

# 开源技术 \* IBM微讲堂

## Tekton从源码构建容器镜像

张涛

zhangtbj@cn.ibm.com

# Knative系列公开课

每周四晚8点档

- ◆09月19日 《Knative概览》
- ◆09月26日 《Tekton从源码构建容器镜像》
- ◆10月10日 《Knative Serving让容器从另扩展到无限》
- ◆10月17日 《Knative Eventing打造标准的云计算事件平台》
- ◆10月24日 《Knative客户端工具介绍》
- ◆10月31日 《案例分享：在DevOps场景中使用Knative》

# 讲师介绍



## 张涛

IBM 资深工程师

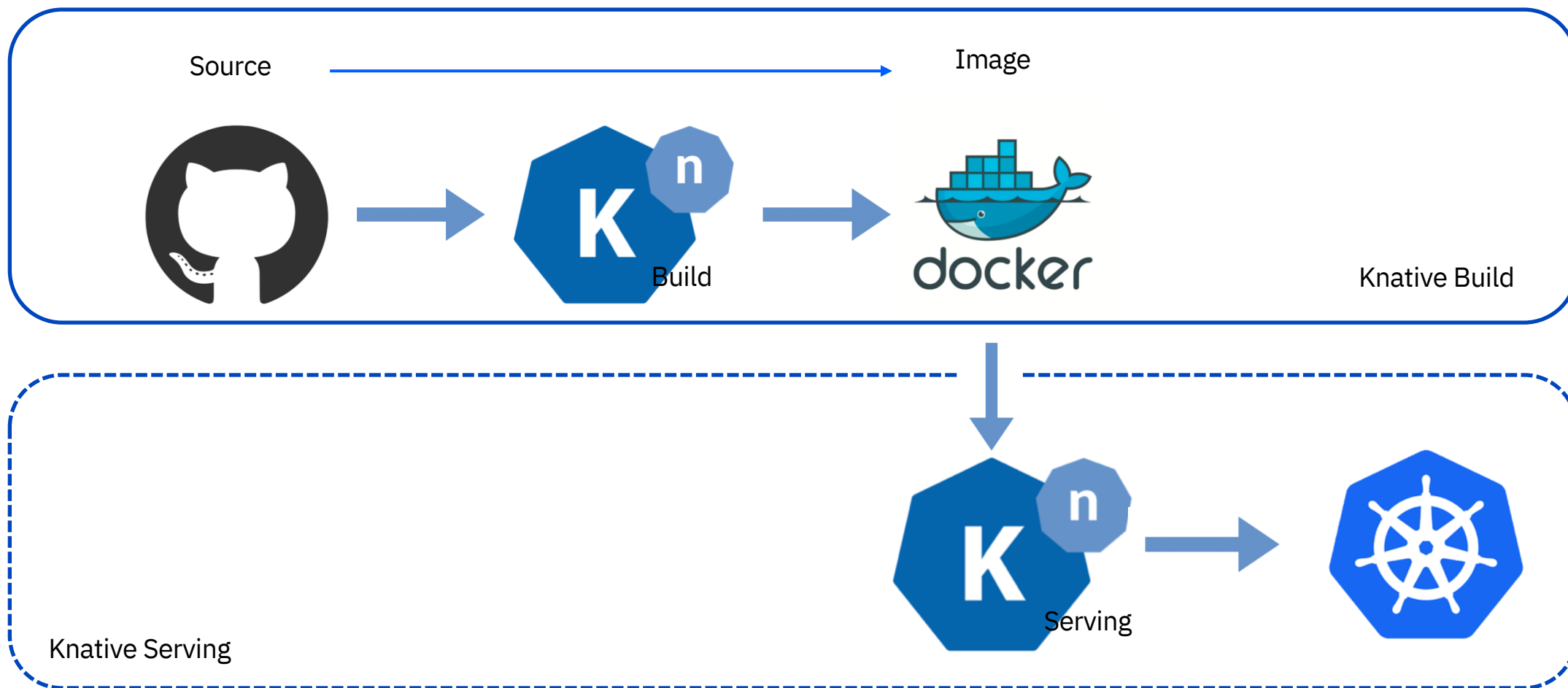
主要负责IBM Cloud Foundry Enterprise Environment服务的构建和开发。近期在从事社区Knative Client和Plugin的开发以及与Tekton整合的相关工作。在这之前主要从事WebSphere中间件的开发工作

# 演讲内容

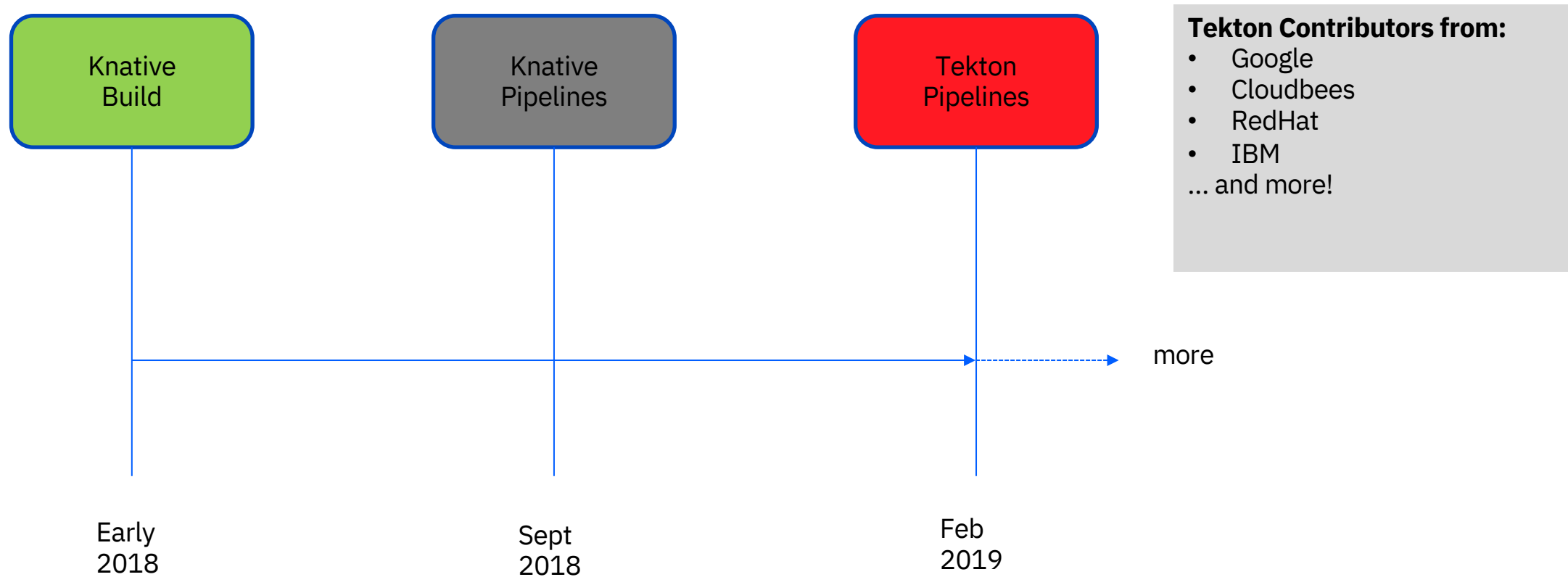
- 1, Knative Build 发展历程
- 2, Knative Build 介绍
- 3, Tekton pipeline 介绍
- 4, Tekton 相关项目介绍



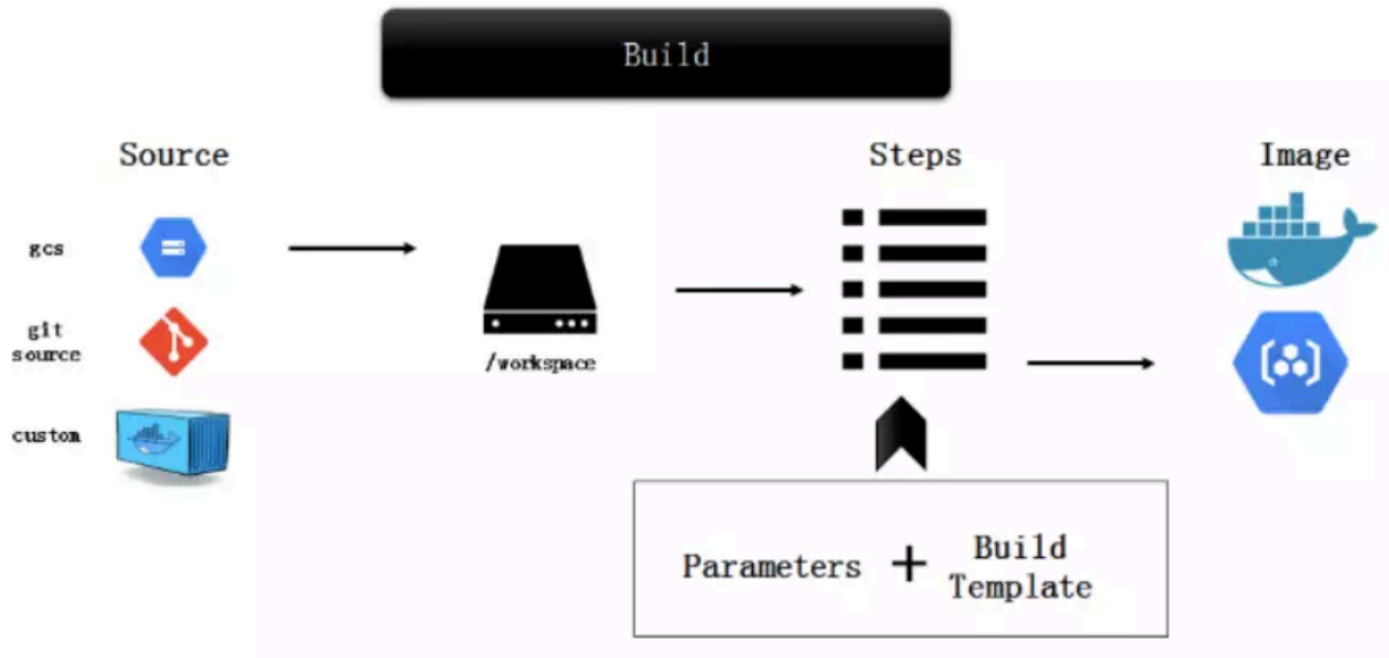
# 1, Knative Build 发展历程



# 1, Knative Build 发展历程



## 2, Knative Build (<https://github.com/knative/build>)



### Deployment:

- build-controller
- build-webhook

### CRD:

- build
- build-template (namespace scope)
- cluster-build-template (cluster scope)

A **Build** is a custom resource in Knative that allows you to define a **process** that runs to completion and can provide status. For example, fetch, build, and package your code by using a Knative **Build** that communicates whether the process succeeds.

**NOTE:** Knative Build is **deprecated** in favor of Tekton Pipelines. There are no plans to produce future releases of this component.

## 2, Knative Build source-to-image example

### 1, build.yaml

```
apiVersion: build.knative.dev/v1alpha1
kind: Build
metadata:
  name: build-helloworld-go
spec:
  serviceAccountName: build-bot
  source:
    git:
      url: "https://github.com/knative/docs.git"
      revision: "v0.4.x"
      subPath: "docs/serving/samples/hello-world/helloworld-go/"
  steps:
    - name: build-and-push
      image: "gcr.io/kaniko-project/executor:latest"
      args: ["--dockerfile=workspace/Dockerfile",
        # Replace {username} with your actual DockerHub
        "--destination=docker.io/{username}/helloworld-go:0.4.x"]
```

### 2, Apply the build

```
kubectl apply -f build-helloworld-go.yaml
```

### 3, Run the build

```
kubectl logs --follow --container=build-step-build-and-push build-helloworld-go-pod-5ecc5b

INFO[0000] Downloading base image golang:1.12
2019/04/04 19:00:37 No matching credentials were found, falling back on anonymous
INFO[0001] Unpacking rootfs as cmd RUN CGO_ENABLED=0 GOOS=linux go build -v -o helloworld requires it.
INFO[0018] Taking snapshot of full filesystem...
INFO[0027] Skipping paths under /kaniko, as it is a whitelisted directory
INFO[0027] Skipping paths under /var/run, as it is a whitelisted directory
INFO[0027] Skipping paths under /builder/home, as it is a whitelisted directory
INFO[0027] Skipping paths under /dev, as it is a whitelisted directory
INFO[0027] Skipping paths under /sys, as it is a whitelisted directory
INFO[0027] Skipping paths under /proc, as it is a whitelisted directory
INFO[0027] Skipping paths under /workspace, as it is a whitelisted directory
INFO[0032] WORKDIR /go/src/github.com/knative/docs/helloworld
INFO[0032] cmd: mkdir
INFO[0032] Changed working directory to /go/src/github.com/knative/docs/helloworld
INFO[0032] Creating directory /go/src/github.com/knative/docs/helloworld
INFO[0032] Taking snapshot of files...
INFO[0032] Using files from context: [/workspace]
INFO[0032] COPY . .
INFO[0032] Taking snapshot of files...
INFO[0032] RUN CGO_ENABLED=0 GOOS=linux go build -v -o helloworld
INFO[0032] cmd: /bin/sh
INFO[0032] args: [-c CGO_ENABLED=0 GOOS=linux go build -v -o helloworld]
net
crypto/x509
net/textproto
internal/x/net/http/httpproxy
internal/x/net/http/httputils
crypto/tls
net/http/httputils
net/http
github.com/knative/docs/helloworld
INFO[0035] Taking snapshot of full filesystem...
INFO[0043] Skipping paths under /kaniko, as it is a whitelisted directory
INFO[0043] Skipping paths under /var/run, as it is a whitelisted directory
INFO[0043] Skipping paths under /builder/home, as it is a whitelisted directory
INFO[0043] Skipping paths under /dev, as it is a whitelisted directory
INFO[0043] Skipping paths under /sys, as it is a whitelisted directory
INFO[0043] Skipping paths under /proc, as it is a whitelisted directory
INFO[0043] Skipping paths under /workspace, as it is a whitelisted directory
INFO[0047] Storing source image from stage 0 at path /kaniko/stages/0
INFO[0066] Deleting filesystem...
INFO[0071] Downloading base image alpine
2019/04/04 19:01:47 No matching credentials were found, falling back on anonymous
INFO[0071] Taking snapshot of full filesystem...
INFO[0081] Skipping paths under /kaniko, as it is a whitelisted directory
INFO[0081] Skipping paths under /var/run, as it is a whitelisted directory
INFO[0081] Skipping paths under /builder/home, as it is a whitelisted directory
INFO[0081] Skipping paths under /dev, as it is a whitelisted directory
INFO[0081] Skipping paths under /sys, as it is a whitelisted directory
INFO[0081] Skipping paths under /proc, as it is a whitelisted directory
INFO[0081] Skipping paths under /workspace, as it is a whitelisted directory
INFO[0081] COPY --from=builder /go/src/github.com/knative/docs/helloworld/helloworld /helloworld
INFO[0081] Taking snapshot of files...
INFO[0081] ENV PORT 8080
INFO[0081] CMD ["/helloworld"]
2019/04/04 19:01:58 pushed blob sha256:ff2ee7cb646b994a7581534a94c543a5f83d251dc744224878c1b5dbdd06150ec
2019/04/04 19:01:58 docker.io/test-user/helloworld-go:0.4.x: digest:
sha256:9dc6a8212ac4e3a7d412b1fcd9d414b438e0d3a5d01725eb5b1972eff9d997 size: 647
```

### 4, Init the build

```
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
build-helloworld-go-pod-5ecc5b	0/1	Init:1/3	0	5s

### 5, Finish the build

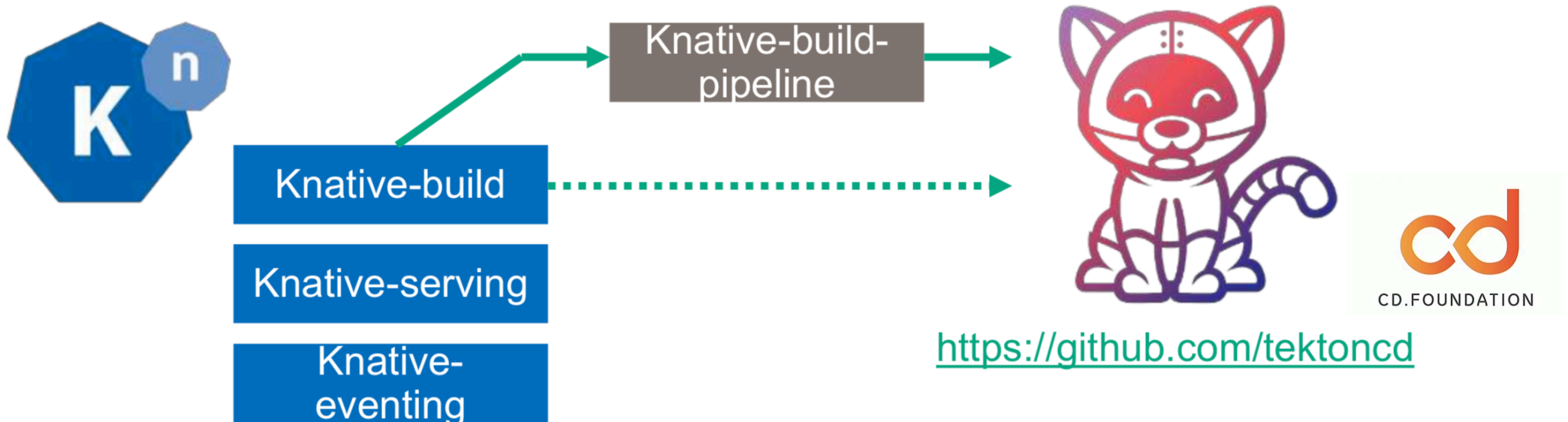
```
kubectl get pods
```

NAME	READY	STATUS	RESTARTS	AGE
build-helloworld-go-pod-5ecc5b	0/1	Completed	0	2m

Knative Build is **not** a comprehensive **CI/CD** solution but allows to offer building blocks that can be **integrated** into existing CI/CD solution.



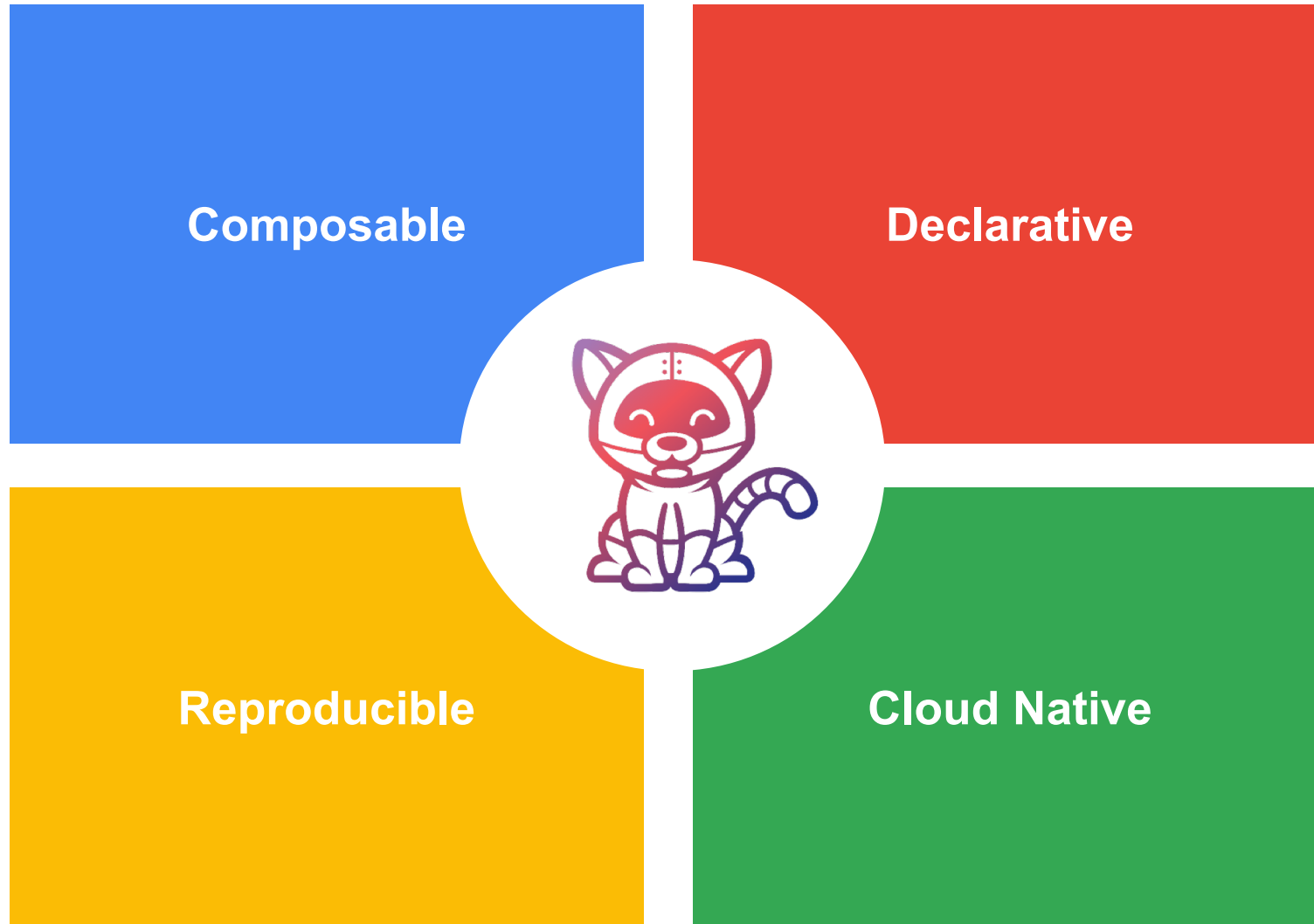
### 3, Tekton pipeline (<https://github.com/tektoncd/pipeline>)



#### **Install Tekton on IKS:**

```
kubectl apply --filename https://storage.googleapis.com/tekton-releases/latest/release.yaml
```

### 3, Tekton pipeline (<https://github.com/tektoncd/pipeline>)



### 3, Knative Build and Tekton pipeline

#### **Knative Build:**

##### Deployments:

- build-controller
- build-webhook

##### CRDs:

- build
- build-template
- cluster-build-template



#### **Tekton Pipeline**

##### Deployments:

- pipeline-controller
- pipeline-webhook

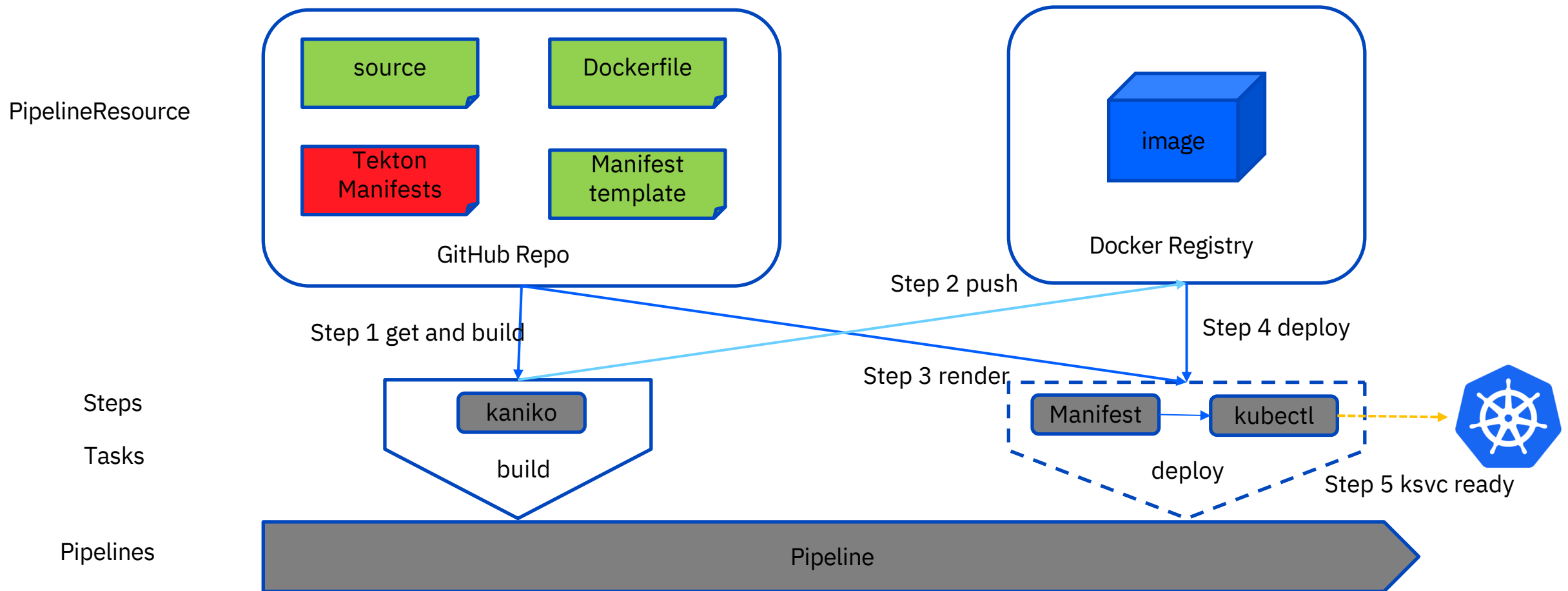
##### CRDs:

- PipelineResource
- Task
- TaskRun
- Pipeline
- PipelineRun
- ClusterTask
- ...

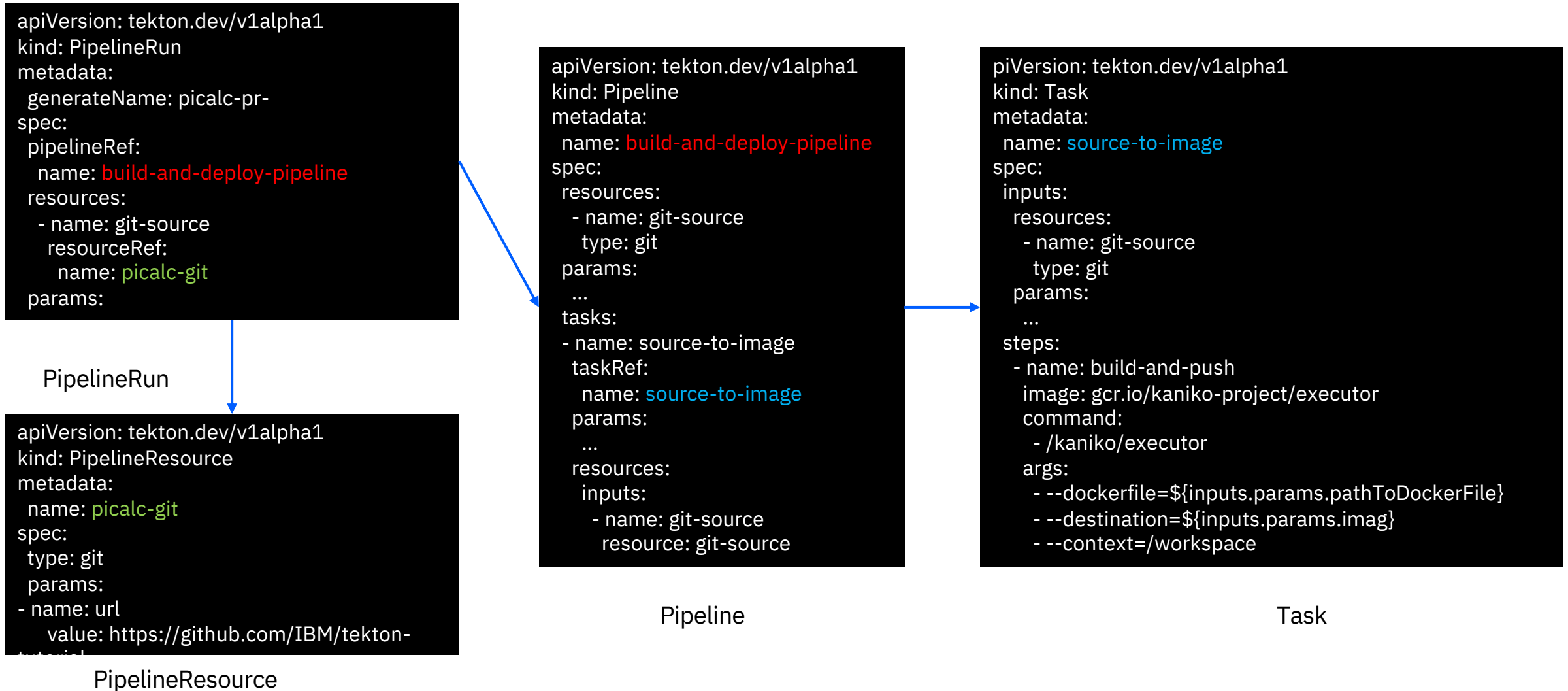
#### **Tekton Addons**

- Dashboard (Web UI)
- Command Line (cli)
- Catalog (shared)

### 3, Tekton pipeline source-to-image example



### 3, Tekton pipeline source-to-image example



# Tekton Demo

**Demo:** pipeline, access app, image, tkn cli, dashboard and catalog

## Deploy a Knative application using Tekton Pipelines on IKS:

<https://developer.ibm.com/tutorials/knative-build-app-development-with-tekton/>

## Tekton Catalog

This repository contains a catalog of `Task` resources (and someday `Pipeline` s and `Resource` s), which are designed to be reusable in many pipelines.

Each `Task` is provided in a separate directory along with a `README.md` and a Kubernetes manifest, so you can choose which `Task` s to install on your cluster.

### Task Kinds

There are two kinds of `Task` s:

1. `ClusterTask` with a Cluster scope, which can be installed by a cluster operator and made available to users in all namespaces
2. `Task` with a Namespace scope, which is designed to be installed and used only within that namespace.

`Task` s in this repo are namespace-scoped `Task` s, but can be installed as `ClusterTask` s by changing the kind .

Catalog Repo: <https://github.com/tektoncd/catalog>

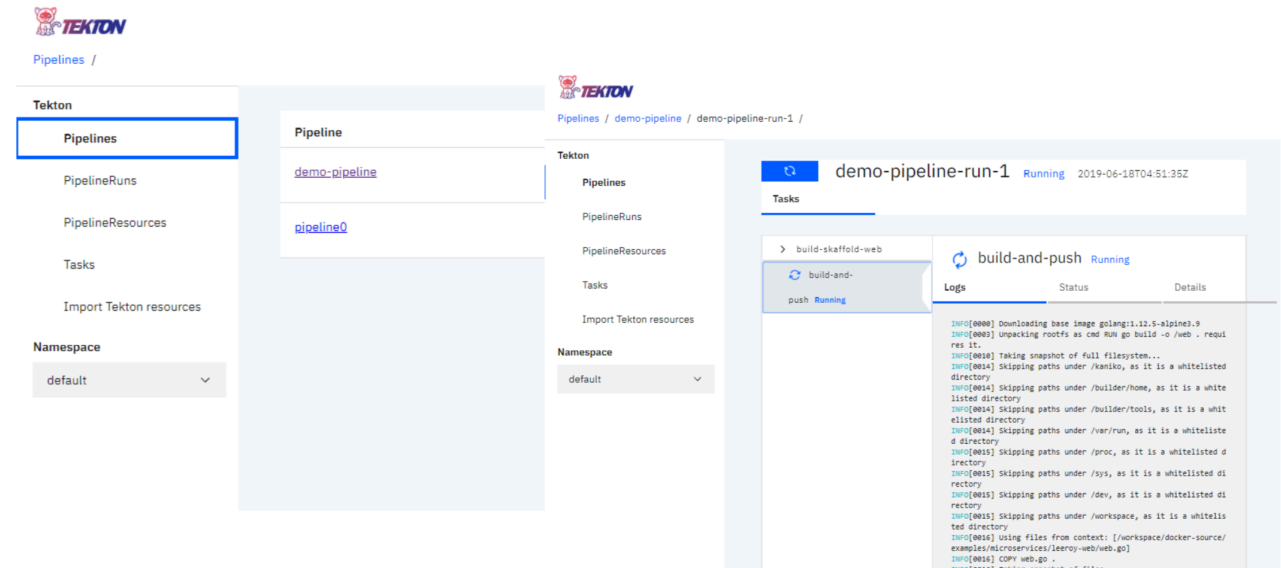
```
✓ $ tkn
Error: Use one of available subcommands: completion [SHELL], help [command], pipeline, pipelinerun, task, taskrun, version
Usage:
  tkn [flags]
  tkn [command]

Available Commands:
  completion  Prints shell completion scripts
  help        Help about any command
  pipeline     Manage pipelines
  pipelinerun Manage pipelineruns
  task        Manage tasks
  taskrun     Manage taskruns
  version     Prints version information

Flags:
  --azure-container-registry-config string  Path to the file containing Azure container registry configuration information.
  -h, --help                                help for tkn

Use "tkn [command] --help" for more information about a command.
```

Cli Repo: <https://github.com/tektoncd/cli>



Dashboard Repo: <https://github.com/tektoncd/dashboard>

## 4, Tekton与Knative client的集成

### Kn client app plugin proposal

```
✓ $ ./app deploy --help
Deploy knative application by building image

Usage:
  app deploy [flags]

Examples:

# Deploy from Git repository into an image
# ( related: https://github.com/maximilien/kn-tkn-plugins/blob/master/plugins/app/doc/deploy-git-resource.md )
app go deploy test -u https://github.com/bluebosh/knap-example -r master

Flags:
  -h, --help                help for deploy
  -n, --namespace string    namespace of app (default "default")
  -r, --revision string     [Git] revision of git repo (default "master")
  -t, --template string     [Git] template of build-to-image task (default "kaniko")
  -u, --url string          [Git] url of git repo

Global Flags:
  --config string    config file (default is $HOME/.app.yaml)
  --kubeconfig string kube config file (default is KUBECONFIG from ENV property)
```

```
jordanzt@zhangtaodembp:~/Works/workspaces/gopath/src/github.com/maximilien/kn-tkn-plugins/plugins/app$ kn app deploy test -u https://github.com/bluebosh/knap-example -r master

[INFO] Building application test image in namespace default
[INFO] From git repo https://github.com/bluebosh/knap-example revision master

[INFO] Git resource test-git exists, updating
[INFO] Git resource test-git created successfully

[INFO] Build task build-to-image exists, updating
[INFO] Build task test-git created successfully

[INFO] Start taskrun test-tr-5-build-to-image-20190901182833 for application test
[INFO] Build taskrun test-tr-5-build-to-image-20190901182833 is still Building , waiting
[INFO] Build taskrun test-tr-5-build-to-image-20190901182833 is still Building , waiting
[INFO] Build taskrun test-tr-5-build-to-image-20190901182833 is still Building , waiting
[INFO] Build taskrun test-tr-5-build-to-image-20190901182833 is still Building , waiting
[INFO] Build taskrun test-tr-5-build-to-image-20190901182833 is still Building , waiting
[INFO] Build taskrun test-tr-5-build-to-image-20190901182833 is still Building , waiting
[INFO] Build taskrun test-tr-5-build-to-image-20190901182833 is ready from 2019-09-01 18:28:33 +0800 CST to 2019-09-01 18:29:36 +0800 CST
[INFO] Complete taskrun test-tr-5-build-to-image-20190901182833 successfully

[INFO] Complete building application test image in namespace default
[INFO] Generate image us.icr.io/knative_jordan/test:5.0 from git repo for application test
[INFO] Deploy application test image in namespace default
[INFO] Knative service test exists, updating
[INFO] Service test is ready
[INFO] Deploy knative service test by using us.icr.io/knative_jordan/test:5.0
[INFO] Service test url is http://test-default.kubecon2019.us-south.containers.appdomain.cloud
```

# 4, Tekton roadmap 2019

## **Version 1.0**

Currently the Pipeline API is considered alpha. In 2019 we aim to release a 1.0 version.

## **Workflow**

Enable more powerful workflows than the simple linear builds we support today. This includes:

- DAG pipelines
- Conditional execution
- Cancelling or pausing a workflow
- Resuming a paused or failed workflow
- Enforcing timeouts on Tasks and on pipelines
- Sending notifications
- ...

## **Triggering**

Automating a workflow means automating how those workflows are started. We need to design support for automatically kicking off pipelines based on events.

## **Security**

Security requirements inform much of the overall Tekton pipelines design, but we still have lots of work to do.

.... (Future plan like Machine Learning Ops)





**谢谢!**

# Links

Knative Build git repo: <https://github.com/knative/build>

Tekton git repo: <https://github.com/tektoncd/pipeline>

Knative client git repo: <https://github.com/knative/client>