

使用 KusionStack 构建更高效更安全的开发者平台

李大元、吴珂
蚂蚁集团 PaaS 核心

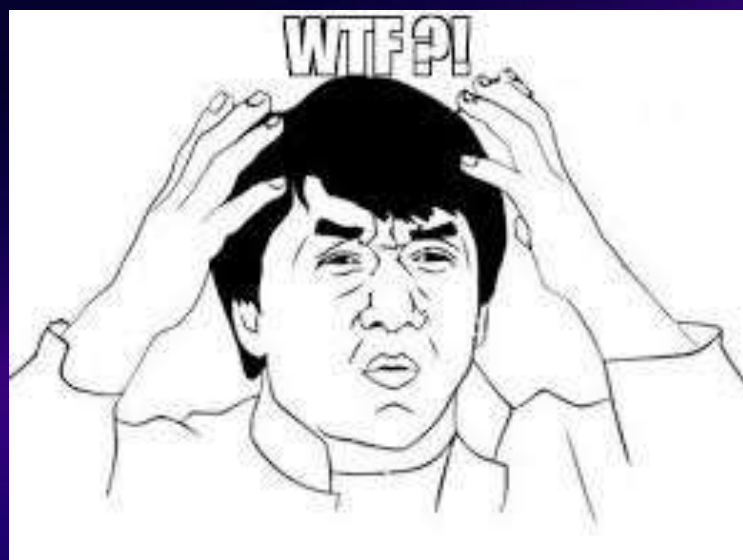
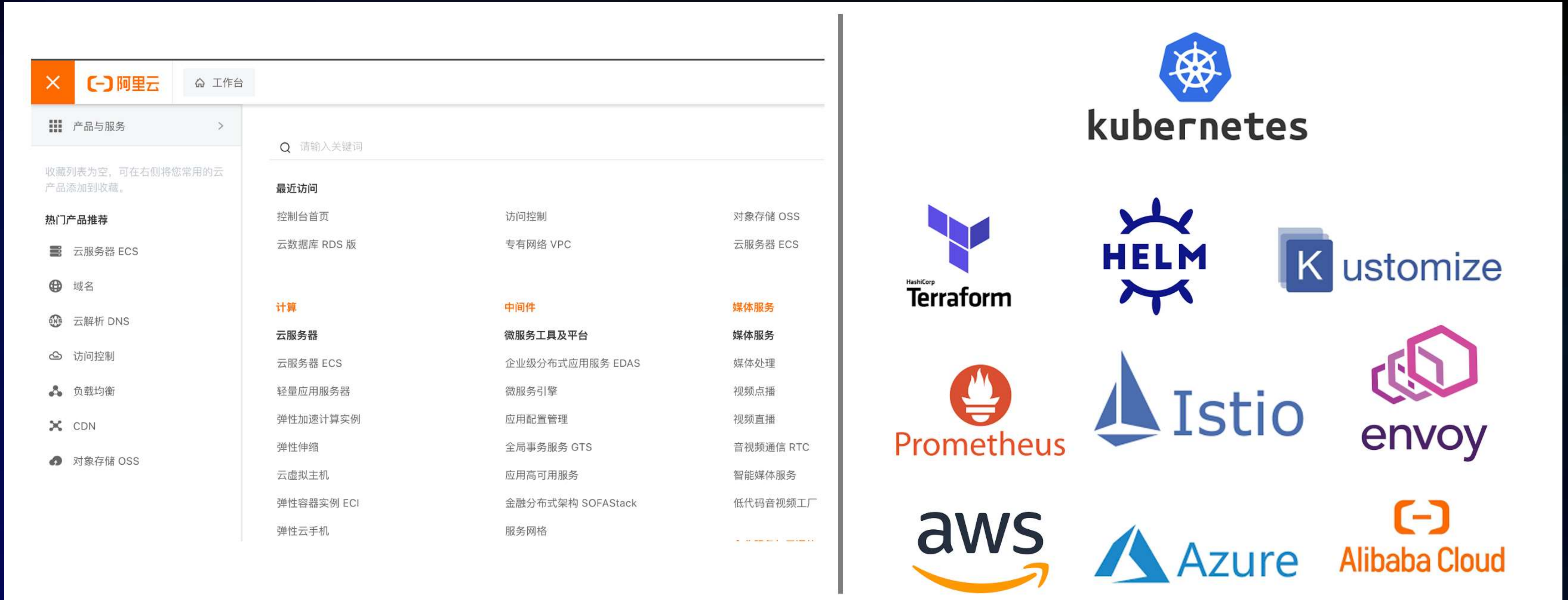
Contents 目录

01 | 云原生时代的运维挑战

02 | KusionStack 介绍

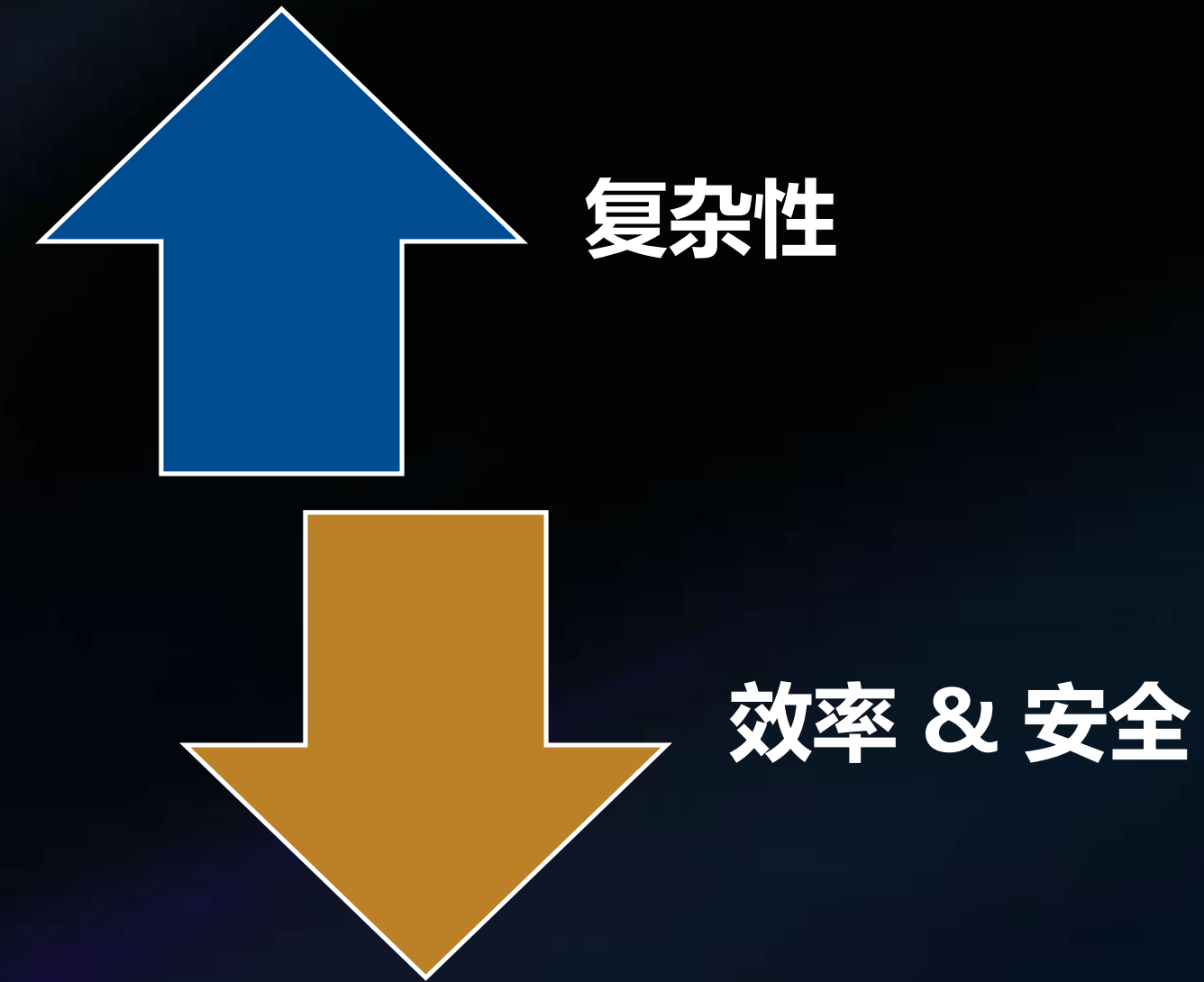
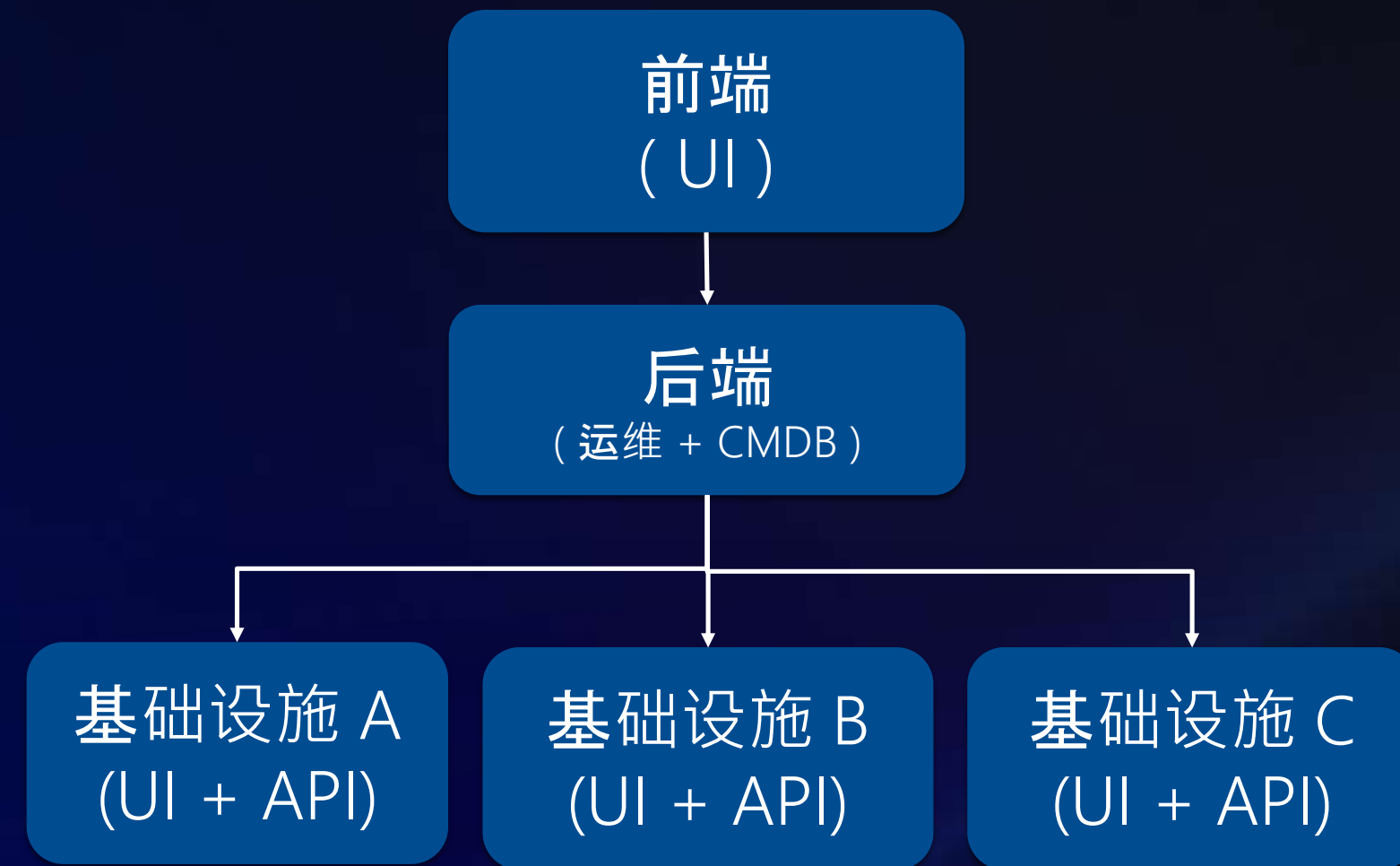
03 | 案例：云原生资源交付与运维

云原生时代的运维挑战



数不清的控制台，数不清的工具

云原生时代的运维挑战



Developer

- 认知负担高
- 平台满足需求时间变长

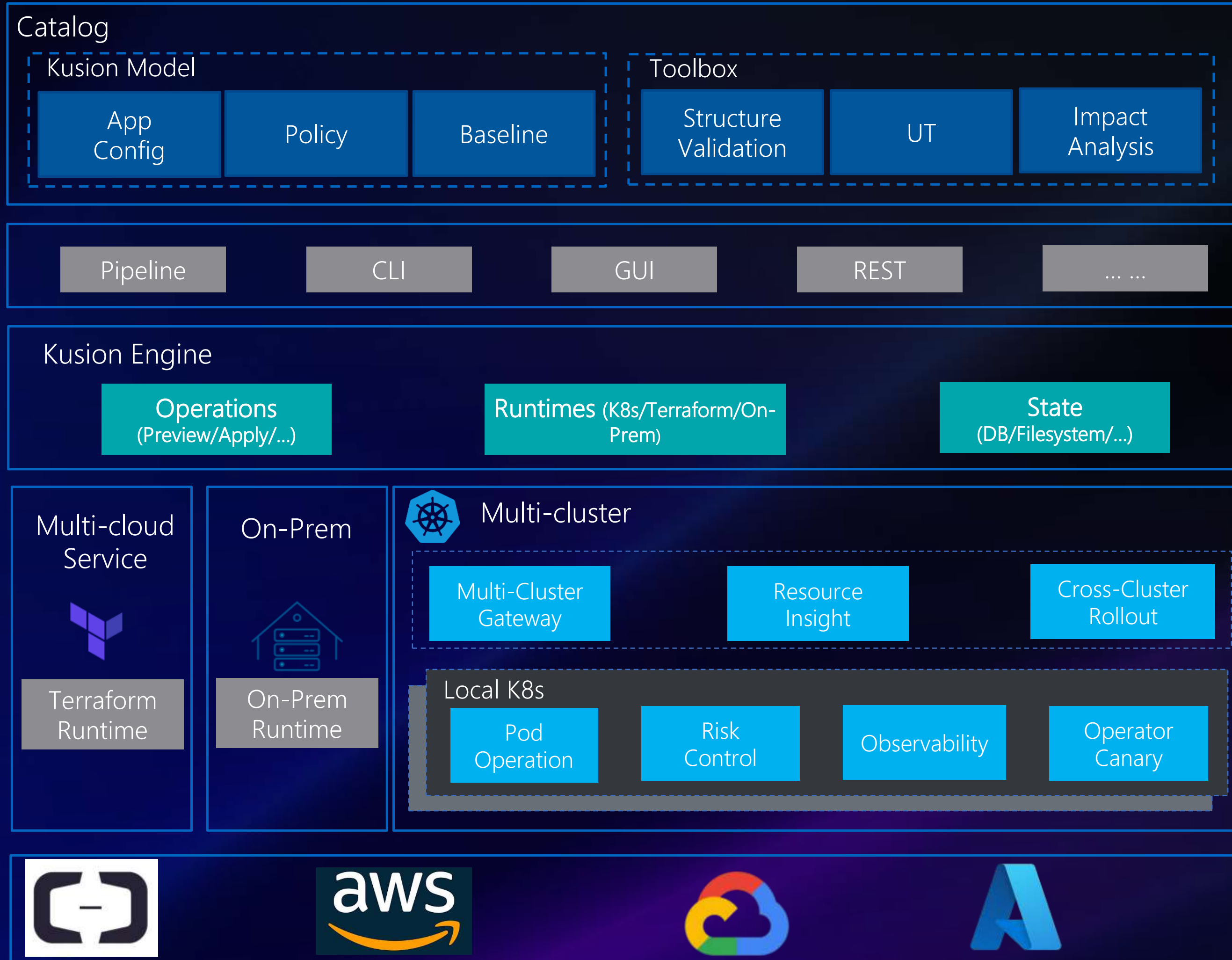
Platform

- 需求的种类与数量大幅增加
- 与基础设施团队沟通成本大幅增加

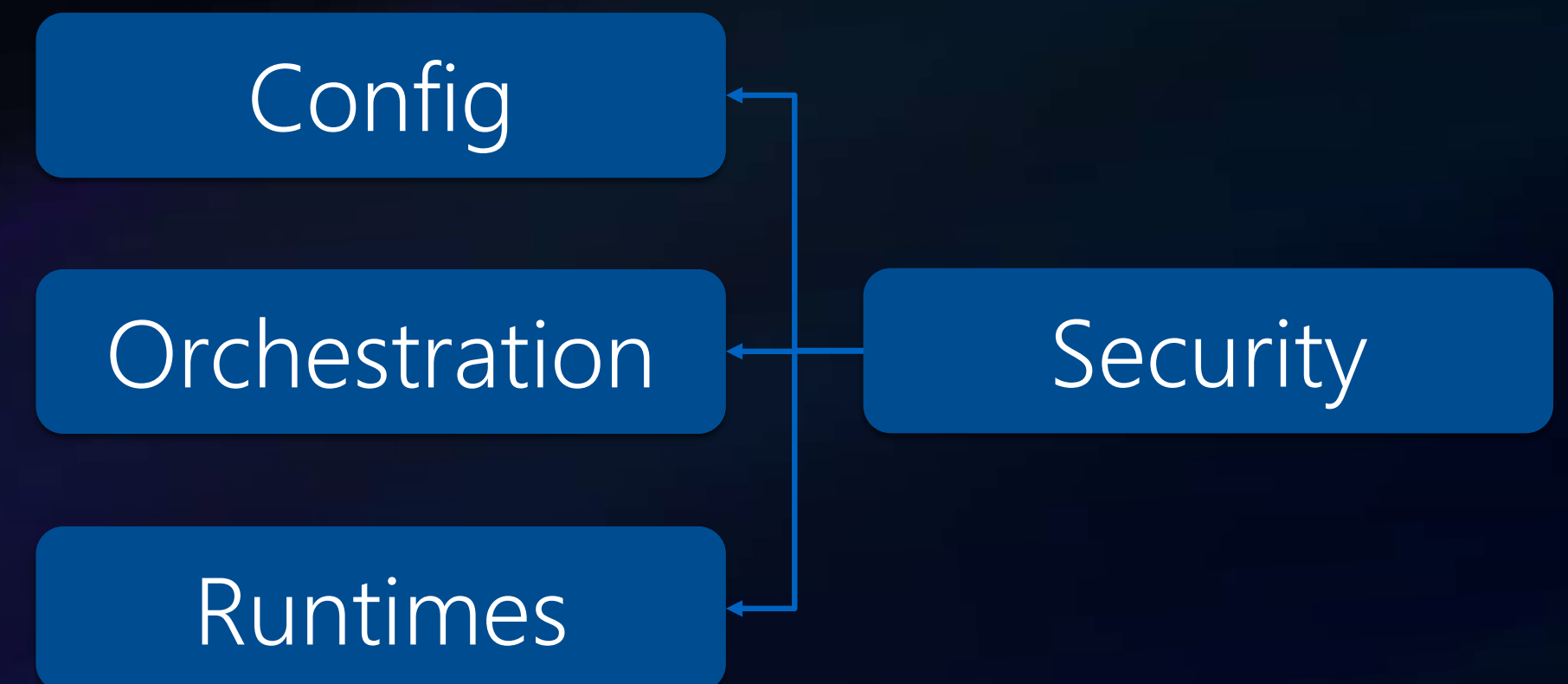
Security

- 多个独立基础设施平台，风险敞口大
- 声明式运维很强大也很危险

KusionStack 架构

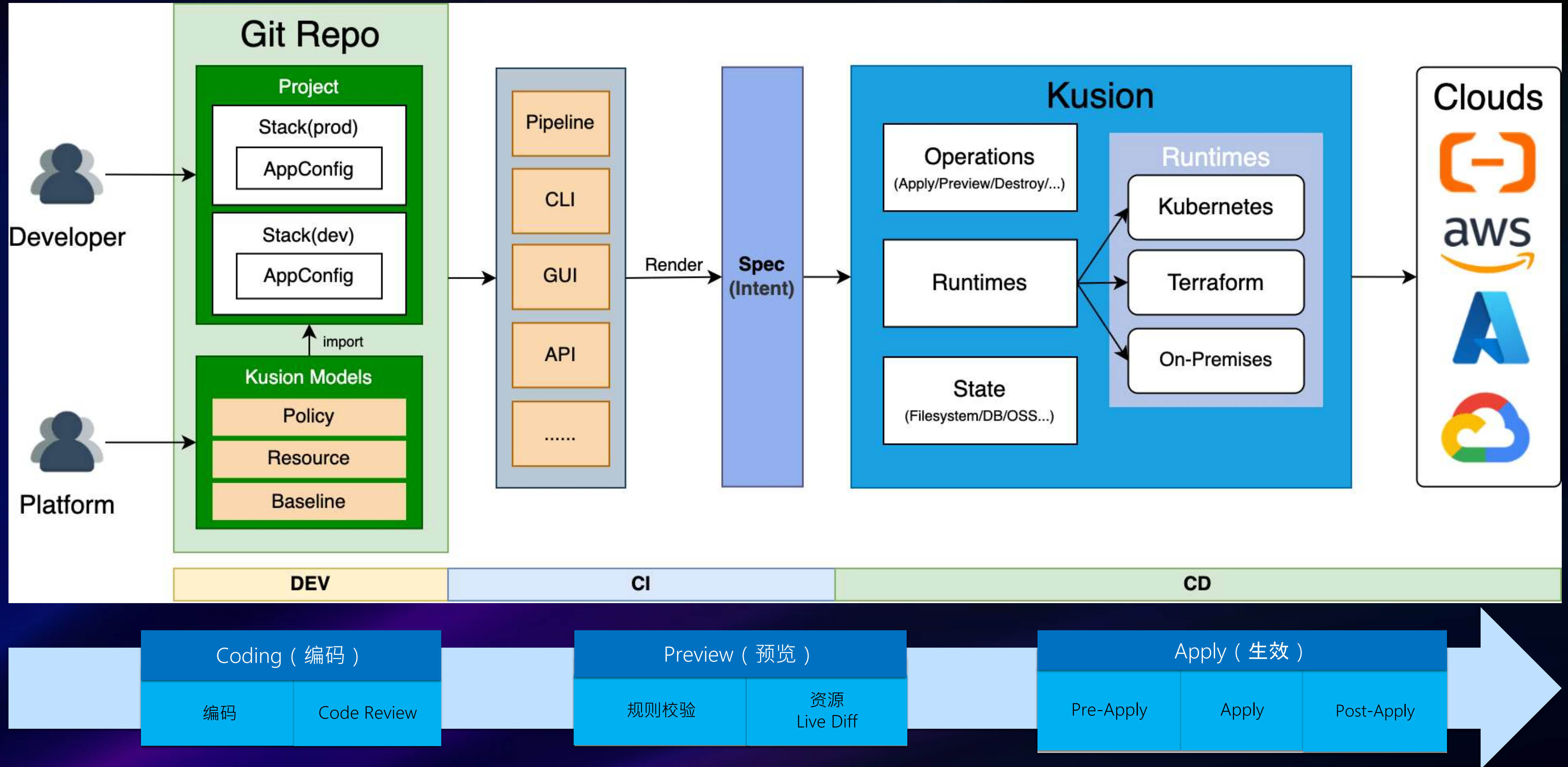


助你 **更快、更安全** 的构建
Developer Platform



即将开源

KusionStack workflow



应用配置

AppConfiguration

- 面向**开发者**的应用界面，屏蔽底层复杂性，降低开发者认知
- 一份配置描述应用所有依赖，覆盖应用**全生命**周期

Topologies

定义应用的拓扑架构，包括多站点多环境多集群等信息



PolicySets

定义应用交付过程需要遵守的安全合规、技术风险策略，减少错配导致的风险



```
1  helloworld: ac.AppConfiguration {
2      workload: wl.Service {
3          replicas: 2
4          containers: {
5              "nginx": c.Container {
6                  image: "nginx:v1"
7                  command: ["/bin/sh", "-c", "echo hi"]
8                  env: {
9                      "key": "value"
10                 }
11                 workingDir: "/tmp"
12                 resources: {
13                     "cpu": "2"
14                     "memory": "4Gi"
15                 }
16                 readinessProbe: p.Probe {
17                     probeHandler: p.Http {
18                         url: "http://localhost:80"
19                     }
20                 }
21             }
22         }
23         ports: [
24             n.Port {
25                 port: 80
26                 targetPort: 8080
27                 exposeInternet: True
28             }
29         ]
30     }
31     topologies: {
32         "ZoneA": Topology {
33             cells: ["GZ99S", "RZ99K"]
34         }
35         "ZoneB": Topology {
36             cells: ["GZ20W"]
37         }
38     }
39     pipeline: {
40         "deploy": Deploy {
41             manualApprove: true
42             rollbackIfFailed: true
43         }
44     }
45     policysets: {
46         "release-strategy-check": PolicySet {
47             kind: "mandatory"
48         }
49         "resource-specification": PolicySet {
50             kind: "advisory"
51         }
52     }
53     dependency: {
54         dependedApps: ["api-server"]
55     }
56 }
```

Components

描述完整交付应用所需的组件，包括工作负载 (workload) 和网络端口、数据库等配件 (accessory)

nginx container

ports database accessory

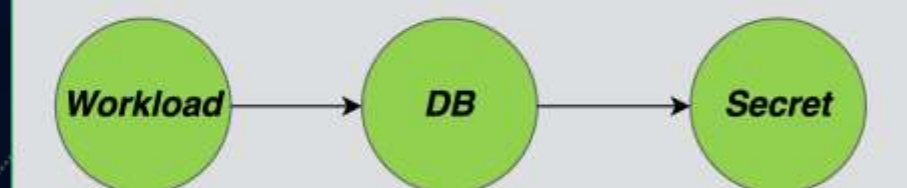
Pipeline

描述应用交付的流程，包括前置检查、审批、部署、后置校验等步骤

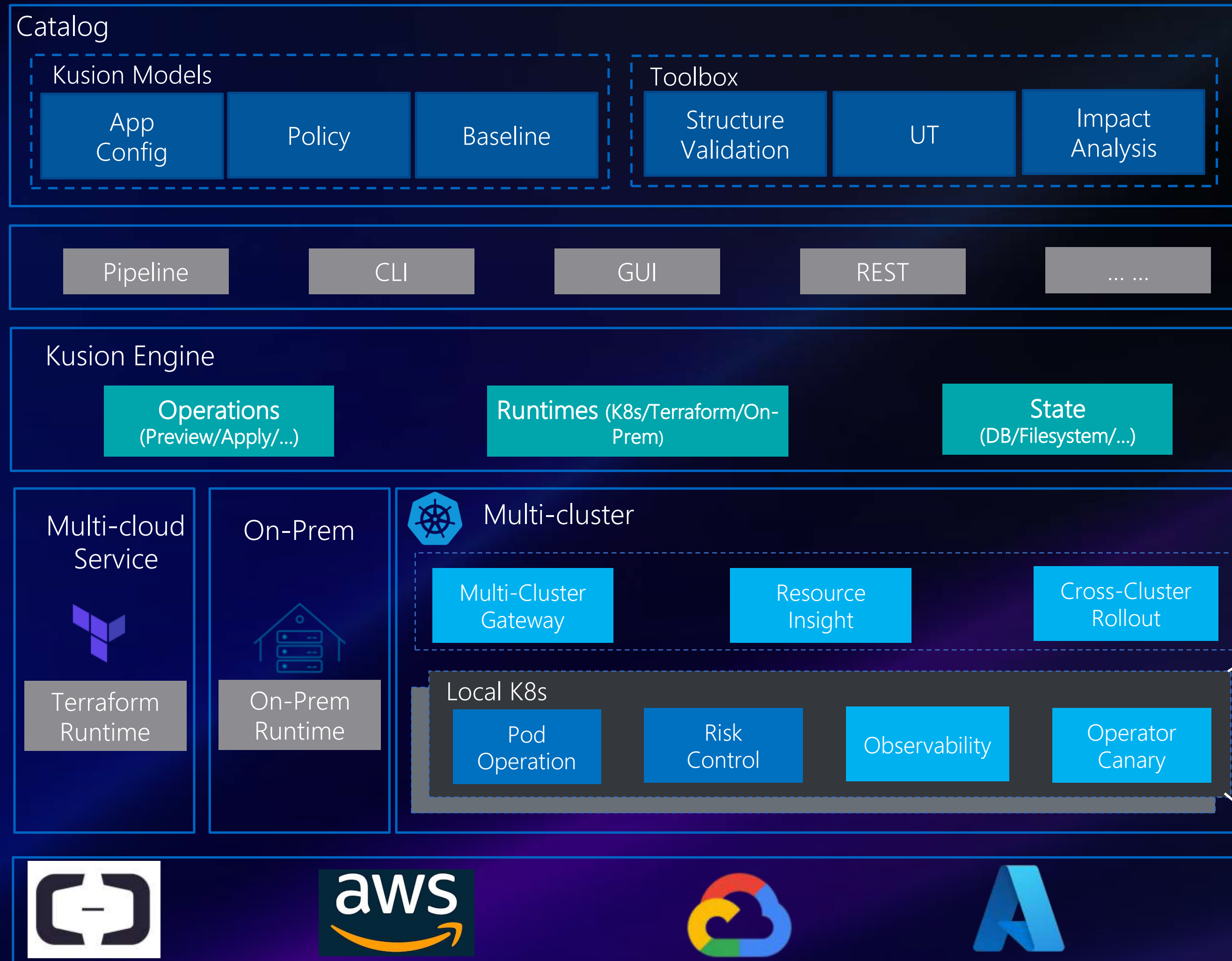


Dependency

描述跨应用依赖关系

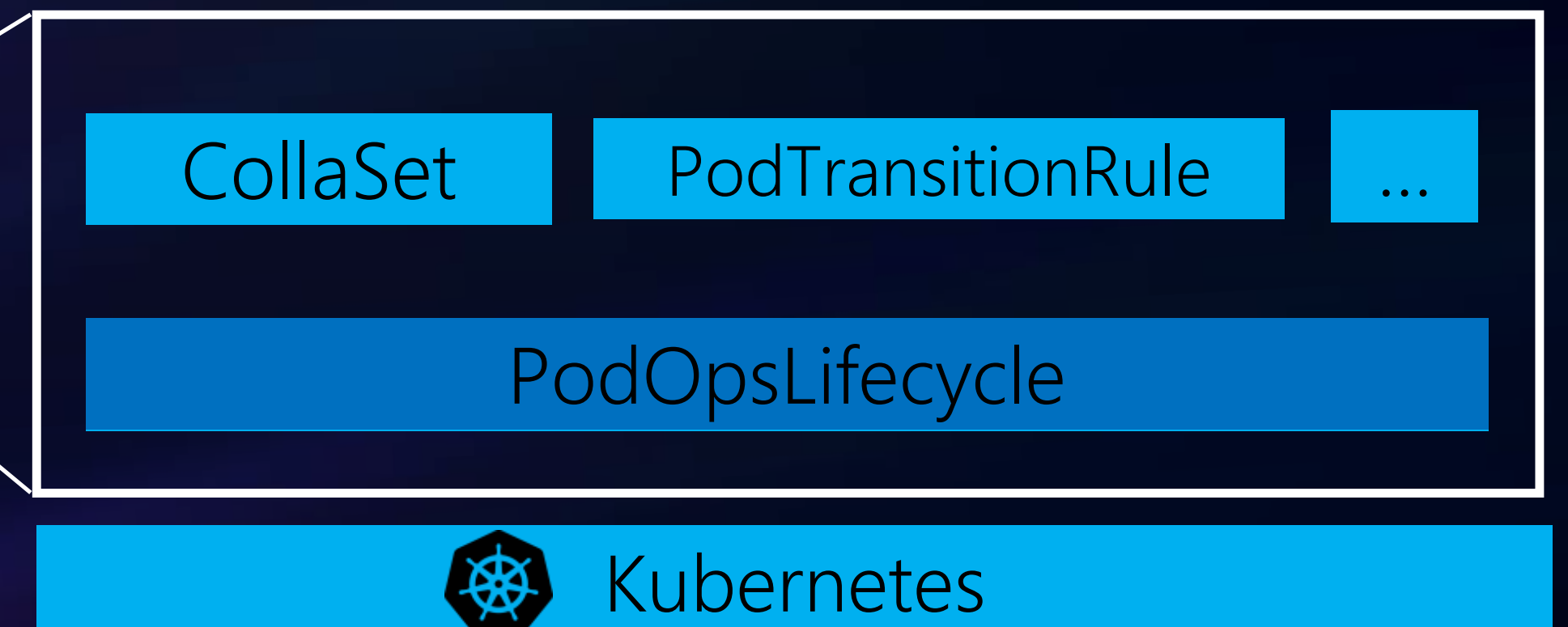


云原生资源交付与运维



挑战：

- 封装云原生运维复杂流程
- 沉淀运维的变更管控能力

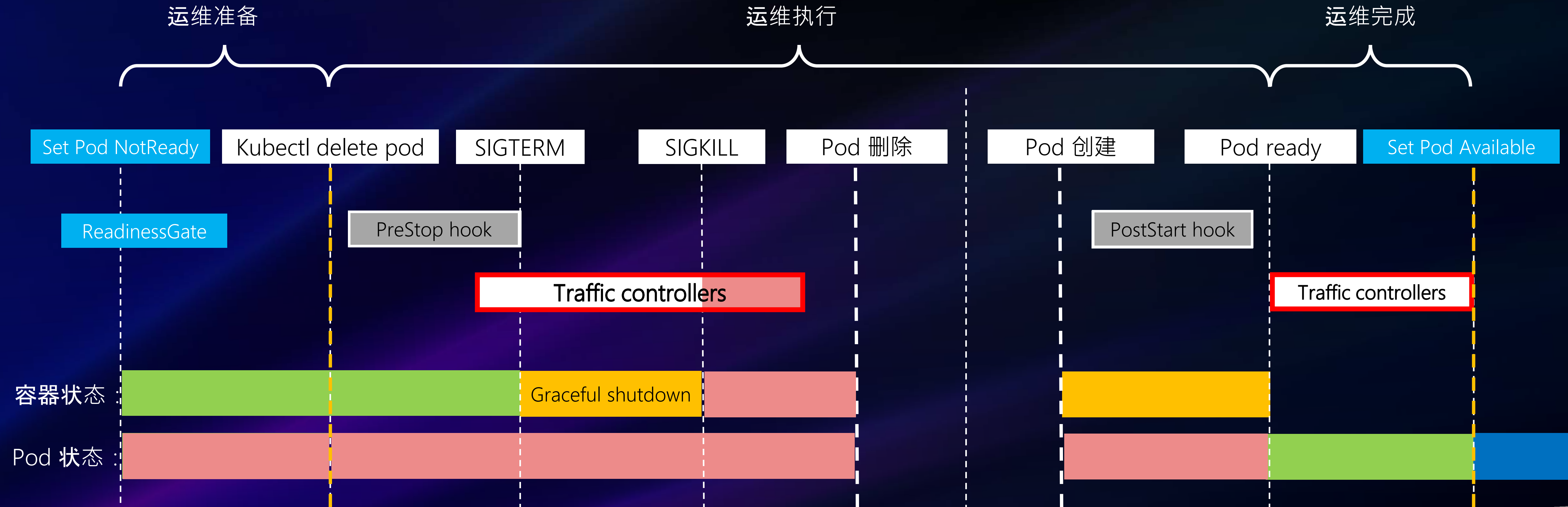


Pod 运维生命周期

原生 Pod 变更过程 K8s 提供的交互和管控能力有限



以流量为例：



Pod 运维生命周期

Pod 运维生命周期



Demo:流量无损运维

- Create a CollaSet to provision application Pods
- Create a K8s service to Provision SLB
- Create a PodTransitionRule to control operation risks

欢迎加入我们

- Web Site

- <https://kusionstack.io/>

- Github

- <https://github.com/KusionStack/>

- Twitter

- [@KusionStack](https://twitter.com/KusionStack)

- 微信公众号



微信小助手



钉钉群



THANKS