Self-Service Stream Processing Platform on Kubernetes at Apple

Chenya Zhang

KubeCon 2023 | Apple Inc. | November 7th

Agenda

Moving to the Cloud and Kubernetes

Adopting Apache Flink for Real-Time Data Processing

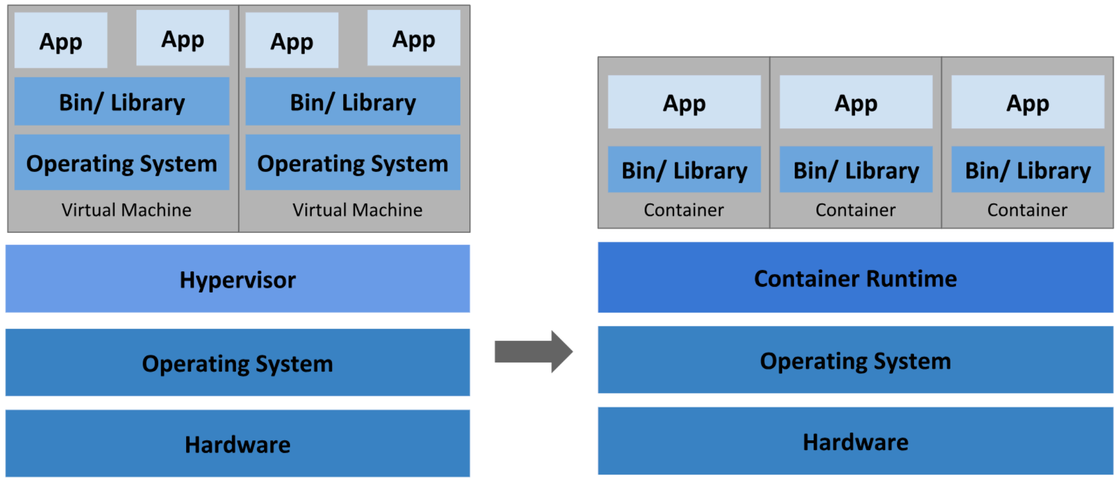
Challenges at Production Scale

Introducing Flink Control Plane on Kubernetes

Solutions on Cost Efficiency and Automation

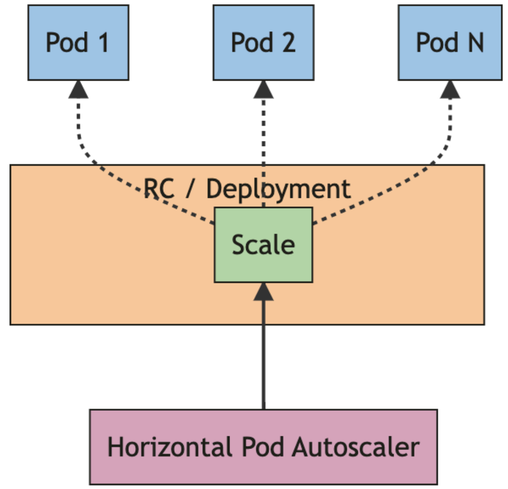
Moving to the Cloud and Kubernetes

- Automatic provision based on real-time demands
- Physical abstraction via containerized infrastructure
- Flexible interoperation across multi-cloud environments



Virtualized Deployment

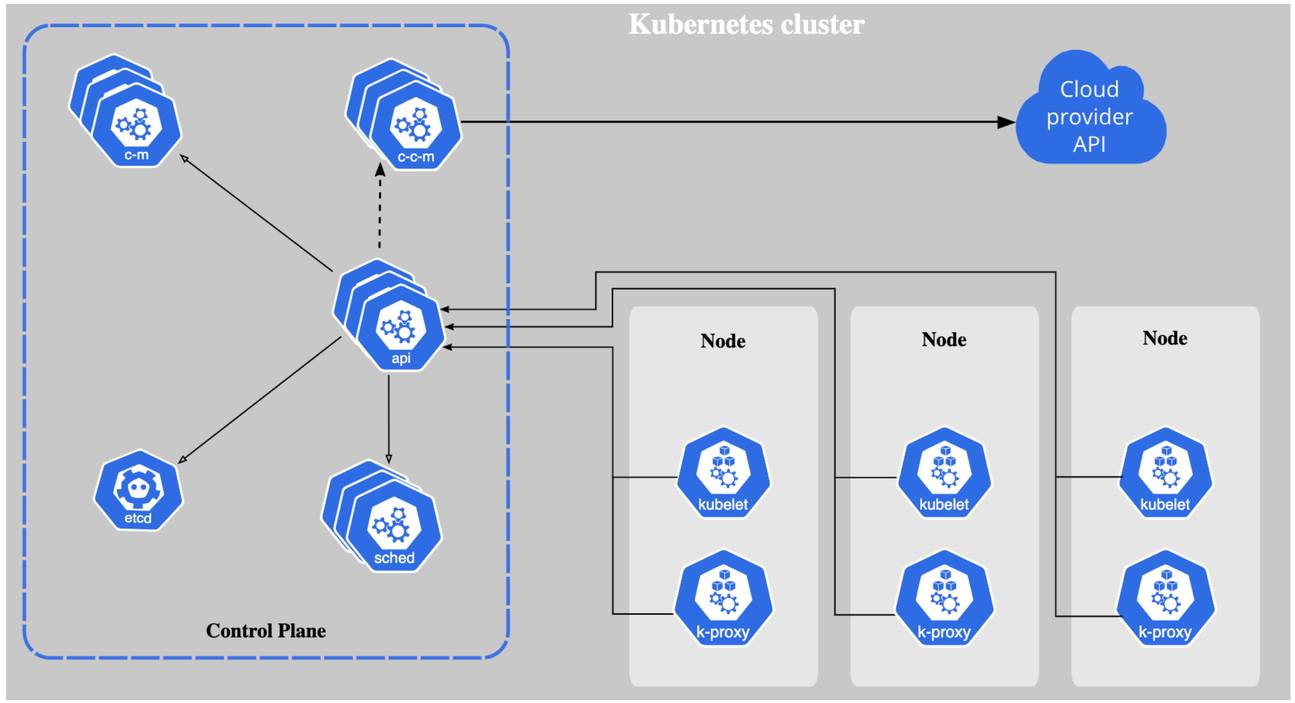
Container Deployment



[1] https://kubernetes.io/docs/concepts/overview/, [2] https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/

Moving to the Cloud and Kubernetes

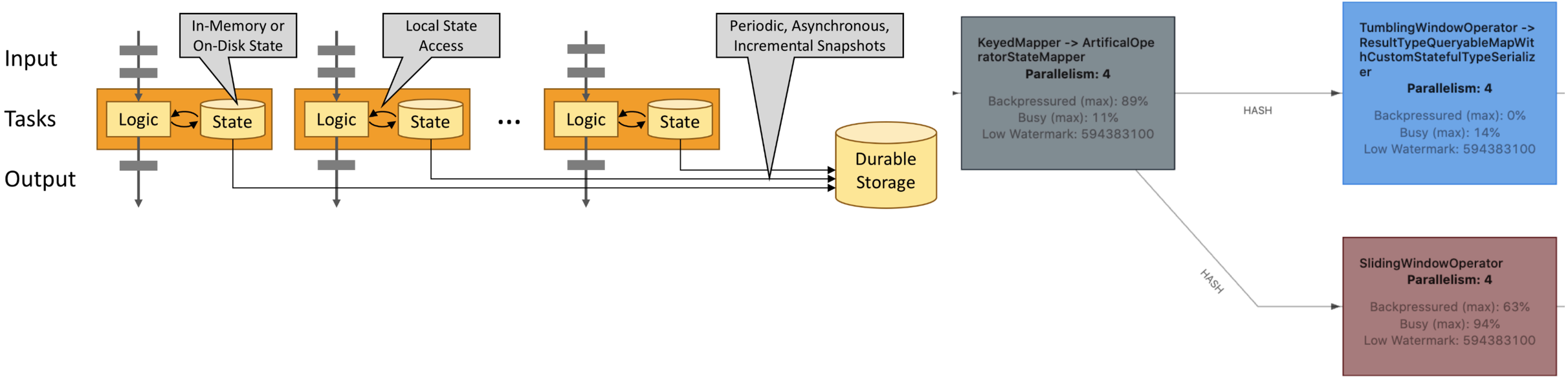
- Auto-adjustment for system consistency and resiliency
- Adaptability enabled by a modularized architecture



[1] https://kubernetes.io/docs/concepts/overview/components/

Adopting Apache Flink

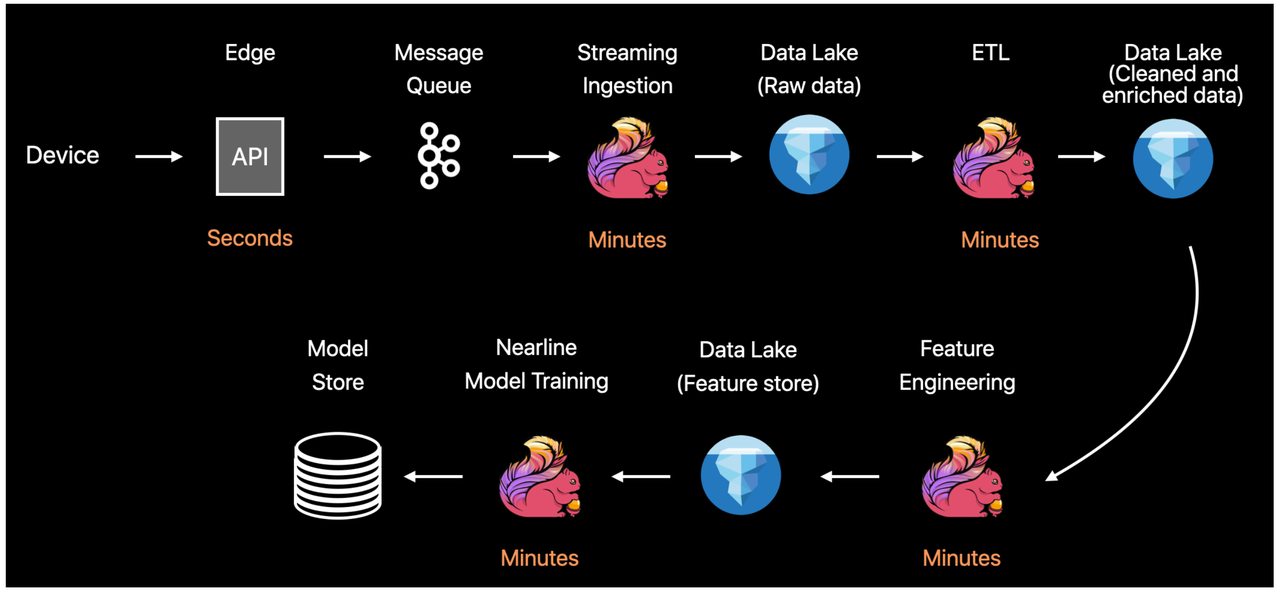
- In-memory pipelined execution with late data and back pressure handling
- Consistent snapshot to preserve accuracy and resilience of TB-state scale

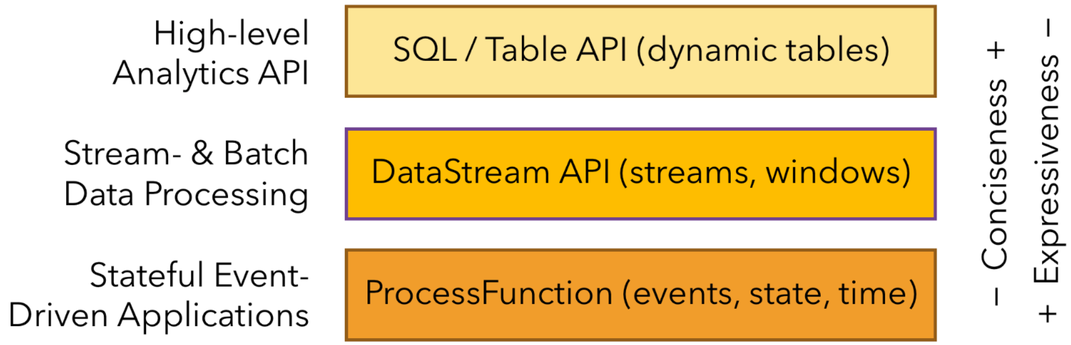


[1] https://nightlies.apache.org/flink/flink-docs-release-1.18/docs/learn-flink/overview/, [2] https://flink.apache.org/2021/07/07/how-to-identify-the-source-of-backpressure/

Adopting Apache Flink

- Built-in connectors for Kafka/Iceberg/... to simplify system integration
- Layered APIs with ProcessFunction and advanced windowing mechanism





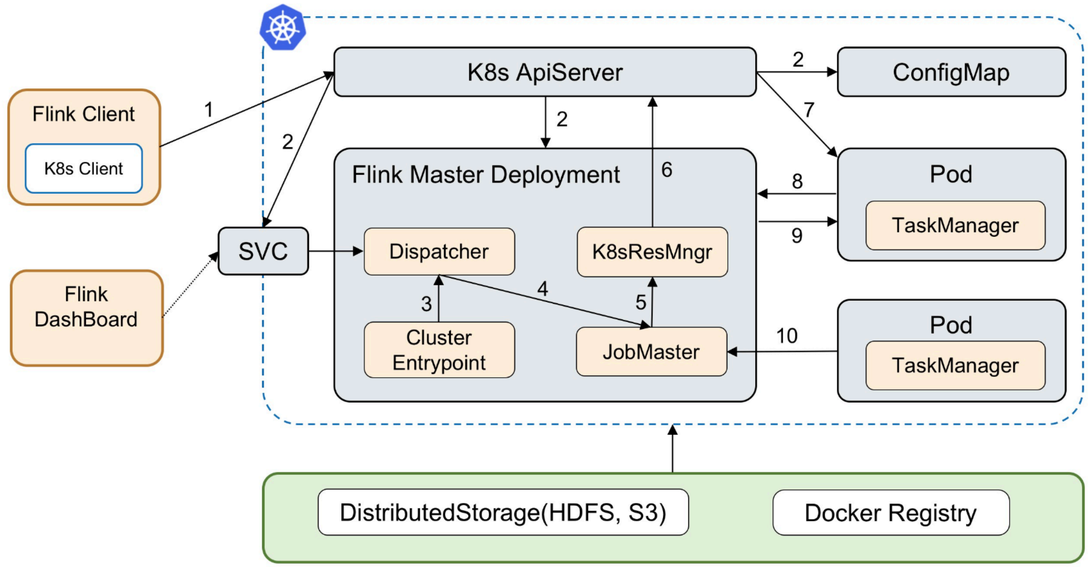
[1] https://nightlies.apache.org/flink/flink-docs-release-1.16/docs/learn-flink/overview/

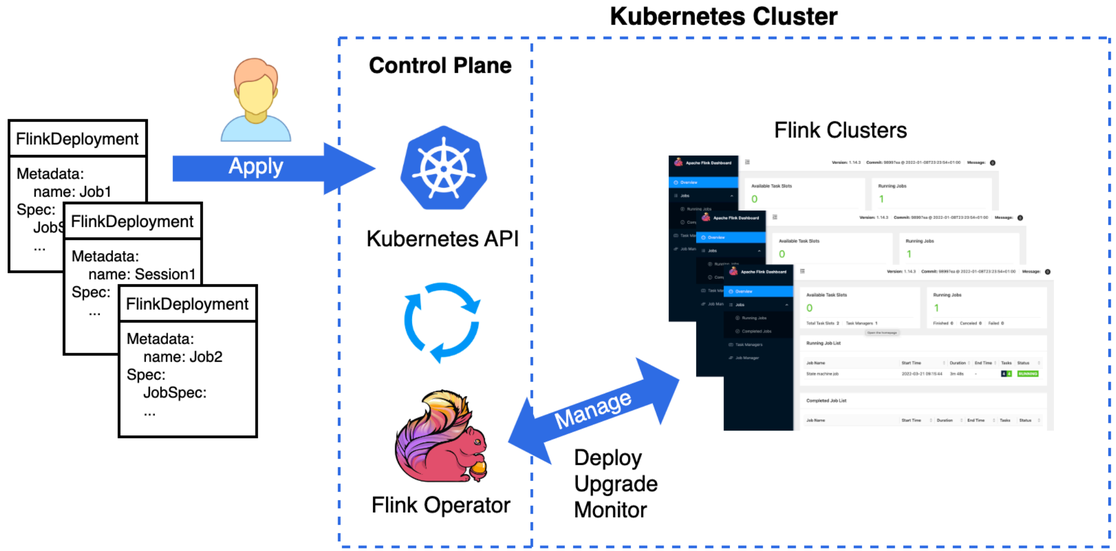
Challenges at Production Scale

- Need an automated approach to deploy Flink in Kubernetes mode
- Miss a control plane to scale beyond a single Kubernetes cluster
- Self-onboard, observe, and troubleshoot without admins in the loop
- Bulk operate on a large amount of deployments by users and admins
- Automatically scale stateless and stateful apps per incoming traffic
- Manage resource availability and cluster upgrade in the cloud

Deploy Flink on Kubernetes

- Self-contained Flink native K8s support with embedded K8s client
- K8s Custom Resource and Operator to manage application lifecycle
- Automation and abstraction of application upgrade & failure recovery

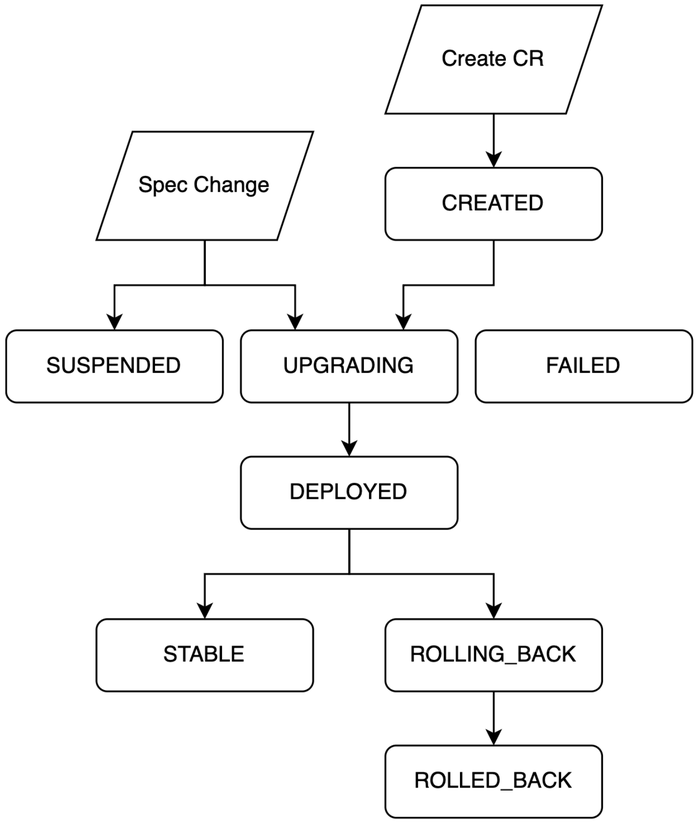


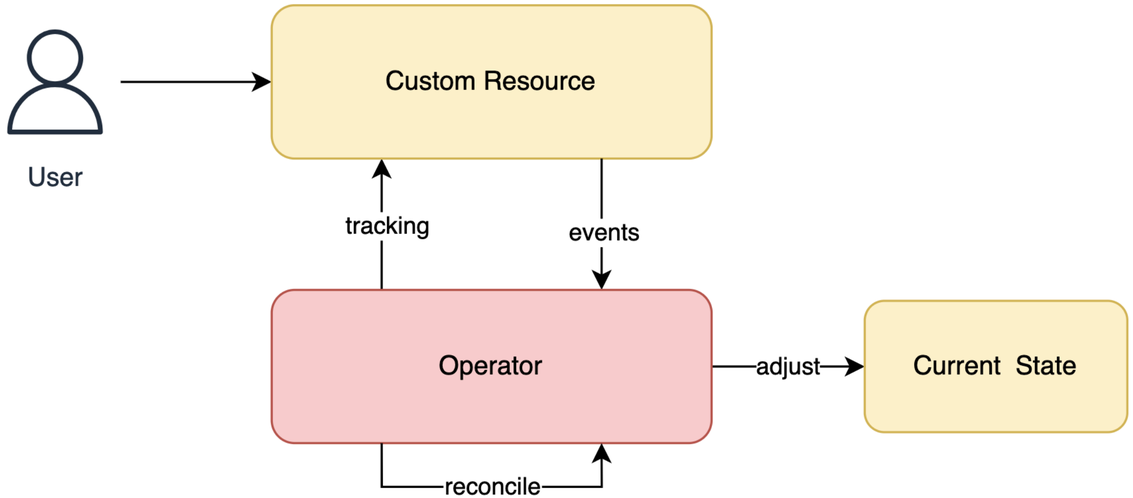


[1] https://flink.apache.org/2021/02/10/how-to-natively-deploy-flink-on-kubernetes-with-high-availability-ha/, [2] https://nightlies.apache.org/flink/flink-kubernetes-operator-docs-main/

Deploy Flink on Kubernetes

- FLIP-212: Introduce Flink Kubernetes Operator
- Reconcile any required changes and execute upgrades





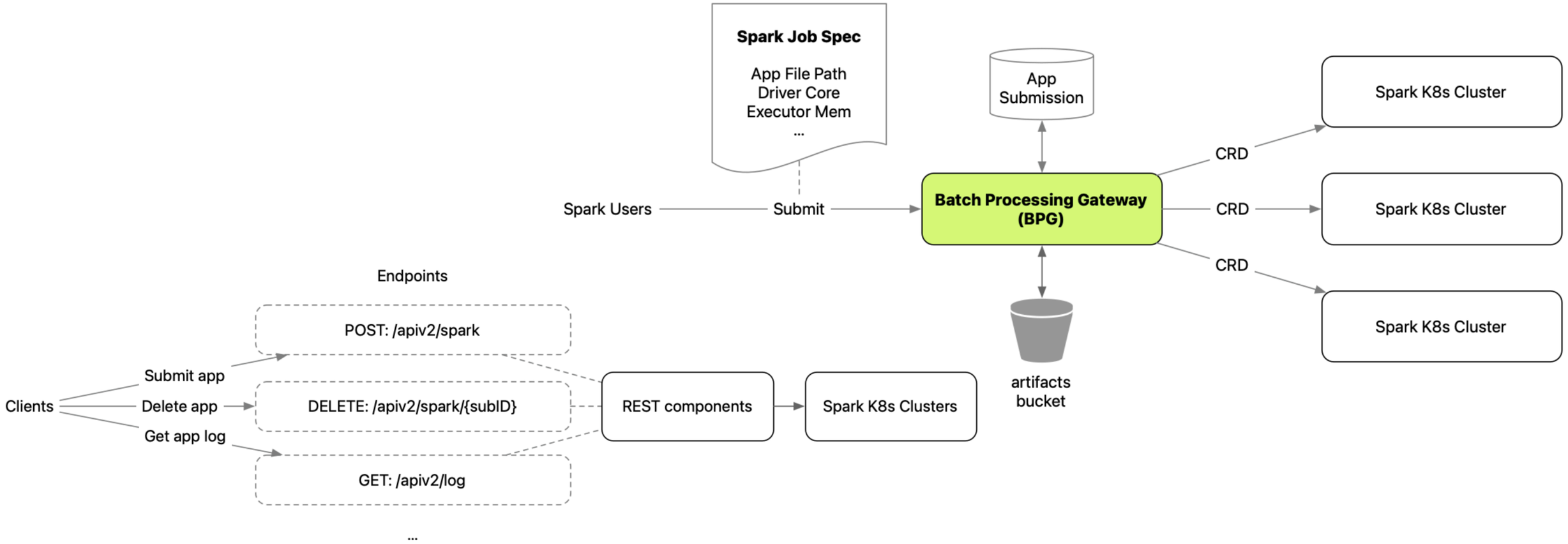
[1] https://cwiki.apache.org/confluence/display/FLINK/FLIP-212%3A+Introduce+Flink+Kubernetes+Operator, [2] https://nightlies.apache.org/flink/flink-kubernetes-operator-docs-main/docs/concepts/architecture/, [3] https://nightlies.apache.org/flink/flink-kubernetes-operator-docs-main/docs/concepts/controller-flow/

What If ...

- Kubernetes cluster or Flink K8s Operator fails and goes down?
- User group need to map beyond one Kubernetes namespace?
- Authentication and authorization need to meet extra requirements?
- Deployment information need to be persisted for audit or recovery?
- Admins need to operate across multiple clusters and accounts?
- Other service or platform need to integrate for stream processing?

Lessons from BPG

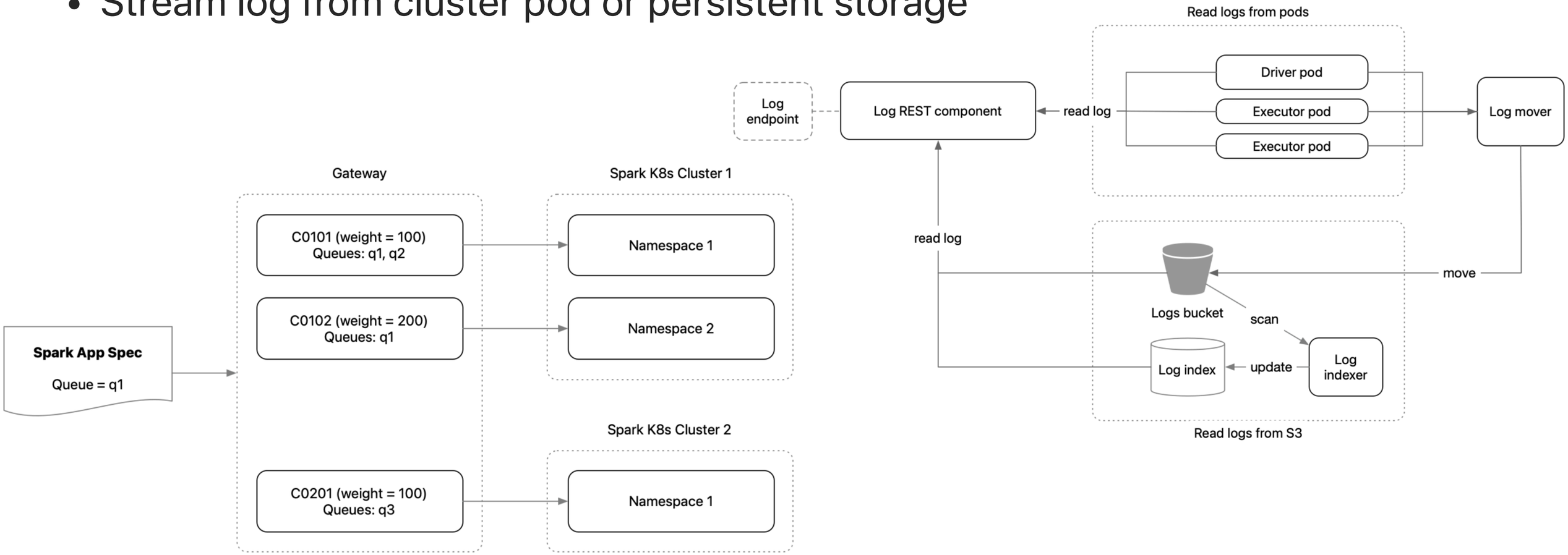
- Apple’s batch processing gateway - REST APIs exposed to clients
- Translate user requests to CRDs supported by Spark K8s operator



[1] https://github.com/apple/batch-processing-gateway

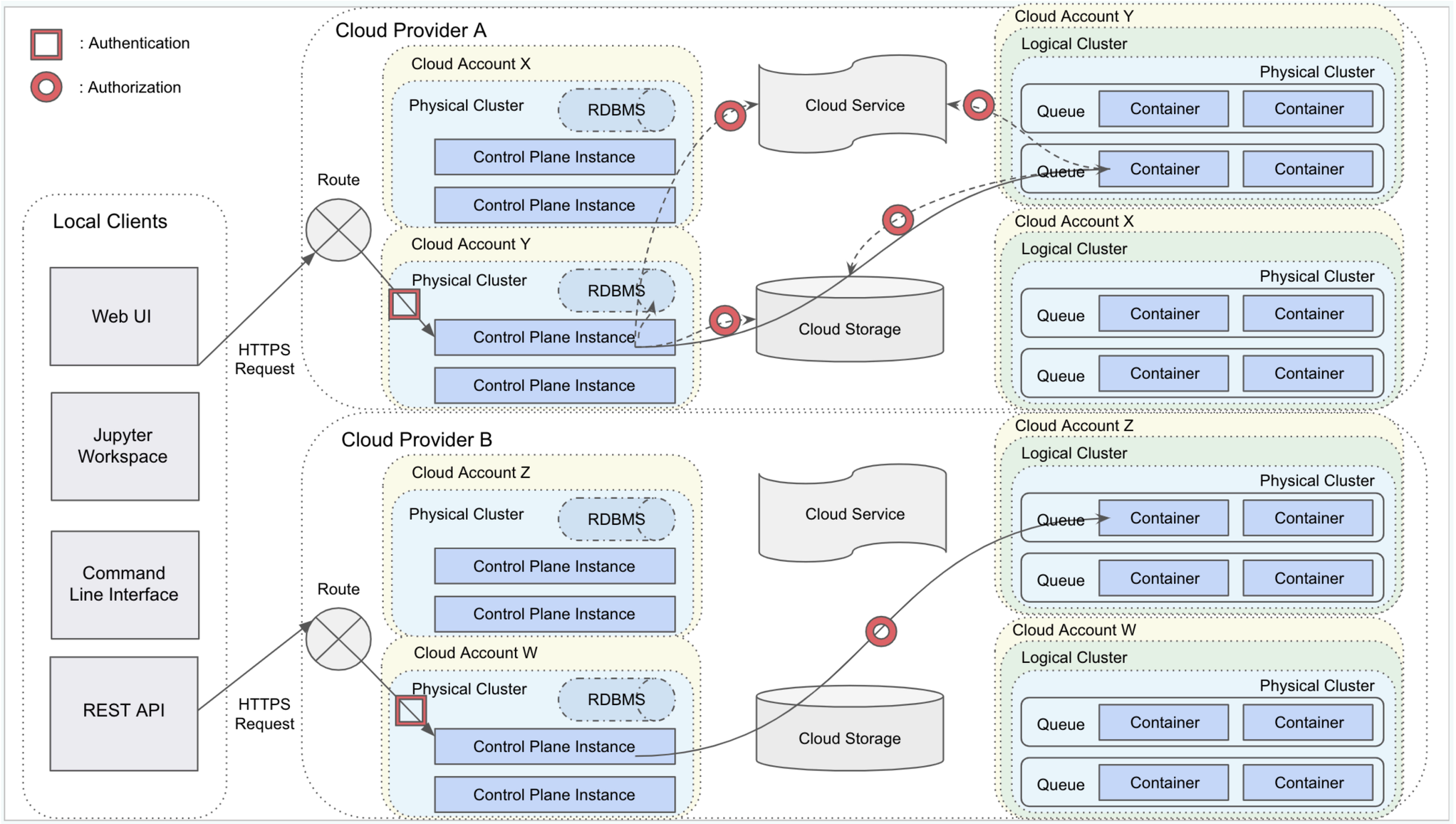
Lessons from BPG

- Route requests based on queues and weights
- Stream log from cluster pod or persistent storage



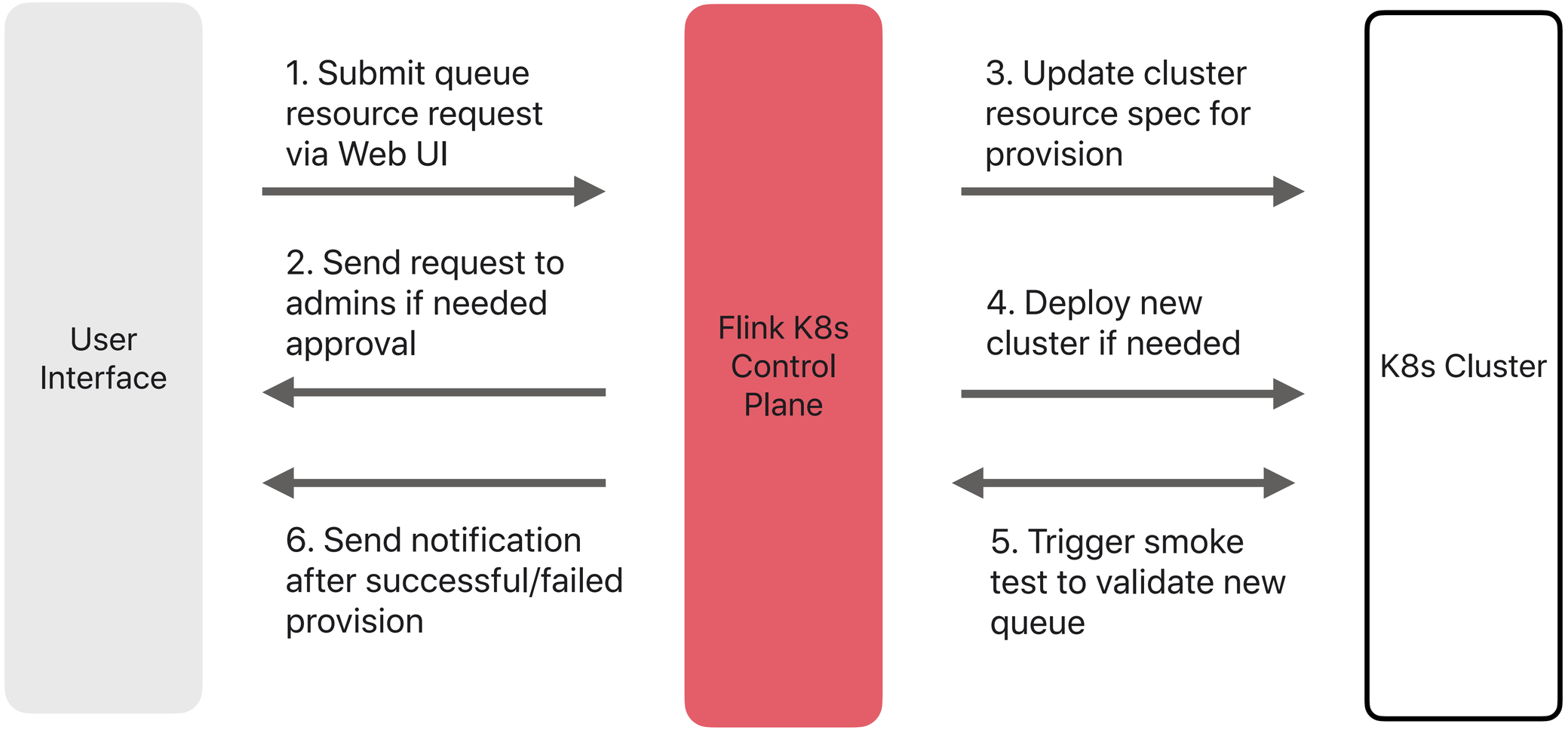
[1] https://github.com/apple/batch-processing-gateway

Introduce Flink Control Plane



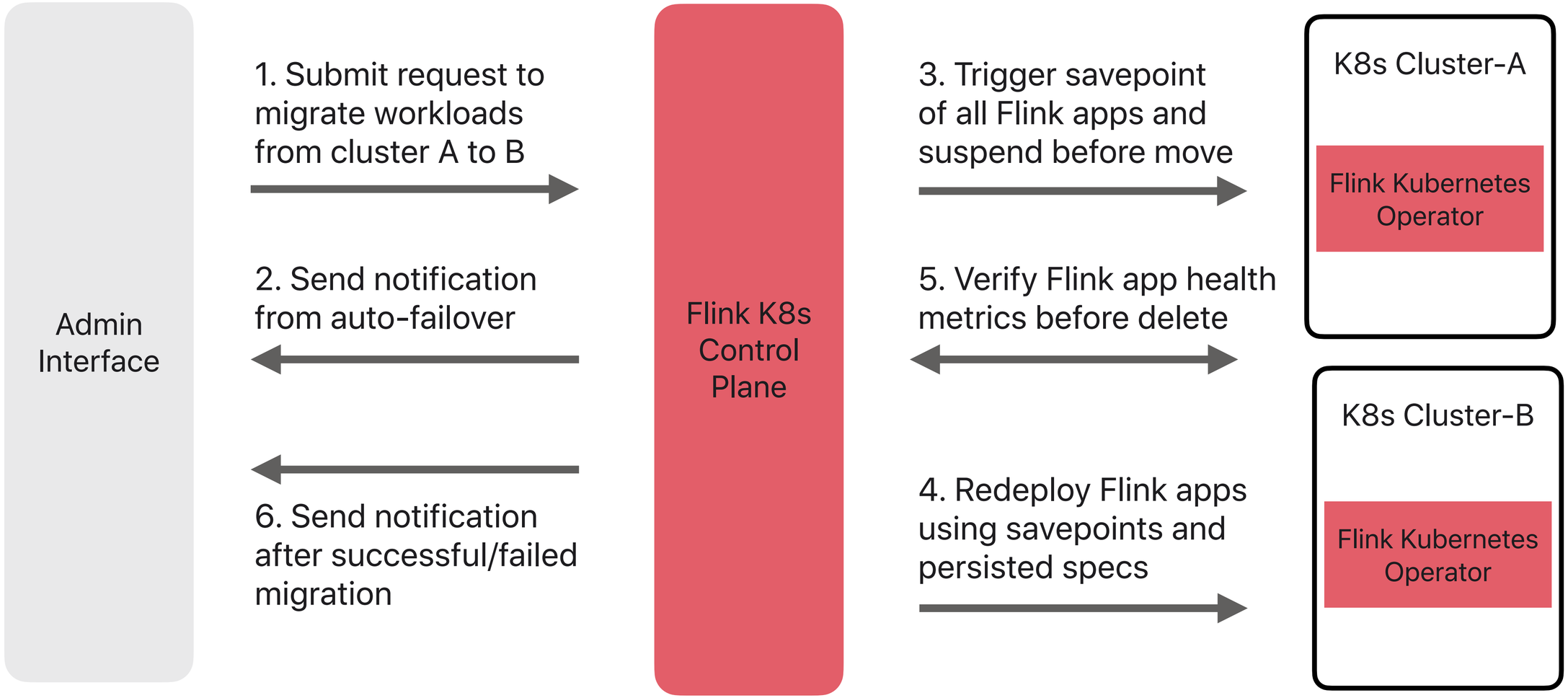
Introduce Flink Control Plane

- Self-service user onboarding experience



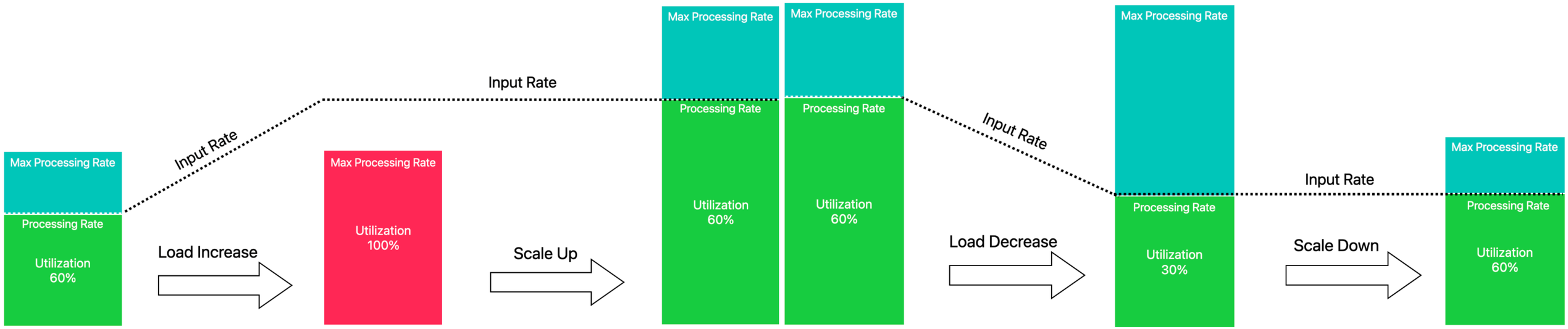
Introduce Flink Control Plane

- Automated cluster migration and failover



Autoscale Flink Application

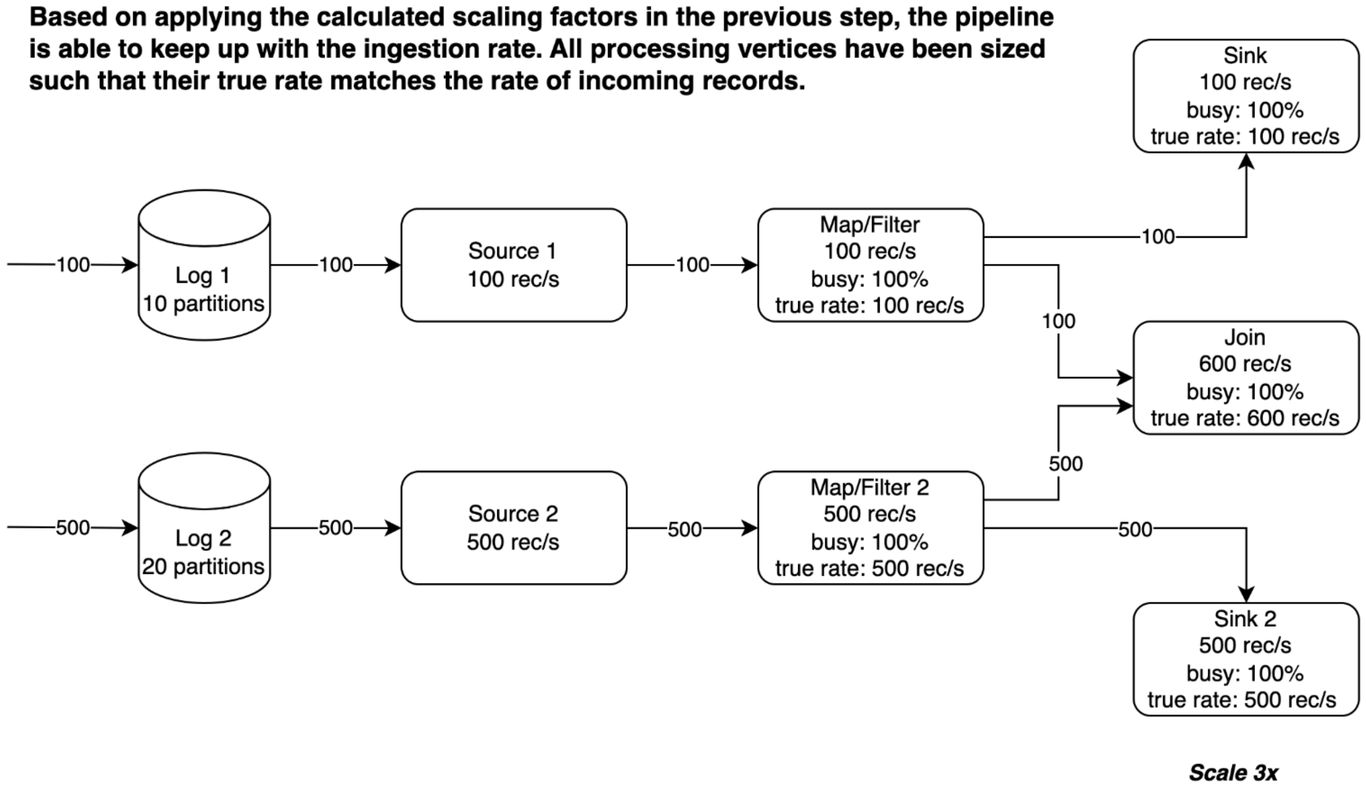
- FLIP-271: Autoscaling - when and how much to scale?
- Each scaling decision has an associated cost



[1] https://nightlies.apache.org/flink/flink-kubernetes-operator-docs-main/docs/custom-resource/autoscaler/

Autoscale Flink Application

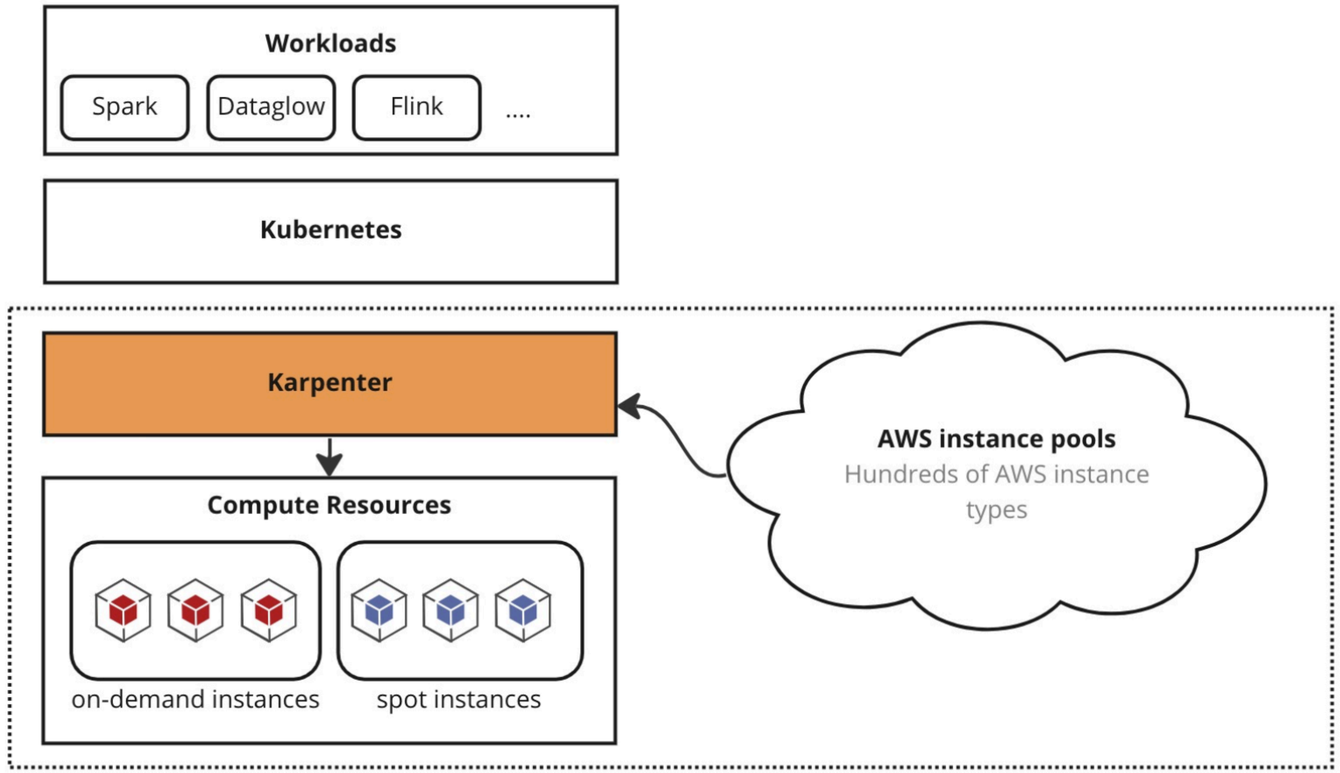
- Predict parallelism change effects on downstream job vertices
- Flink Kubernetes Operator integration and deployment metrics



[1] https://cwiki.apache.org/confluence/display/FLINK/FLIP-271%3A+Autoscaling

Stay Resource-Aware

- From Cluster Autoscaler to Karpenter
- Customized scheduler to improve bin-packing of sticky deployment



[1] https://github.com/aws/karpenter

Wrap up: Self-Service Platform