

Tekton 初体验

阳明



Who am I?



阳明

- 半自由职业
- K8S技术圈社区维护者
- 优点知识站长



TEKTON

Kubernetes-native CI/CD

Documentation →

GitHub ↻

Agenda



- Tekton 是什么
- Tekton 的安装
- Tekton 的组件
- Tekton 使用示例

为什么需要 CI/CD



- **DevOps** 集文化理念、实践和工具于一身，可以提高组织高速交付应用程序和服务的能力
- 能够帮助组织更快地发展和改进产品，这种速度使组织能够更好服务其客户，并在市场上更高效地参与竞争
- **CI** 代表持续集成，**CD** 代表持续交付和持续部署
- 将软件开发生命周期的整个过程都自动化
- 为了使整个软件开发流程处于 DevOps 模式，我们就需要对 CI/CD 流水线进行自动化



Tekton 是什么

Tekton 是一款 Kubernetes 原生的应用发布框架，主要用来构建 CI/CD 系统。

它原本是 knative 项目里面一个叫做 build-pipeline 的子项目，用来作为 knative-build 的下一代引擎，随着社区里各种各样的需求涌入，这个子项目就慢慢成长为一个通用的框架了，能够提供灵活强大的能力去做基于 k8s 的构建发布。



安装 Tekton



直接使用官方的仓库 tektoncd/pipeline 中的资源清单安装即可，下面是镜像上传到 DockerHub 的版本

\$ kubectl apply -f <https://www.qikqiak.com/k8strain/devops/manifests/tekton/release.yaml>

此外也可以安装 Tekton CLI 来管理资源(我这里是 Mac 系统)

\$ brew tap tektoncd/tools

\$ brew install tektoncd/tools/tektoncd-cli

安装 Tekton Dashboard

\$ kubectl apply -f <https://www.qikqiak.com/k8strain/devops/manifests/tekton/dashboard.yaml>

安装 Tekton

```
$ kubectl get pods -n tekton-pipelines
```

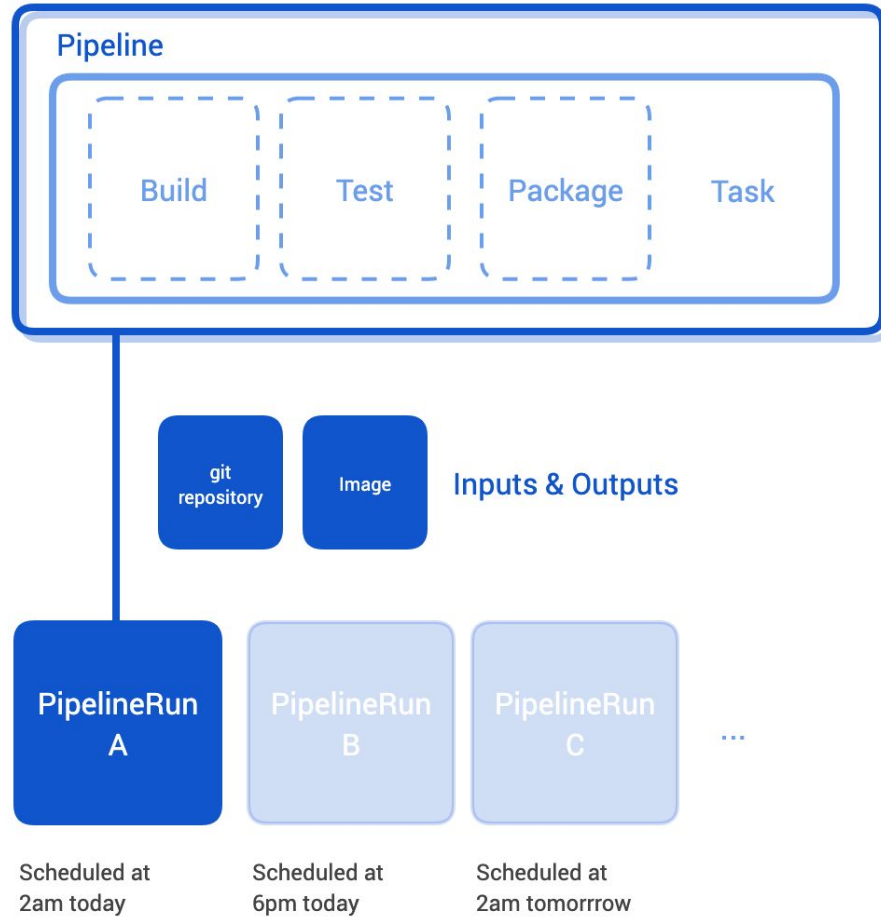
NAME	READY	STATUS	RESTARTS	AGE
tekton-dashboard-69656879d9-7bbkl	1/1	Running	0	11d
tekton-pipelines-controller-67f4dc98d8-pgxrq	1/1	Running	1	30d
tekton-pipelines-webhook-59df55445c-jw76v	1/1	Running	1	30d

The screenshot displays the Tekton dashboard interface. On the left, a sidebar menu lists various resources: Tekton resources (Pipelines, PipelineRuns, PipelineResources, Tasks, ClusterTasks, TaskRuns), Namespace (default), About, Import Tekton resources, and Kubernetes resources (Secrets, ServiceAccounts). The main panel shows a specific pipeline run named 'test-build-push-run' which occurred 18 days ago and is in a 'Succeeded' state. It indicates that 2 tasks were completed and 0 were skipped. A list of steps is shown: 'test', 'run-test' (which is completed), and 'build-and-push'. The 'run-test' step is selected, and its logs are displayed, showing a successful execution with the message 'PASS ok' and a duration of 0.008s.

Tekton 核心对象



- **Task**: 表示执行命令的一系列步骤，task 里可以定义一系列的 steps。
- **ClusterTask**: 覆盖整个集群的任务，而不是单一的某一个命名空间。
- **TaskRun**: task 只是定义了一个模版，taskRun 才真正代表了实际的运行，taskRun 创建出来之后，就会自动触发 task 描述的任务。
- **Pipeline**: 一组任务，表示一个或多个 Task、PipelineResource 以及各种定义参数的集合。
- **PipelineRun**: 类似 task 和 taskRun 的关系，pipelineRun 也表示某一次实际运行的 pipeline。
- **PipelineResource**: 表示 pipeline 输入输出资源，比如 git 的代码或者镜像。



Tekton Demo

test-build-push-run 10 days ago

Rerun ↺

Succeeded Tasks Completed: 2, Skipped: 0 📄

✓ build-and-push

✓ build-and-... Completed

✓ test

✓ build-and-push Completed

Logs

Status

Details

Container status:

```
name: build-and-push
container: step-build-and-push
imageID: >-
  docker-pullable://cnych/kaniko-executor@sha256:39e4b829f93ee2de85050575f7e25111016c5280
terminated:
  containerID: 'docker://9cb45a6024bc8e1a122648c702e2303f42e7c6de1daaaebecc045b5d84ff7956'
  exitCode: 0
  finishedAt: '2020-05-27T01:11:04Z'
  reason: Completed
  startedAt: '2020-05-27T01:06:24Z'
```

演示示例代码和资源清单文件仓库: <https://github.com/cnych/tekton-demo>

高级概念



- Parameter
- Workspace
- Pipeline Controller Metrics
- Trigger
- ...

查看官方文档: <https://tekton.dev/docs/>

直播用户限时优惠（仅限今日）



GitLab CI 实践课
79元



Kubernetes 训练营
免费参加第二期课程
1198元



Jenkins 实践课
168元