KubeCon | CloudNativeCon

North America 2024

Automated Multi-Cloud Large Scale K8s Cluster Lifecycle Management

Sourav Khandelwal
Software Engineer, Databricks

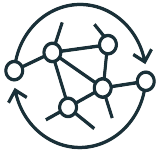
Databricks: E2E Data & AI Platform

Data Lake

Machine Learning

Streaming

Generative AI



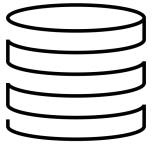
Data Science

Governance

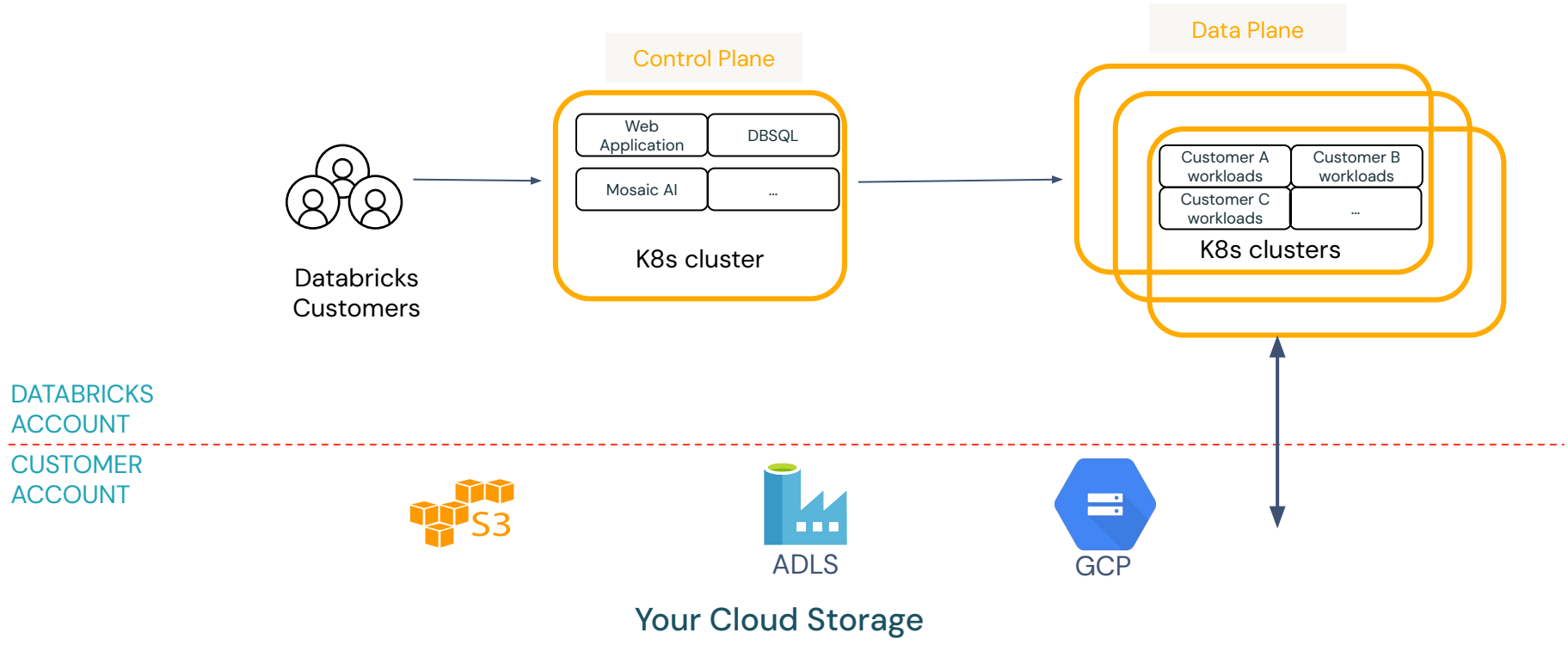
Orchestration & ETL

BI

Data Warehouse



Serverless Architecture



Serverless Compute Footprint

- Manage over 1000 K8s clusters
  - All created in the last 3 years
- 3 clouds - AWS, Azure, GCP
- Deployed in more than 60 regions

Cluster Management

- Provisioning / Deprovisioning of K8s clusters
  - Multi-cloud
  - Scalable
  - Reliable
- Upgrades
  - Support Cluster rotations - bring new clusters with new configuration and retire old clusters
    - necessary to do major infrastructure changes like cluster networking, k8s version upgrades, etc.
  - In-Place - update the nodes of clusters

What Cluster Provisioning Entails ?

- Ready to use clusters by product teams

- Network - Vnet/VPC and Subnets
- Kubernetes cluster in cloud - AKS, EKS, GKE
- Nodepools

- Couple of cloud resources
  - IAM roles
  - EventHub/Kinesis Stream for logs
- Infra setup of the K8s cluster
  - Monitoring
  - Logging
  - RBAC
  - K8s Secrets - harbor image registry secrets, AKS SP secrets, etc
  - Essential infrastructure services
    - Cert Manager - management and issuance of TLS certificates
    - Ingress proxy

Previous Framework

- **Single python script** of many, many steps run as a spinnaker pipeline
  - Very brittle
  - Required manual retries
  - Lacked Idempotency
  - Incomplete Monitoring
  - Expose low-level cloud specific details
  - Poor configuration management
    - one configuration file to read and write input and output

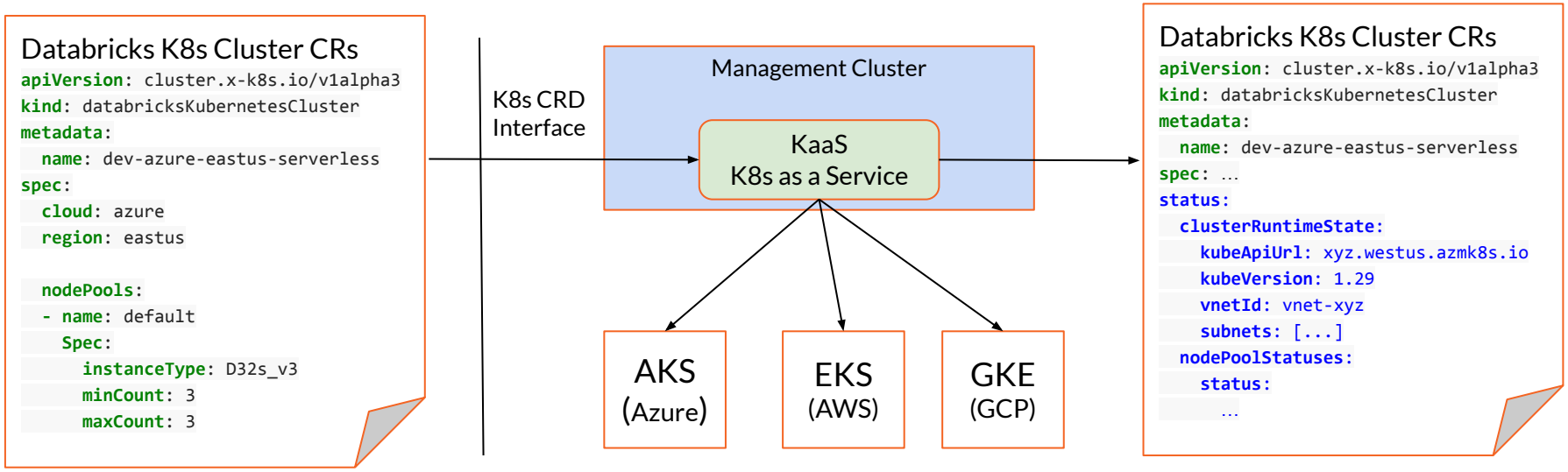
- New cluster creation could take **weeks**

New Cluster Lifecycle Framework

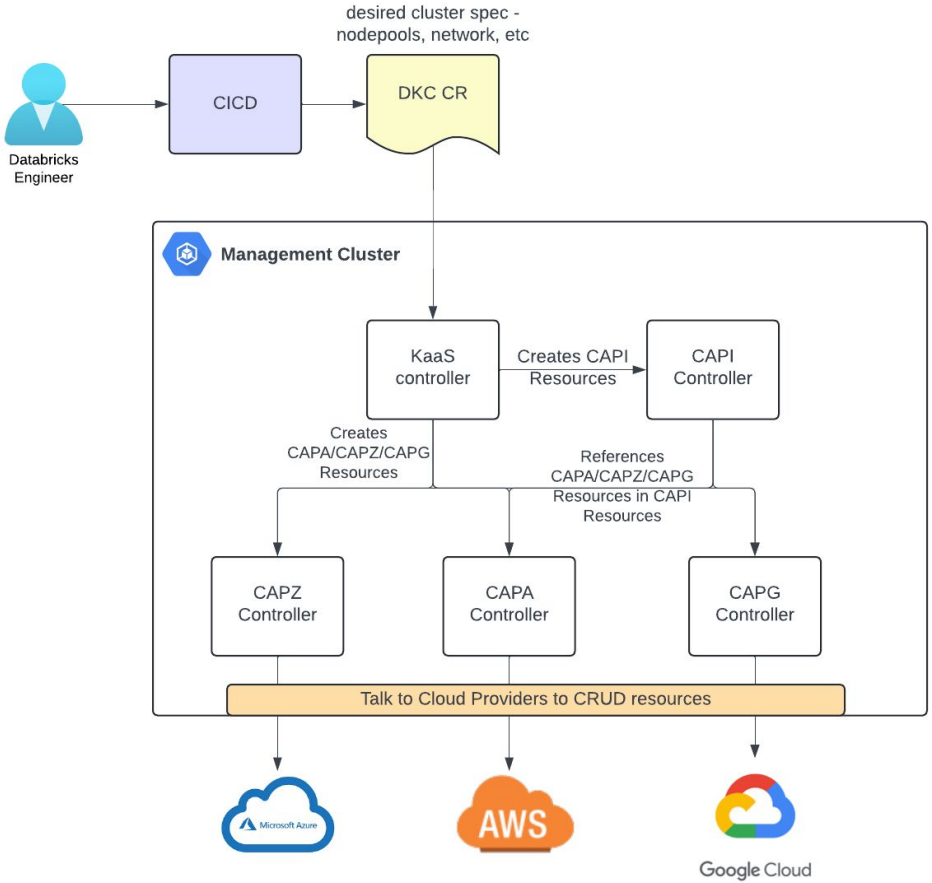
Looked towards Kubernetes for inspiration

Cluster Lifecycle

- Use Kubernetes to manage Kubernetes clusters lifecycle (K8s Operator Pattern)
  - Declarative configuration of kubernetes clusters by modeling it as a "Custom Resource Definition"
  - Continuously reconcile from the current to the desired state
  - Single CRD for all 3 clouds



Cluster API


desired cluster spec - nodepools, network, etc
Databricks Engineer
CICD
DKC CR
Management Cluster
KaaS controller
Creates CAPI Resources
CAPI Controller
Creates CAPA/CAPZ/CAPG Resources
References CAPA/CAPZ/CAPG Resources in CAPI Resources
CAPZ Controller
CAPA Controller
CAPG Controller
Talk to Cloud Providers to CRUD resources
Microsoft Azure
AWS
Google Cloud

Improvements

- Old Python Script run with Spinnaker pipeline
- Kubernetes-as-a-Service Operator

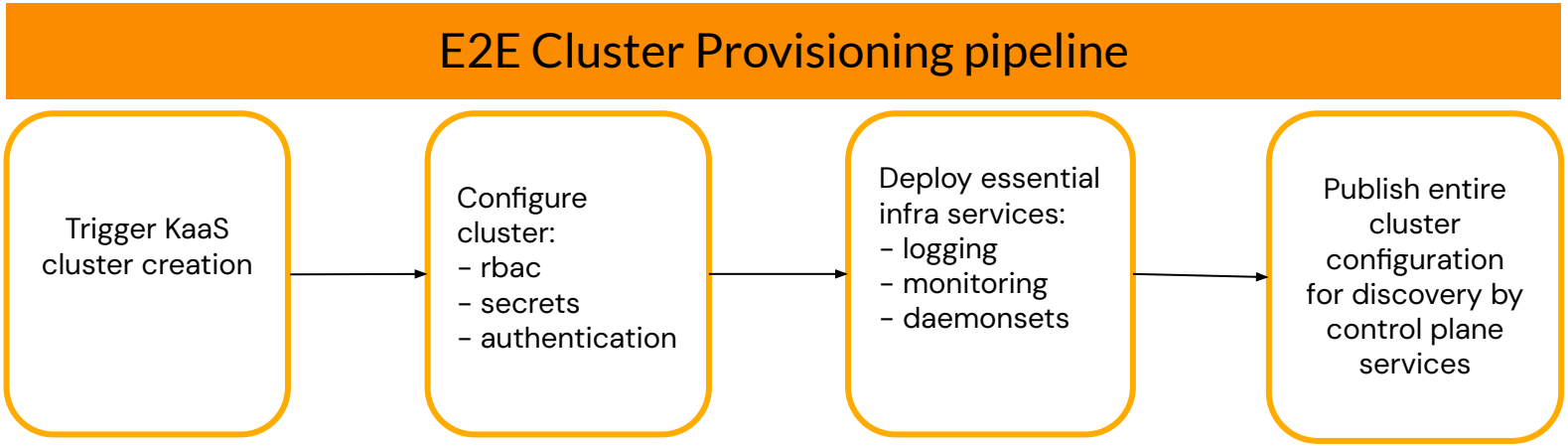
- Required manual retries
- Continuous reconciliation towards desired state

- Lacked Idempotency
- Operations invoked by Controller is idempotent

- Incomplete Monitoring
- Enable rich logging and monitoring

- Expose low-level cloud specific details
- Abstracts out majority of the cloud specific details by offering a consistent interface across all environments

All Problems Solved ?



- Scalable ?
  - Hard to create multiple clusters across different regions at one go
    - Needed to manually construct DKC CR for every cluster and run the pipeline
  - Upgrades that require “**Cluster Rotations**” is painful
    - Required to create multiple CRs and pipeline runs to create replacement clusters and delete old clusters

All Problems Solved ?

- Reliable ?
  - Steps that create cloud resources before and after the k8s cluster itself, and deploy infra services to the k8s cluster, still had the original issues:
    - Required manual retries
    - Lacked Idempotency
    - Incomplete Monitoring

- E2E Latency ?
  - High
  - Cluster configuration and its lifecycle state are merged into our code repo at the end of the successful run of the pipeline
  - Serverless control plane services discover clusters from this checked-in configuration file during their service deployments => this could take a week sometimes

ClusterSet

- A level of abstraction on top of KaaS

- ClusterSet Lifecycle Manager
  - Manage clusters in the unit of “ClusterSet” which is a set of cluster of the same “type”
  - Manages lifecycle of all resources that must be created before or after the K8s cluster itself, in addition to creating the CR that is read by KaaS
  - Implemented as a Kubernetes-style operator

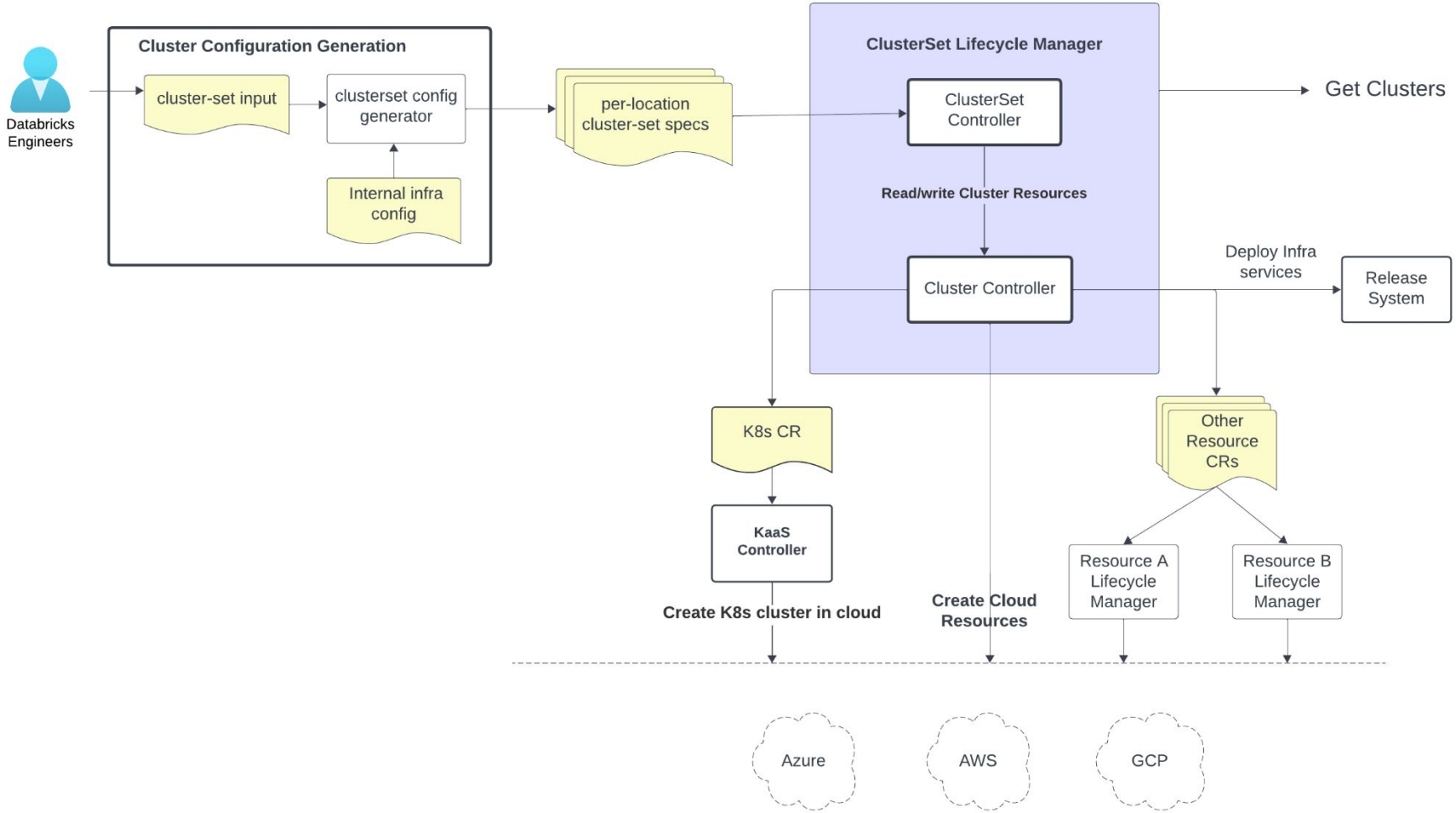
ClusterSet

ClusterSet Spec

```
kind: ClusterSet
metadata:
  location:<env> + <cloud> + <region>
  clusterType: ModelServing
spec:
  count: 5
  features:
  - private-connectivity
  - byon
  nodePools:
  - name: default
    Spec:
      instanceType: D32s_v3
      minCount: 3
      maxCount: 3
  infra-services:
  - rbac
  - namespaces
  - observability
  - ...
  update_strategy:
  - cluster-rotation:
  - max_surge:
  tags:
```

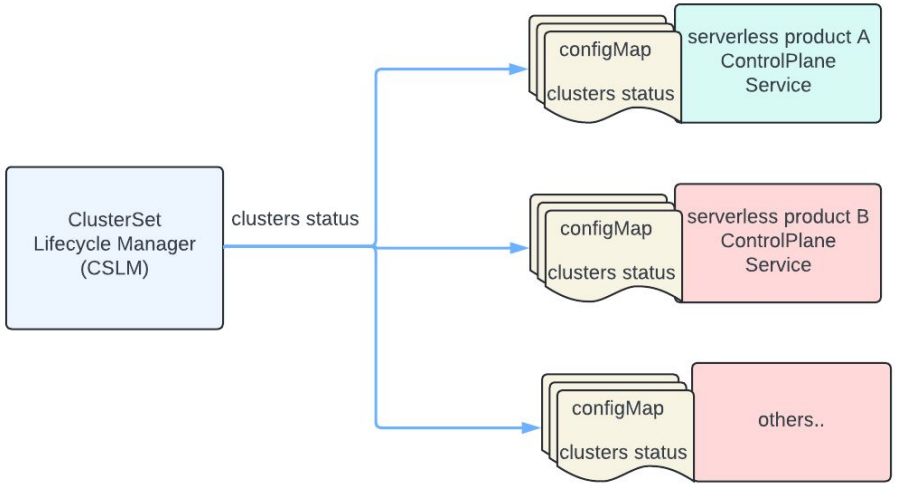
ClusterSet Status

```
status:
  clusters:
  - cluster-xyz
    kind: Cluster
    metadata:
      location:<env> + <cloud> + <region>
      clusterType: ModelServing
    spec:
      features: ...
      nodePools:
      - name: default
      - spec: ...
      infra-services:
      tags:
    status:
    - state: <CREATING / READY / DELETION>
    - network:
    - infra-services-status:
    - ...
```


Databricks Engineers
Cluster Configuration Generation
cluster-set input
clusterset config generator
Internal infra config
per-location cluster-set specs
ClusterSet Lifecycle Manager
ClusterSet Controller
Get Clusters
Read/write Cluster Resources
Cluster Controller
Deploy Infra services
Release System
K8s CR
KaaS Controller
Create K8s cluster in cloud
Create Cloud Resources
Other Resource CRs
Resource A Lifecycle Manager
Resource B Lifecycle Manager
Azure
AWS
GCP

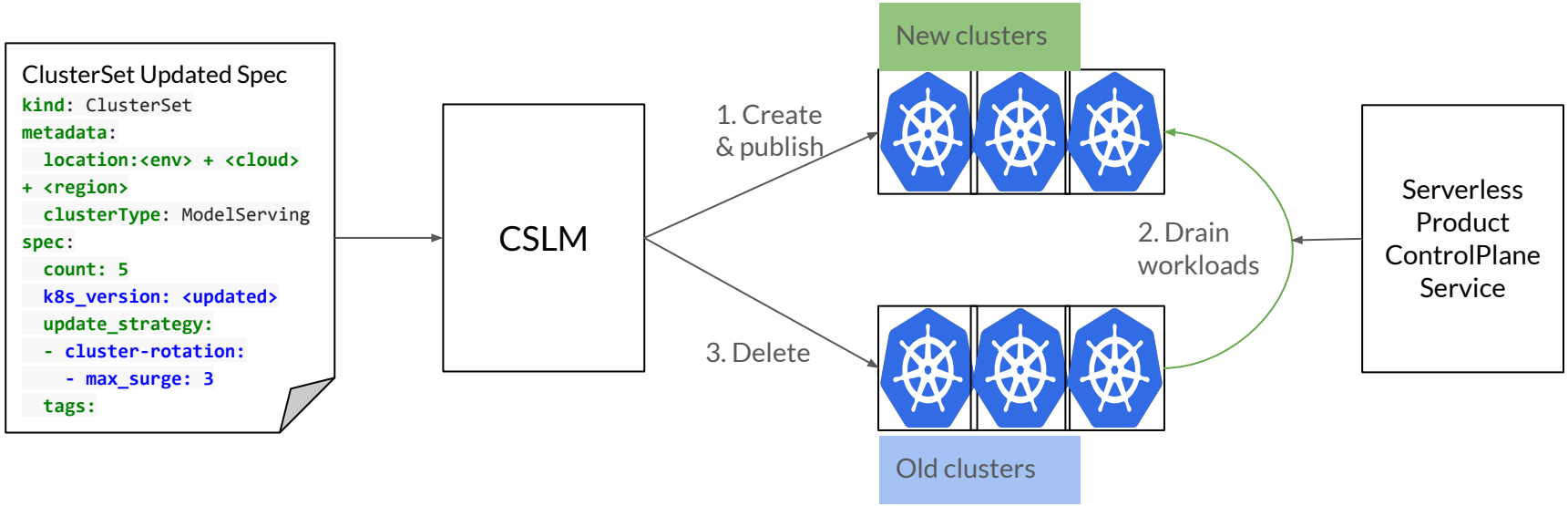
Cluster Discovery

- Publish cluster configurations automatically to serverless product control plane services.
- Reduces latency of control plane services discovering clusters to < 1 hour

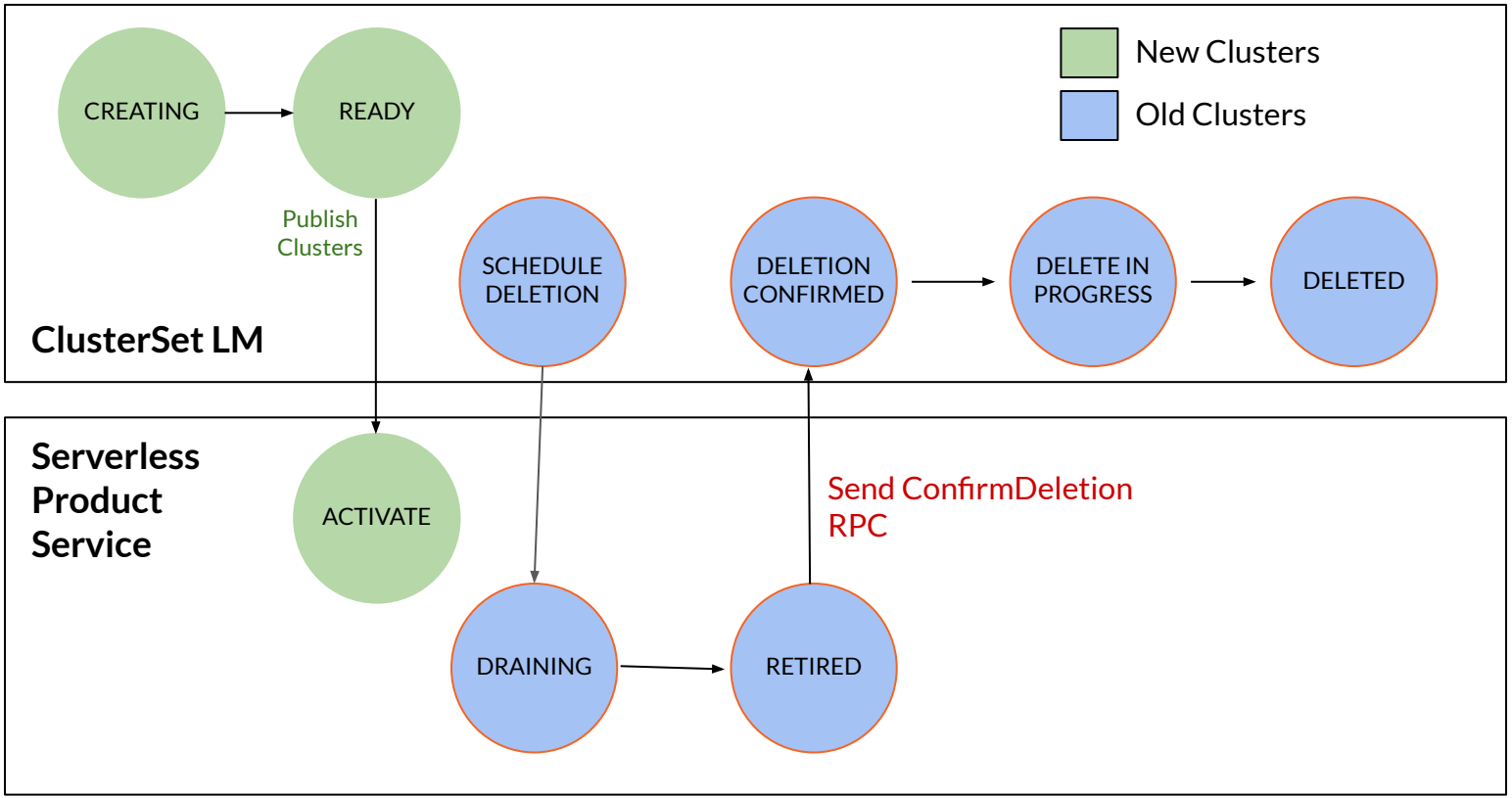


Cluster Rotation Upgrades

1. Create a batch of new clusters with updated configurations
2. Drain workloads from old clusters to new clusters
3. Delete the batch old clusters
4. Repeat steps 1-3 until all clusters are upgraded in the set



Upgrades: Cluster State Machine



Results

- Scalable ?
  - Create multiple clusters across different regions at one go
  - Automated cluster swap upgrades
- Reliable ?
  - All provisioning steps have continuous reconciliation
    - Idempotency
    - Automatic Retries
- E2E Latency ?
  - Dynamic cluster discovery within 1 hour

Questions

Thank You

We’re hiring!
https://www.databricks.com/company/careers/open-positions
https://www.linkedin.com/in/souravkhandelwal/