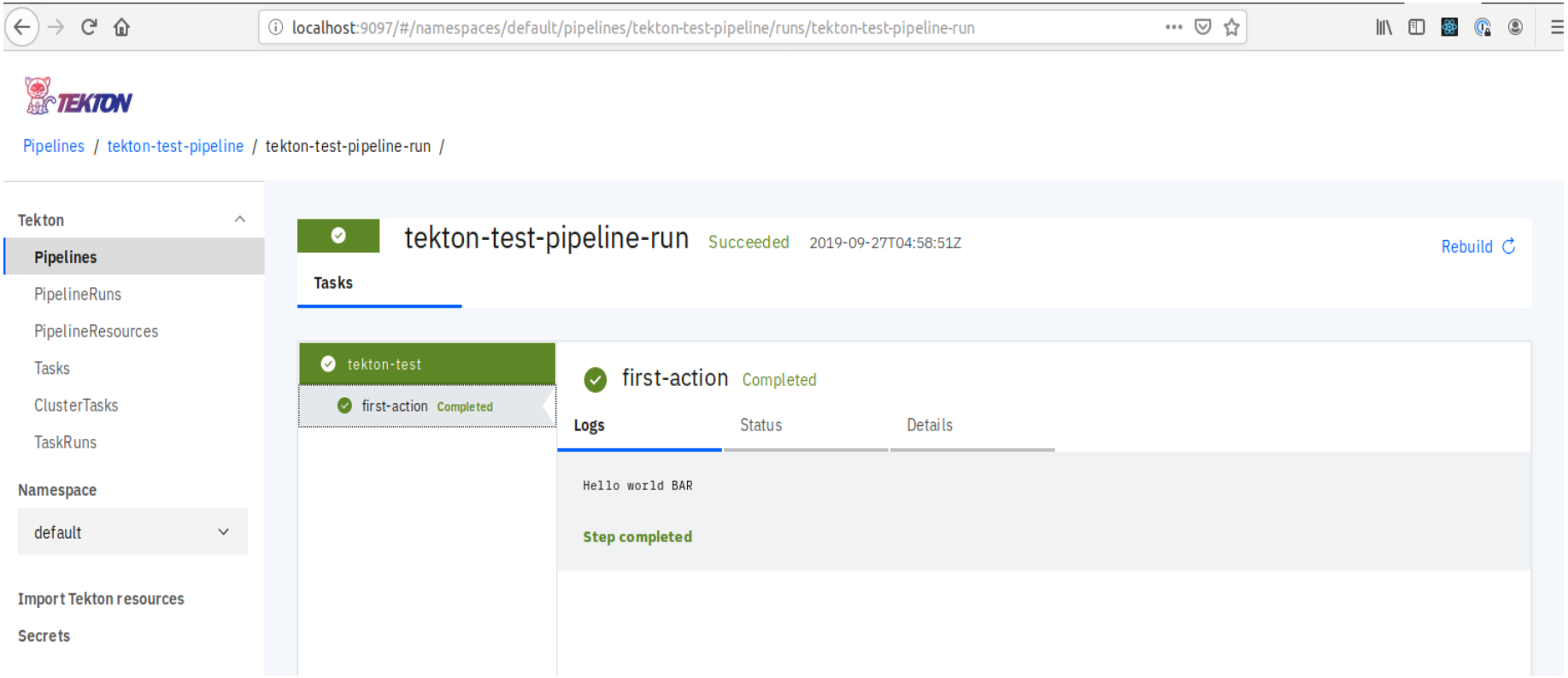
What Is Tekton?

- Originally knative-build
  - Assist with building serverless functions
- k8s Primatives for CI/CD
- Components
  - Dashboard (UI)
  - Pipeline (main component)
  - Triggers
    - Knative Eventing without Knative
    - v0.1 Released Oct 14, 2019

Tekton High Level Overview



Tekton Dashboard



Tekton Pipeline

- Controller
  - Orchestrator of tasks, steps, and resources
- Prerequisites
  - Docker registry
  - Git repo
  - GCS (optional)

Tekton CRDs

- Runners
  - PipelineRun
  - TaskRun
- Pipelines
  - PipelineResource
  - Tasks
- Tasks
  - Steps
- Conditions (new)

Tekton Resources

- PipelineResource and TaskResource are the same concept
- Define an input resource or output resource
- Consists of:
  - Git Repository
  - OCI Image
  - Jar file to be uploaded to GCS

Tekton Tasks (ClusterTask)

- Input/Output resources
- Steps to go from here to there
  - Each step requires it’s own container
  - Think of a cloudbuild step with a dedicated container
- Tasks are namespace scoped, but ClusterTask is global to the cluster
- Sidecars
  - Allow for running DND or mock services

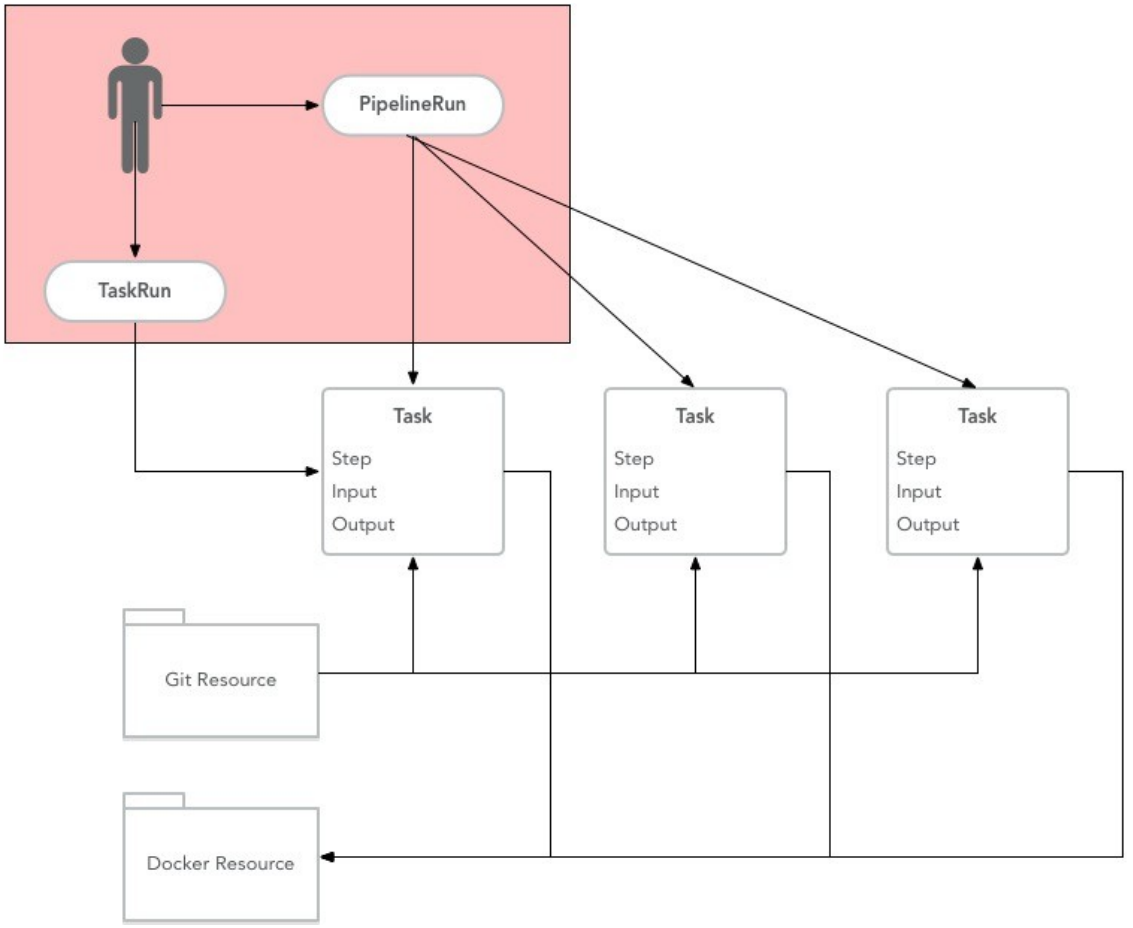
Tekton Pipelines

- Collection of tasks
- Define all resources
- Allow for tying resources to tasks and define task ordering
- Define conditions to indicate if a task should be run in a pipeline

Tekton PipelineRun TaskRun

- One-shot CRD that kicks off a task or pipeline run
- Can modify Input/Output resources depending on context
- Allow for passing in ServiceAccount details
- Specify ‘pod templates’ for Tasks

Tekton Plumbing



Installing Tekton

- Run kubectl to download Tekton Pipeline

```
$ kubectl apply -f \
https://github.com/tektoncd/pipeline/releases/
download/v0.7.0/release.yaml
```

- Run kubectl to download Tekton Dashboard

```
$ kubectl apply -f \
https://github.com/tektoncd/dashboard/releases
/download/v0.1.1/release.yaml
```

Hello World Example

```
apiVersion: tekton.dev/v1alpha1
kind: TaskRun
metadata:
  name: test-taskrun
spec:
  taskRef:
    name: helloTask
```

```
apiVersion: tekton.dev/v1alpha1
kind: TaskResource
metadata:
  name: hello-repo
spec:
  params:
  - name: revision
    value: master
  - name: url
    value: https://github.com/cab105/aktion-test.git
  type: git
```

```
apiVersion: tekton.dev/v1alpha1
kind: Task
metadata:
  name: helloTask
spec:
  steps:
  - args:
    - Hello
    - world
    - $(FOO)
    command:
    - echo
    env:
    - name: FOO
      value: BAR
    image: ubuntu
    name: hello-world
```

Tekton Demo

What Uses Tekton

- Jenkins X
- TriggerMesh Aktion
- Others being added frequently (Tekton Friends)

Jenkins X

- CI/CD Solution around k8s
- Originally a Jenkins sub-project now independent
- Co-lives with Jenkins
- Utilizes a CLI utility (jx) to perform actions

TRIGGERMESH AKTION

RUN GITHUB ACTIONS ON KUBERNETES WITH TEKTON AND KNATIVE

- Convert Github Actions to Tekton Pipelines
- Allow for running Actions outside of Github
- Easier syntax than Tekton’s YAML

Triggermesh Aktion Sample Code

```
---
apiVersion: tekton.dev/v1alpha1
kind: Task
metadata:
  creationTimestamp: null
  name: tekton-test
spec:
  steps:
  - args:
    - Hello
    - world
    - $(FOO)
    command:
    - echo
    env:
    - name: FOO
      value: BAR
    image: centos
    name: first-action
    resources: {}
---
apiVersion: tekton.dev/v1alpha1
kind: Pipeline
metadata:
  creationTimestamp: "2019-09-29T05:09:37Z"
  name: tekton-test-pipeline
spec:
  tasks:
  - name: tekton-test
    taskRef:
      name: tekton-test
status: {}
```

```
workflow "Tekton test" {
  on = "push"
  resolves = [
    "First Action",
  ]
}

# Old syntax
action "First Action" {
  uses = "docker://centos"
  runs = "echo"
  env = {
    FOO = "BAR"
  }
  args = "Hello world $FOO"
}
```

Resources

- Tekton: https://tekton.dev
  - tektoncd.slack.com
- Aktion: https://github.com/triggermesh/aktion
- Jenkins X: https://jenkins.io/projects/jenkins-x/

Thank You