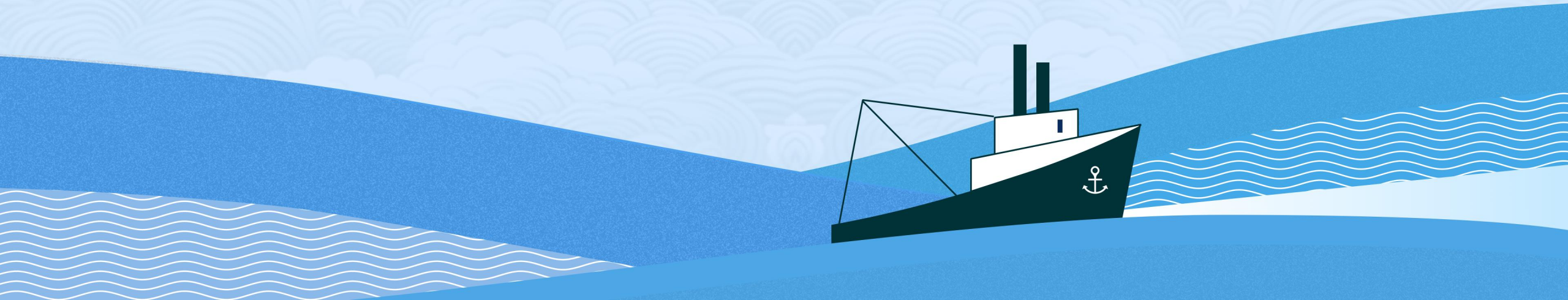
KUBERNETES COMMUNITY DAYS BEIJING 2023

KUBERNETES
COMMUNITY DAYS | BEIJING

KWOK 低成本模拟集群

张世明 DaoCloud



KUBERNETES
COMMUNITY DAYS | BEIJING

Content

目录

KUBERNETES
COMMUNITY DAYS | BEIJING

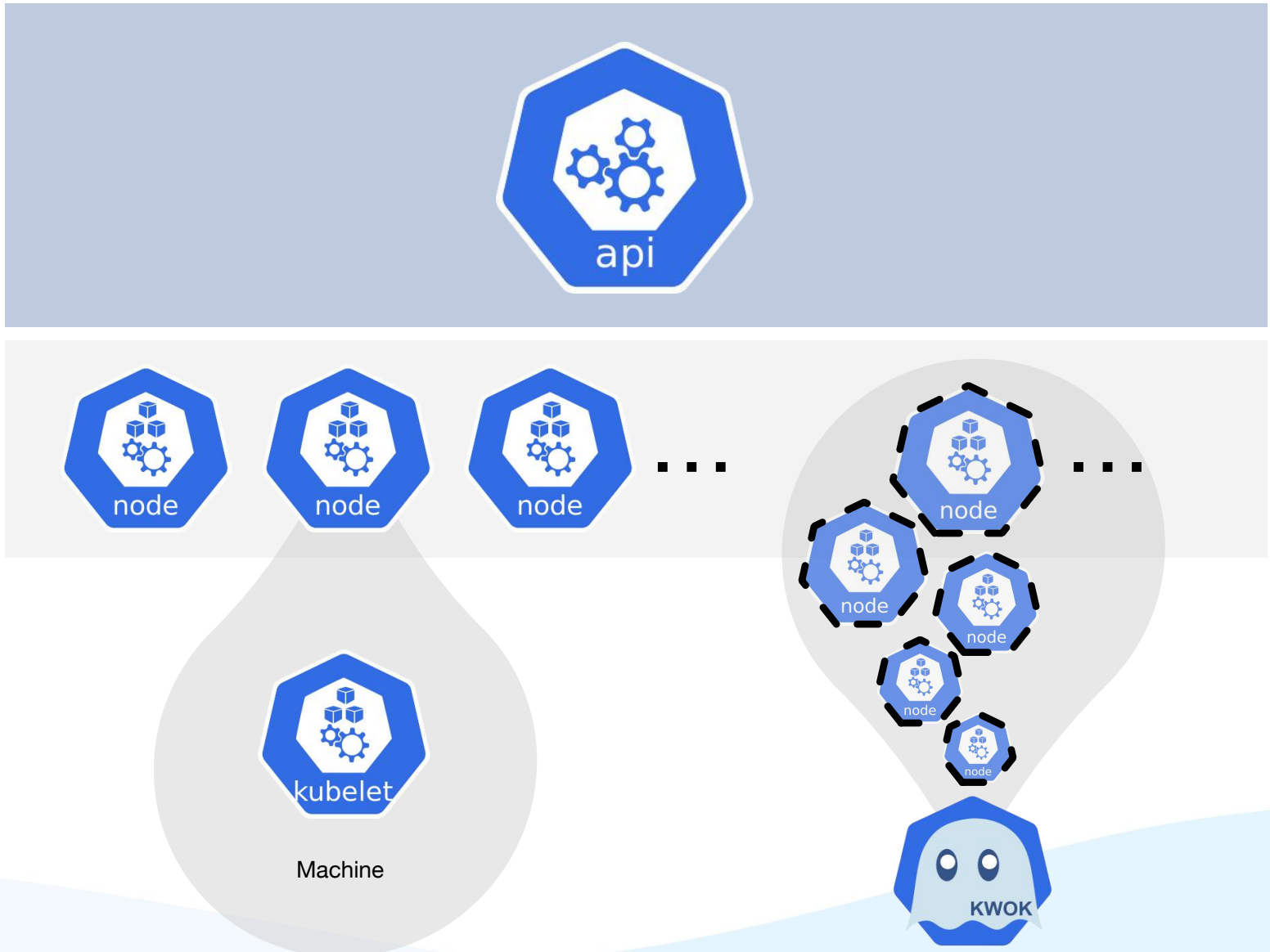
Part 01

What is KWOK

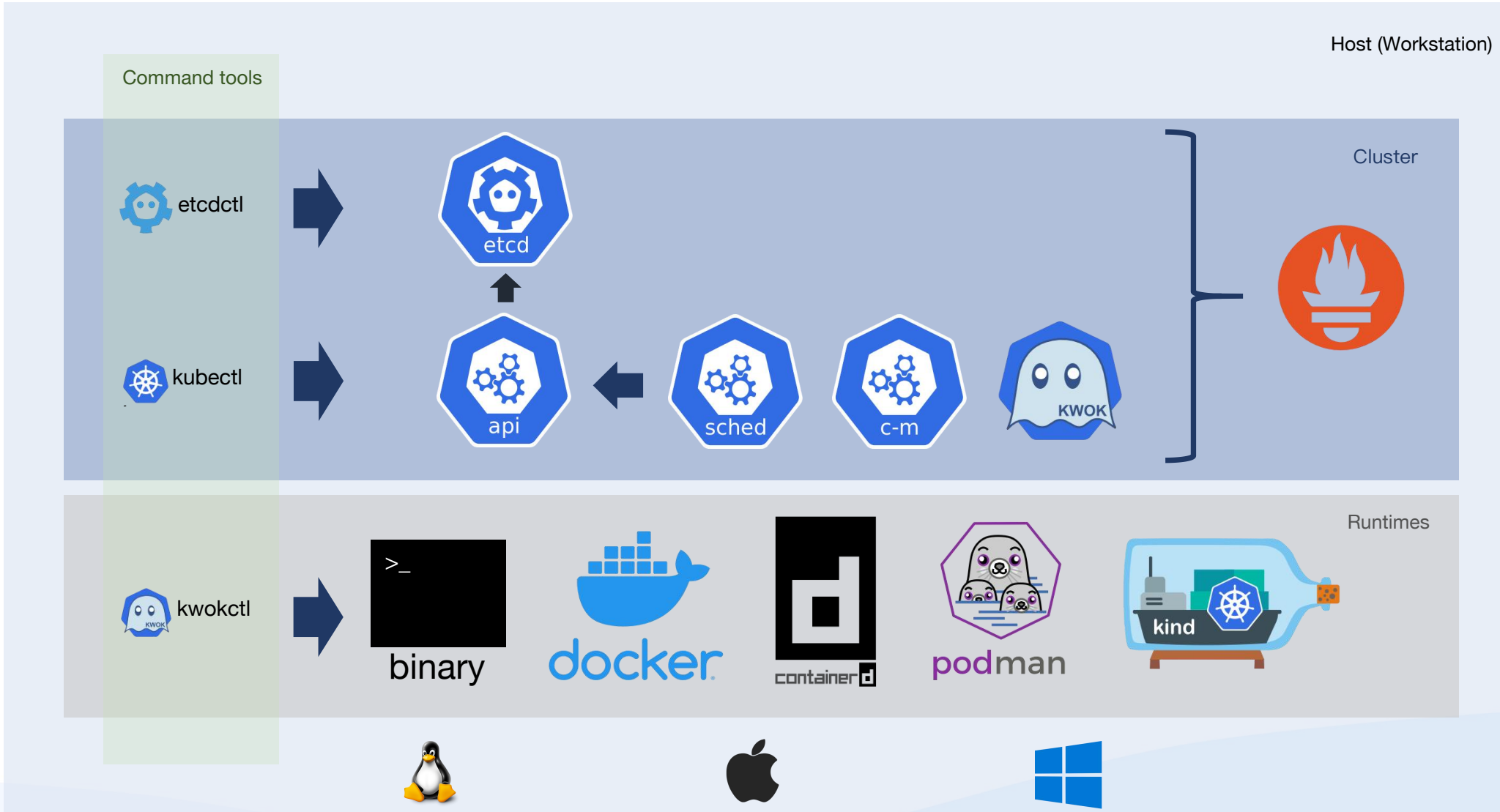
KWOK

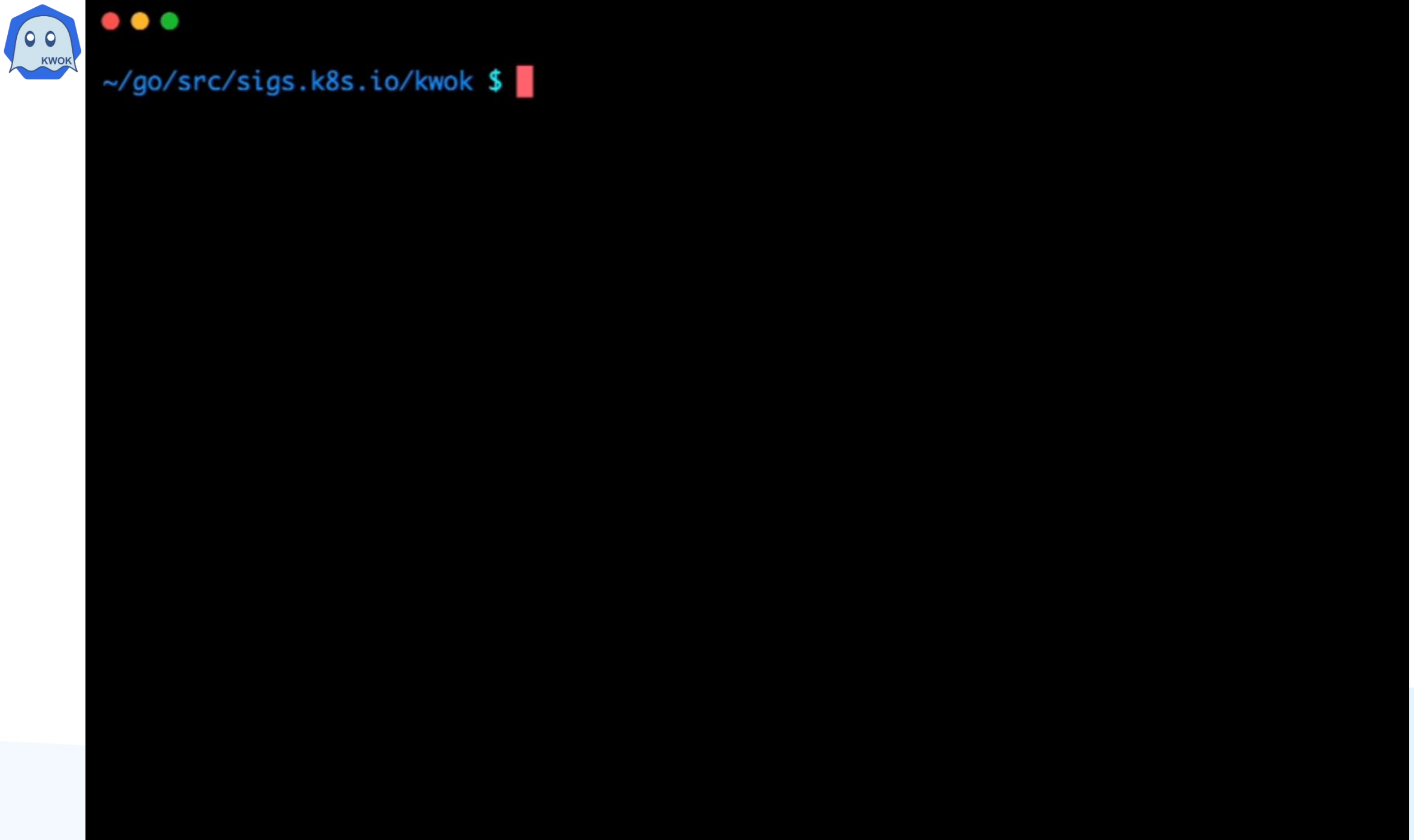
ubernetes ith ut ubelet

- **K**ubernetes **W**ith**O**ut **K**ubelet
- **kwok (kwok-controller)**
  - 模拟任意数量的 Node/Pod, 无实际节点和进程, 零成本
  - 可以自定义 Node/Pod 的生命周期
- **Kwokctl**
  - 创建集群只需几秒 ���


api
node
node
node
node
node
node
node
node
node
kubelet
Machine
KWOK

kwokctl


Host (Workstation)
Command tools
etcdctl
kubectl
kwokctl
etcd
api
sched
c-m
KWOK
Cluster
Runtimes
>_
binary
docker
container
podman
kind


KWOK
~/go/src/sigs.k8s.io/kwok $

KUBERNETES
COMMUNITY DAYS | BEIJING

Part 02

Why is KWOK

预期支持的场景

Mock 数据

- 支撑云管平台 demo 站
- 各种 Apiserver 兼容的 GUI

测试

- Controller e2e
- 边缘case模拟

压测

- Scheduler
- Apiserver
- CRD Controller

KWOK vs Kubemark

	KWOK	Kubemark
兼容性	不兼容 Kubelet 的行为，只做 API 层面看得见的	API 层面行为和 Kubelet 一模一样
维持 Node	一个进程可以维持任意数量	同 Kubelet 一样每个进程维持一个
维持 Pod	一个进程可以维持任意数量	受到 Kubemark 所在机器 内存限制
Pod Status	可以配置	是一个固定值
Node Status	可以配置	是一个固定值
易用性	一份 yaml 直接安装/一行命令直接创建	需要自己编译和修改固定值
资源消耗	极低	只比 Kubelet 少了容器的消耗
适用场景	控制面大规模压测/所有调度场景的测试	偏向 API 层面的兼容性测试或小规模压测

使用 kwok 模拟(只 kwok 自身)

- 不管多少 Node 和 Pod 都小于 100M

使用 Kubemark 模拟(单单创建节点)

- 1个节点 60M
- 5000 个节点 300G

- KWOK 可以模拟一个集群的控制面
  - 尽量模拟真实情况, 但是并不完全相同
- KWOK 不能替代 Kubelet 的测试
  - 不能替代真实的集群

KUBERNETES
COMMUNITY DAYS | BEIJING

Part 03

How to use KWOK

Kensei Nakada
@sanposhiho
kube-schduler-simulator Owner
当前就职于 Mercar, JP

kubernetes-sigs / kube-scheduler-simulator Public

generated from kubernetes/kubernetes-template-project

Watch 14 | Fork 98 | Star 543

Code | Issues 33 | Pull requests 6 | Actions | Projects | Security | Insights

Integrate kwok into simulator #251

New issue

Open sanposhiho opened this issue on Nov 22, 2022 · 11 comments

sanposhiho commented on Nov 22, 2022 • edited — Member

In the simulator, we just run almost pure kube-apiserver. The simulator doesn't have much controllers, and something seems actually weird to users sometimes. (like some fields in Pods and Nodes don't get changed, etc)
And, for example, #243 directly affects the scheduling results, which we must avoid as a simulator.

I feel we can almost resolve them by integrating Kwok which watches Node and Pods and behaves like kubelet. We can expect Pods and Nodes to behave more like a real cluster but without running any real instances or containers.

/kind feature
/triage accepted
/area simulator
/priority important-longterm

2

k8s-ci-robot added kind/feature triage/accepted area/simulator priority/important-longterm labels on Nov 22, 2022

sanposhiho mentioned this issue on Nov 22, 2022

Node's `.status.allocatable` should be changed for NodeResourceFit plugin #243 — Closed

Assignees
sivchari

Labels
area/simulator kind/feature priority/important-longterm triage/accepted

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

Notifications — Customize
Unsubscribe
You're receiving notifications because you were mentioned.

https://github.com/kubernetes-sigs/kube-scheduler-simulator/issues/251

Vladimir Vivien
@vladimirvivien
e2e-framework Owner
当前就职于 VMware, USA

kubernetes-sigs / e2e-framework Public

generated from kubernetes/kubernetes-template-project

Watch 14 | Fork 64 | Star 279

Code | Issues 15 | Pull requests 9 | Actions | Projects | Security | Insights

Support for using Kubernetes-SIGs Kwok cluster simulator #214

New issue

Open vladimirvivien opened this issue 3 weeks ago · 1 comment

vladimirvivien commented 3 weeks ago — Member

Proposal

kwok is a Kubernetes cluster simulator, from Kubernetes-SIGs, that can start fully Kubernetes API-compatible cluster. This project should investigate how e2e-framework can leverage Kwok for end-to-end testing. Currently, the project already has environment functions that can start/stop Kind. It would be useful to create environment functions to start/configure/stop kwok clusters at test time for quickstart end-to-end tests.

The following is an illustrative mock up of what that support could look like. This example would create a simulated cluster, run the tests, then teardown that cluster.

```
func TestMain(m *testing.M) {
        testenv = env.New()
        clusterName := envconf.RandomName("some-cluster", 16)
        namespace := envconf.RandomName("kind-ns", 16)

        testenv.Setup(
                envfuncs.CreateKwokCluster(kindClusterName),
                envfuncs.CreateNamespace(namespace),
        )

        testenv.Finish(
                envfuncs.DeleteNamespace(namespace),
                envfuncs.DestroyKwokCluster(kindClusterName),
        )
        os.Exit(testenv.Run(m))
}
```

Assignees
No one assigned

Labels
kind/feature

Projects
None yet

Milestone
No milestone

Development
No branches or pull requests

Notifications Customize
Subscribe
You're not receiving notifications from this thread.

2 participants

https://github.com/kubernetes-sigs/e2e-framework/issues/214

John Howard
@howardjohn
Istio Owner/ Top 1 Contributor
当前就职于 Google, Sunnyvale, CA

howardjohn / pilot-load Public

Watch 2 | Fork 17 | Starred 29

Code | Issues 1 | Pull requests | Actions | Projects | Security | Insights

Explore kwok #9

New issue

Open howardjohn opened this issue on Feb 6 · 0 comments

howardjohn commented on Feb 6 — Owner

https://github.com/kubernetes-sigs/kwok

might allow us to remove a lot of our logic

❤️ 1

Assignees
No one assigned

Labels
None yet

Projects

https://github.com/howardjohn/pilot-load/issues/9

Sunyanan Choochotkaew
@sunya-ch
当前就职于 IBM Research, JP

Sunyanan Choochotkaew

Feb 16 · 9 min read · Listen

Going Beyond Limits: Scalability Test CI for Kubernetes CNI Operator with Simulated Cluster

CNI (Container Network Interface) is a framework to configure Linux container network interfaces for Kubernetes *Pods* . It requires putting an executable file implementing the CNI to every Node. To do that plus enabling pod-to-pod communication, most CNI projects adopt *operator framework* to have a controller do the magic. When the cluster becomes larger, the controller can become a bottleneck. That's why we need scalability test CI in CNI operator development.

https://medium.com/@sunyanan.choochotkaew1/going-beyond-limits-scalability-test-ci-for-kubernetes-cni-operator-with-simulated-cluster-ed53e772dfa5

大规模集群仿真模拟与调度器压测方法

发布于2023-01-18 12:11:50　　阅读 291

星辰算力平台基于深入优化云原生统一接入和多云调度，加固容器运行态隔离，挖掘技术增量价值，平台承载了腾讯内部的CPU和异构算力服务，是腾讯内部大规模离线作业、资源统一调度平台。

背景

在大规模 Kubernetes 集群中，集群瞬息万变，每时每刻可能都有相关用户、集群组件、运维人员对集群进行操作。根据大规模集群的注意事项，Kubernetes v1.26 单个集群支持的最大节点数为 5000。更具体地说，Kubernetes 旨在适应满足以下所有标准的配置：

- 每个节点的 Pod 数量不超过 110
- 节点数不超过 5000
- Pod 总数不超过 150000
- 容器总数不超过 300000

在这样大规模的集群下，通常我们需要压测各类组件来保障集群在突发状况（如高峰时间段）下的性能和可靠性。对于 apiserver、etcd 这类基础组件，我们只要将服务启动后，可以非常容易地进行压测，如通过 clusterloader2 并发创建大量的请求等方式。但针对调度器，我们却需要一个含有大量节点的集群进行模拟测试，但通常情况下很难短时准备如此多的空闲节点，且测试时对节点资源也是一种浪费。

万幸的是，调度器是负责将 Pod 调度到合适的 Node 上，并不关心后续 Pod 的生产过程。如果能够在集群中虚拟出大量的 Node，就可以完成大规模集群的模拟环境搭建。碰巧，Kubernetes 社区开源的新项目 KWOK(Kubernetes WithOut Kubelet) 为我们带来了解决方案。本文将阐述如何快速模拟大规模测试环境（你甚至可以在自己的 minikube 上搭建），并简要给出调度器的压测结果。同时，由于我们在生产环境中大量使用拓扑感知调度功能并已经贡献至 Crane 开源社区，本文中的压测环境也是基于带有拓扑感知调度增强功能的 crane-scheduler。

- 大规模集群的注意事项：https://kubernetes.io/zh-cn/docs/setup/best-practices/cluster-large/
- clusterloader2：https://github.com/kubernetes/perf-tests/blob/master/clusterloader2/docs/design.md
- KWOK：https://github.com/kubernetes-sigs/kwok
- Crane 开源社区：https://gocrane.io/
- crane-scheduler：https://github.com/gocrane/crane-scheduler

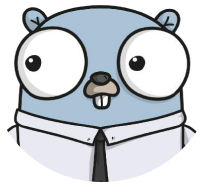
作者介绍

腾讯云原生团队

腾讯云原生

关注　专栏

文章	阅读量	获赞	作者排名
525	326K	3.8K	20



Garry Fang
@Garrybest
当前就职于 腾讯

https://cloud.tencent.com/developer/article/2210383

加入我们

- #kwok on the Kubernetes slack.k8s.io
- Repo: sigs.k8s.io/kwok
- Docs: kwok.sigs.k8s.io

- 微信群:

KUBERNETES
COMMUNITY DAYS | BEIJING

Thanks.