TEL4VN

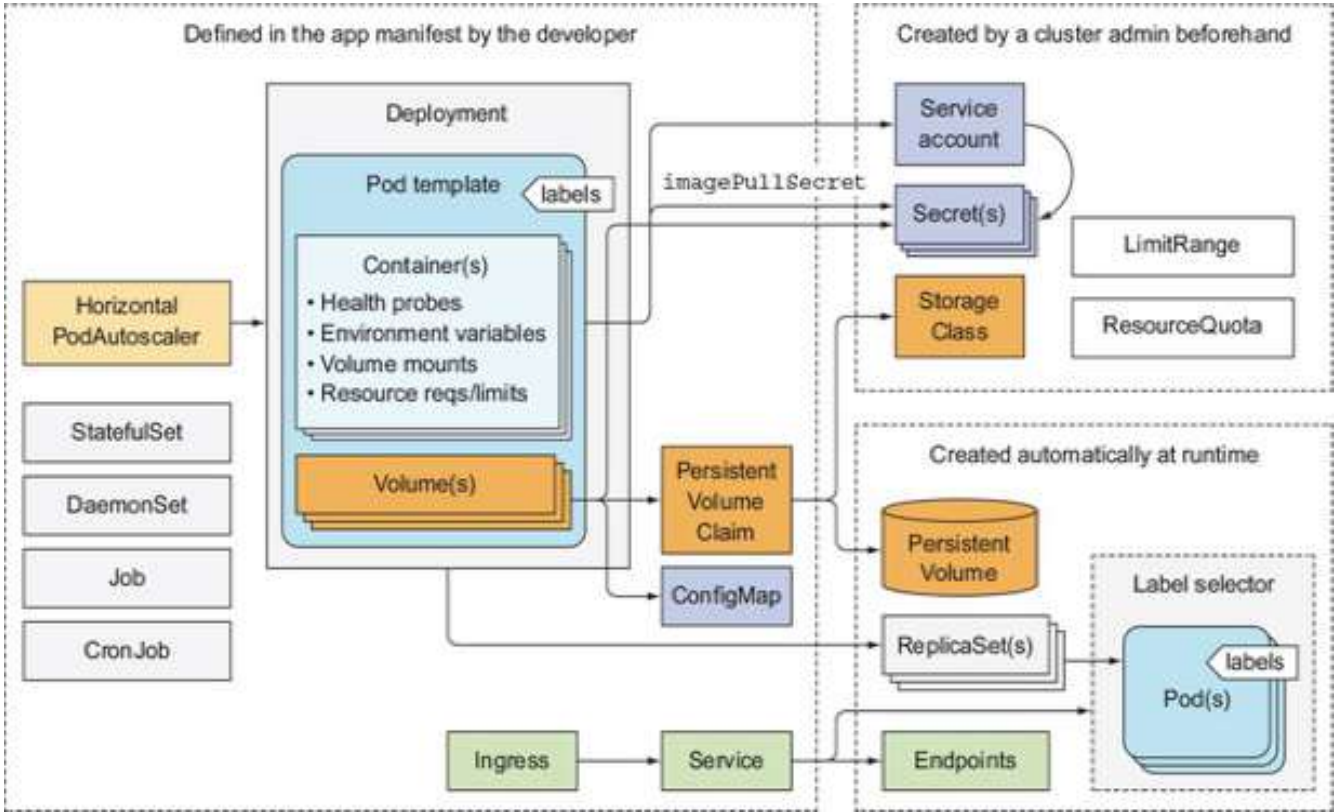
What's Next

THAOLUONG
09/2020

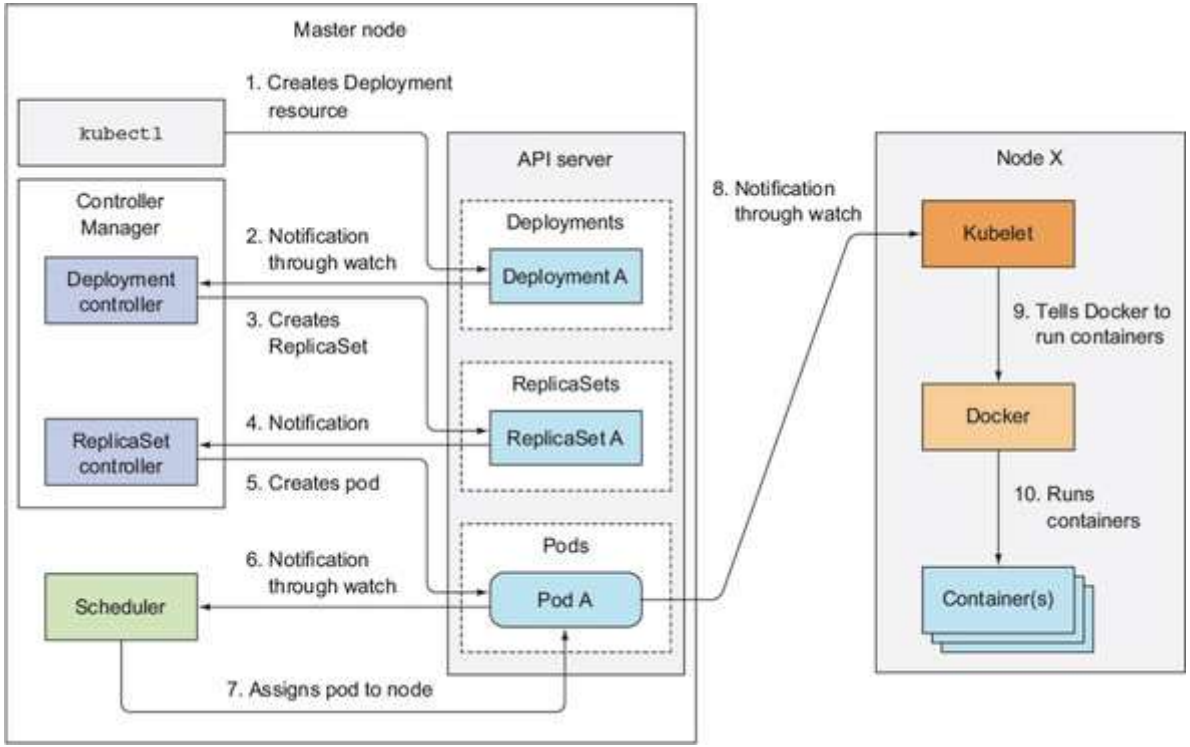
Content

- ❑ Overview
- ❑ Advance
- ❑ What’s next?

Bring Everything Together



How They Cooperate



Automatic scaling of pods and cluster nodes

- Horizontal Pod Autoscaler (HPA)
- Vertical Pod Autoscaler (VPA)
- Cluster Autoscaler (CA)

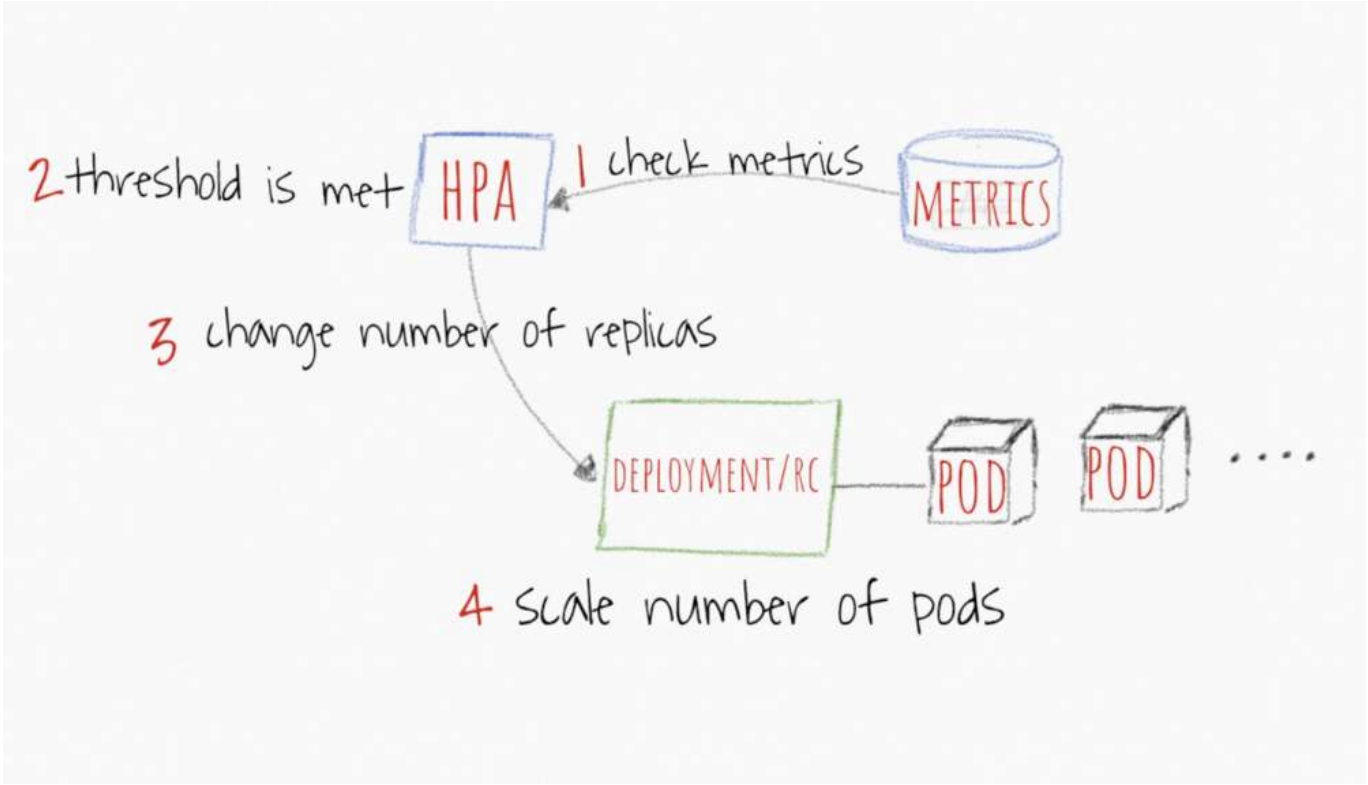
High-level HPA workflow

- HPA continuously checks metrics values you configure during setup AT A DEFAULT 30 SEC intervals

- HPA attempts to increase the number of pods If the SPECIFIED threshold is met

- HPA mainly updates the number of replicas inside the deployment or replication controller

- The Deployment/Replication Controller WOULD THEN roll-out ANY additional needed pods



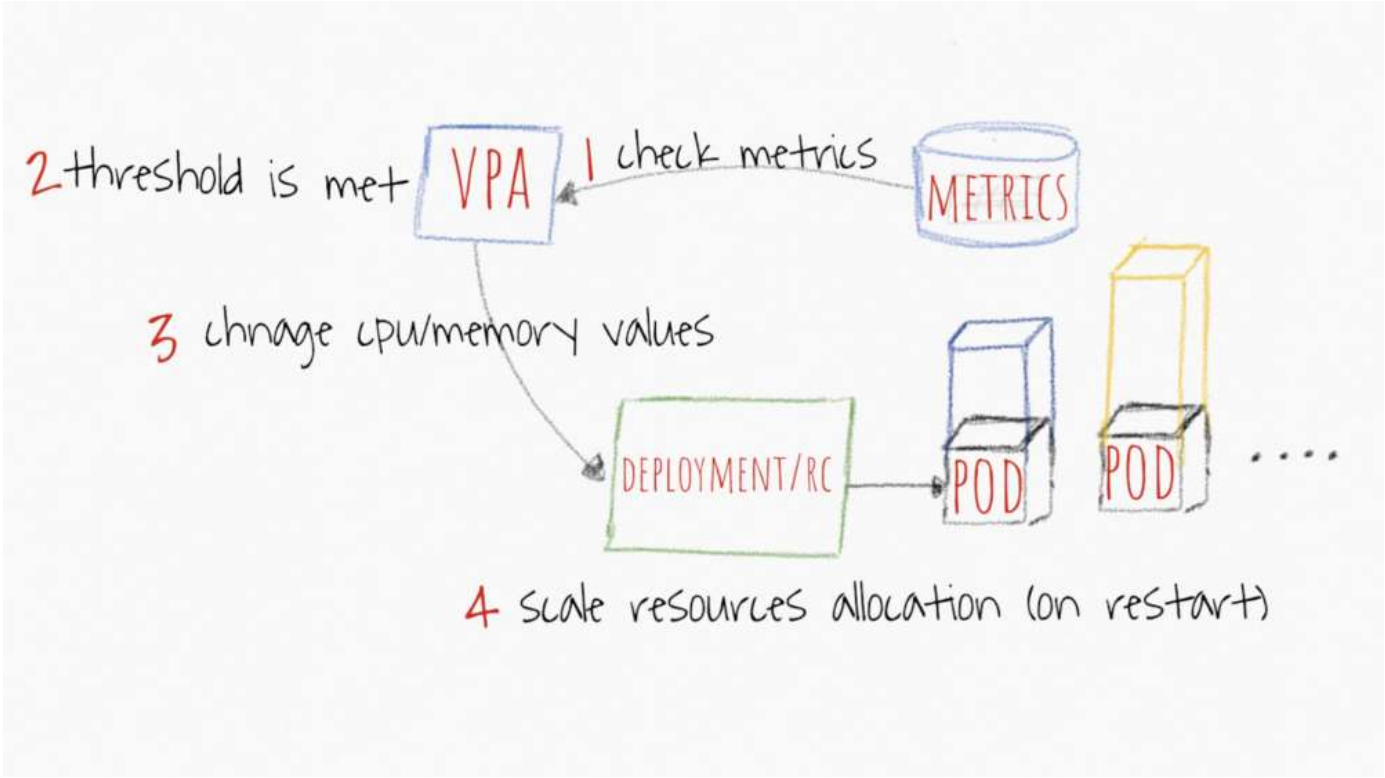
High-level VPA workflow

- VPA continuously checks metrics values you configured during setup AT A DEFAULT 10 SEC intervals

- VPA attempts to change the allocated memory and/or CPU If the threshold is met

- VPA mainly updates the resources inside the deployment or replication controller specs

- When pods are restarted the new resources all applied to the created instances.

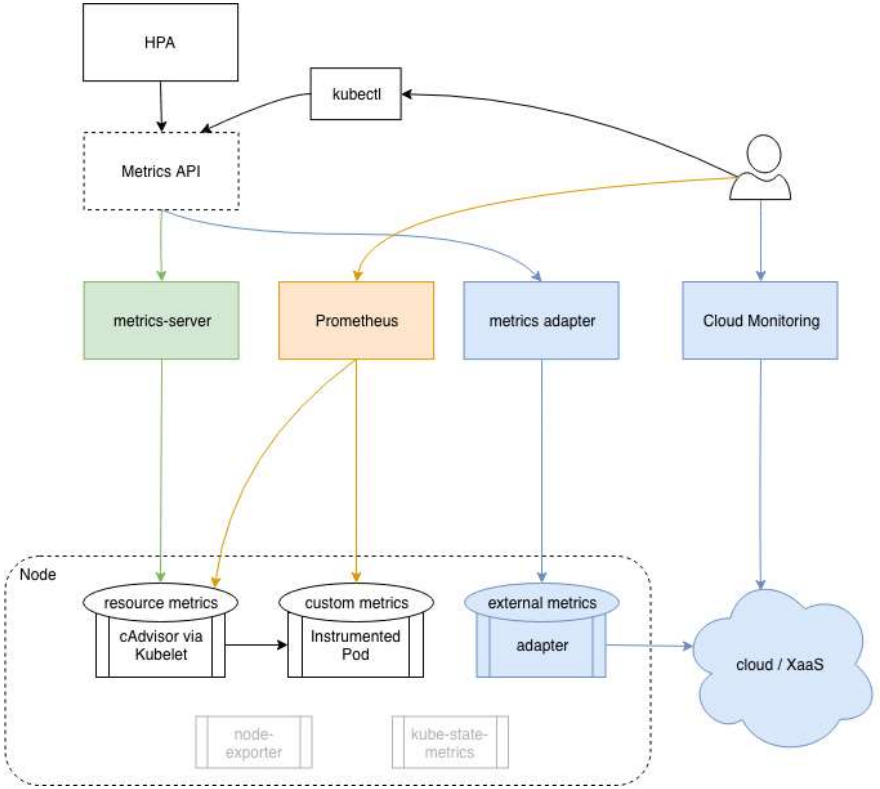


Metrics Server

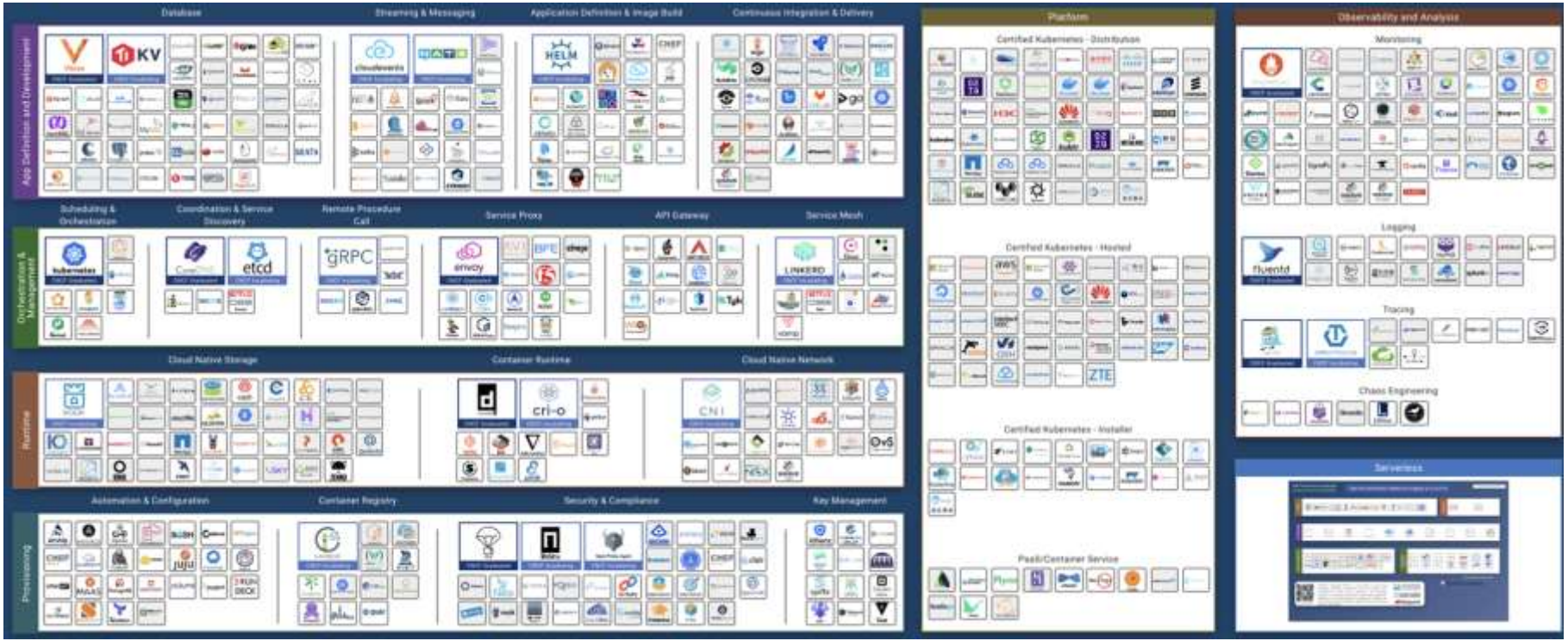
- We still have everything from the basic setup (metrics-server + prometheus)

- We also use the cloud provided monitoring solution.

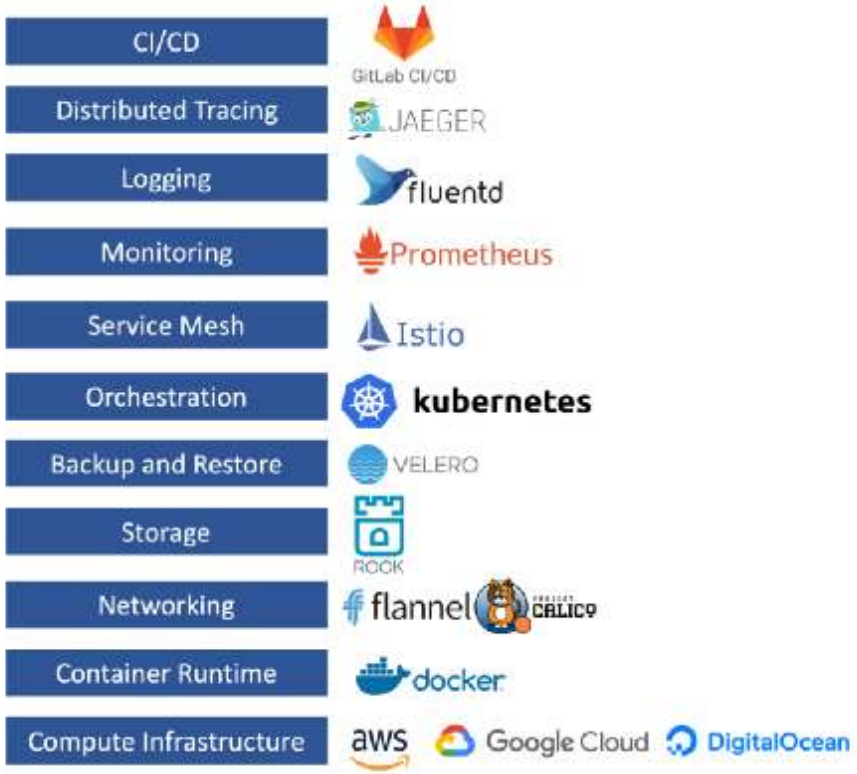
- We have a metrics adapter that exposes additional cloud metrics through the Kubernetes metrics API.



What's next ?


Database
Streaming & Messaging
Application Definition & Image Build
Continuous Integration & Delivery
App Definition and Orchestration
Scheduling & Orchestration
Coordination & Service Discovery
Remote Procedure Call
Service Proxy
API Gateway
Service Mesh
Orchestration & Management
Cloud Native Storage
Container Runtime
Cloud Native Network
Runtime
Automation & Configuration
Container Registry
Security & Compliance
Key Management
Provisioning
Platform
Certified Kubernetes - Distribution
Certified Kubernetes - Hosted
Certified Kubernetes - Installer
PaaS/Container Service
Observability and Analysis
Monitoring
Logging
Tracing
Chaos Engineering
Serverless

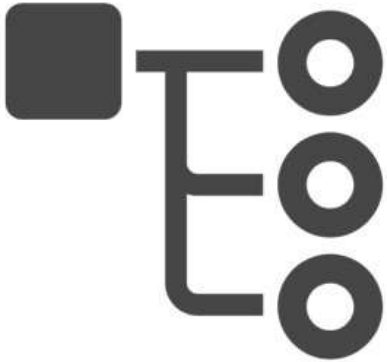
Cloud Native Stack



Observability

Three pillars of observability

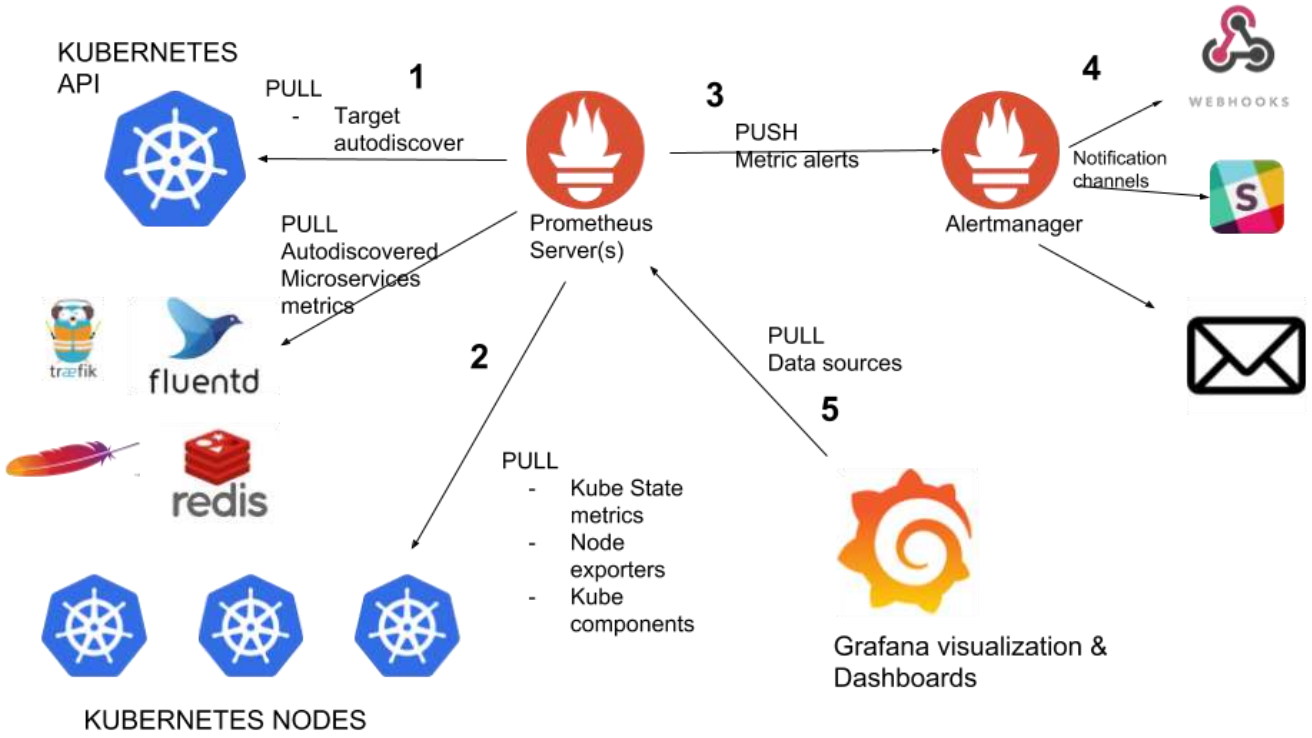
Metrics



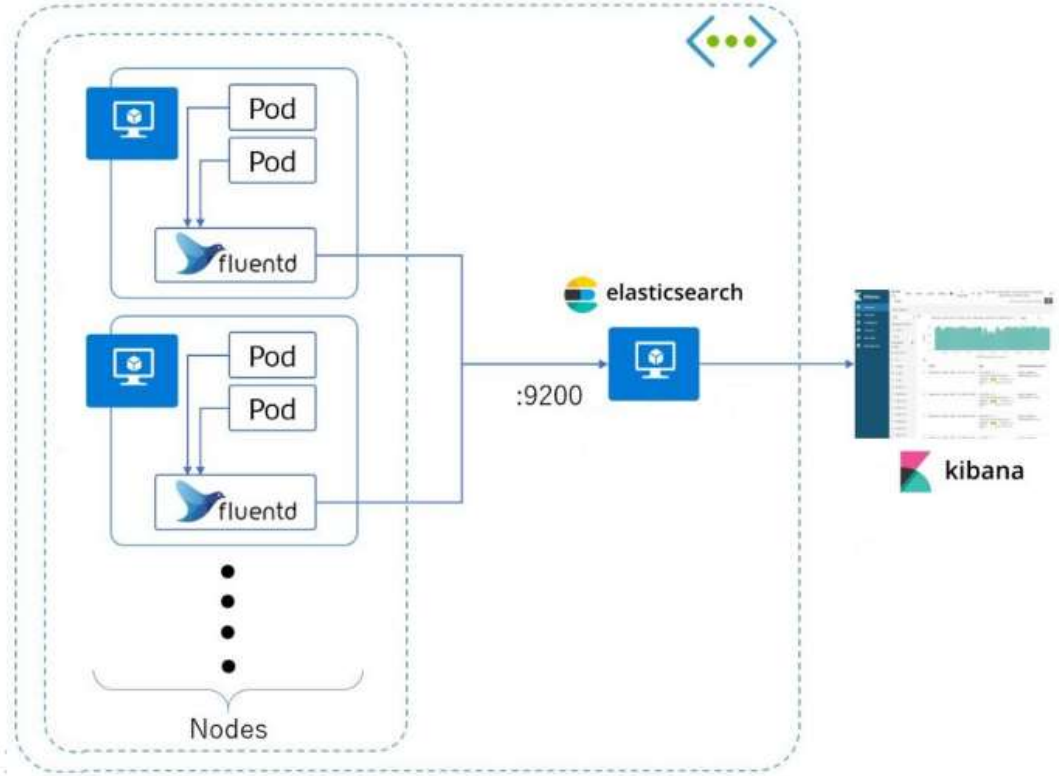
Traces

Logs

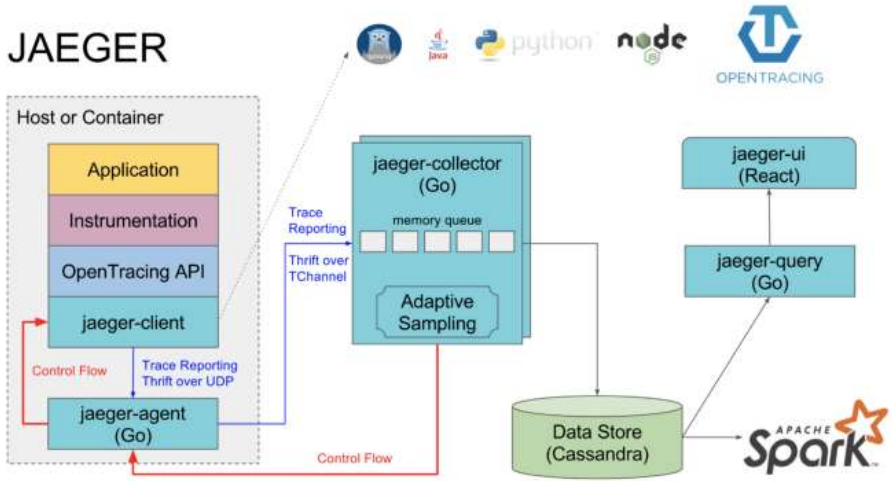
Modern Monitoring on Kubernetes



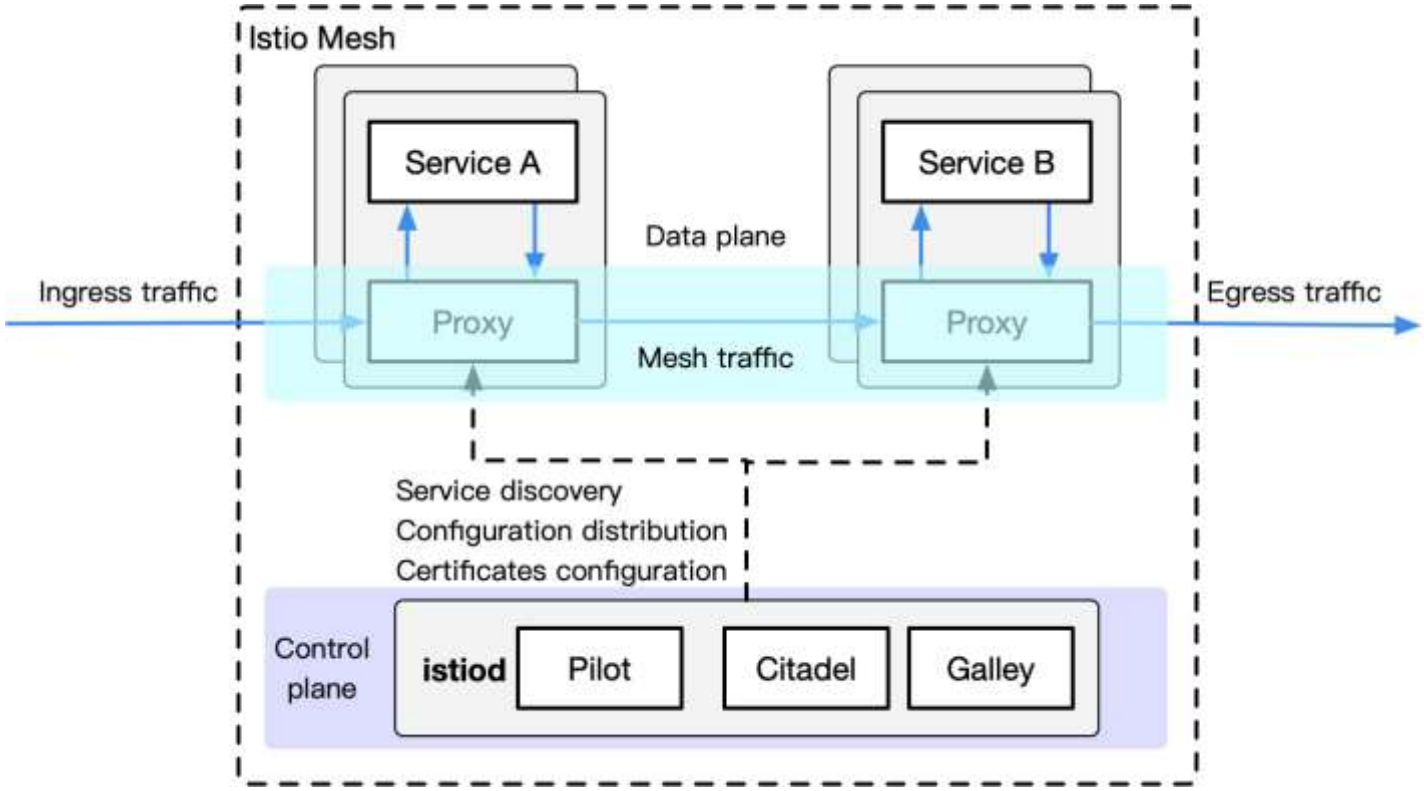
Modern Logging system on Kubernetes



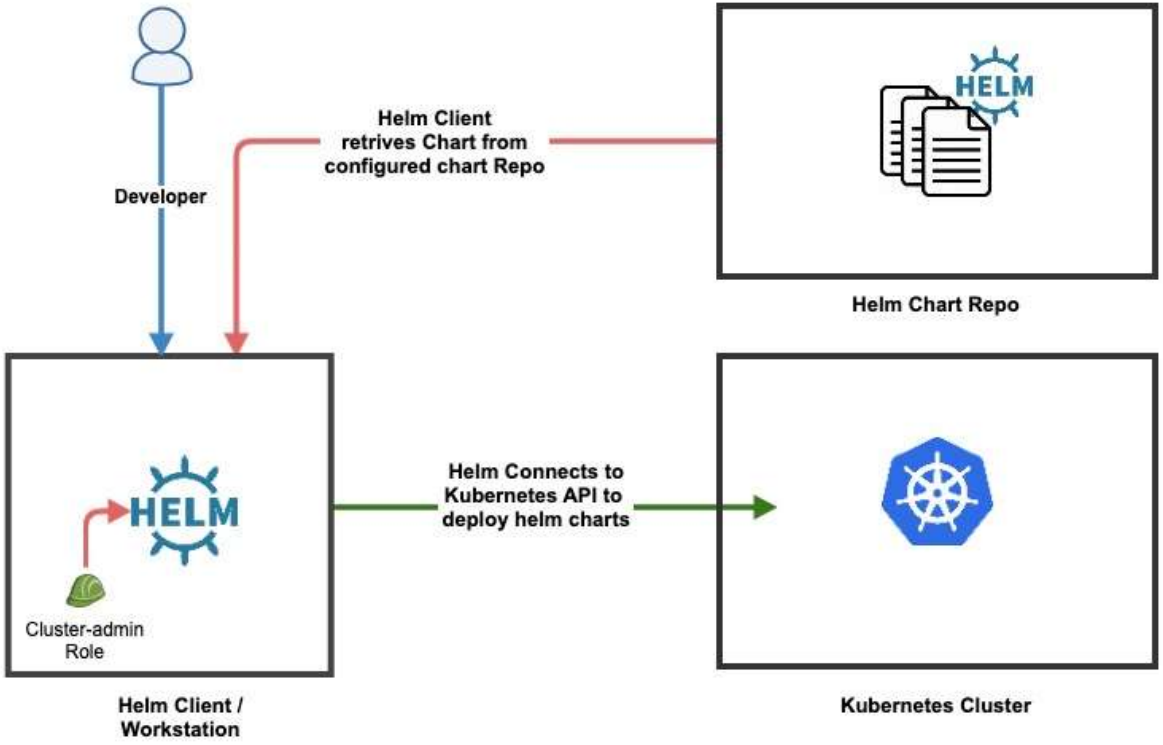
Tracing system on Kubernetes



Service Mesh

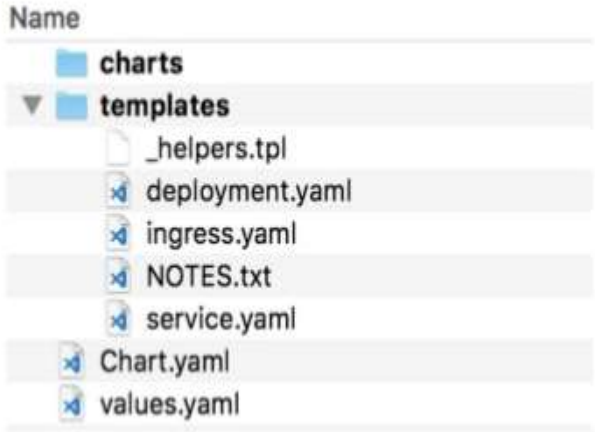


The Package Manager for Kubernetes



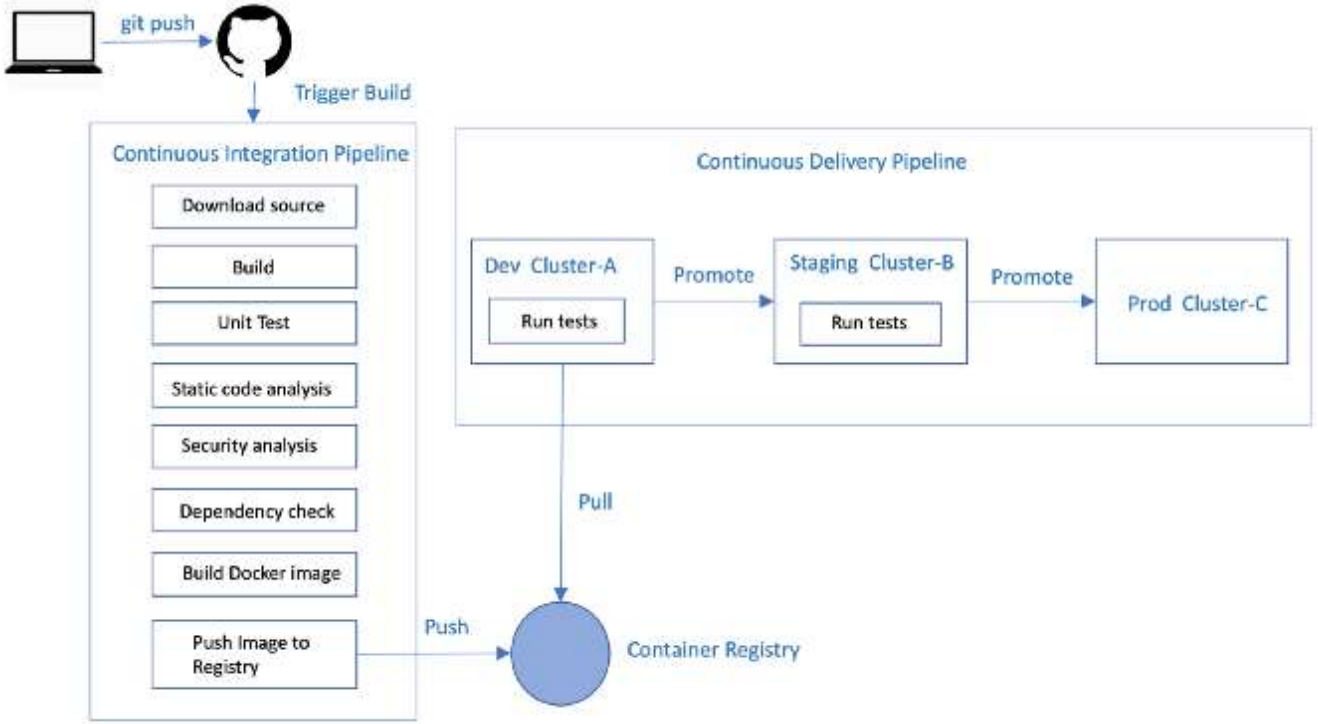
Helm Template

Chart Directory Structure

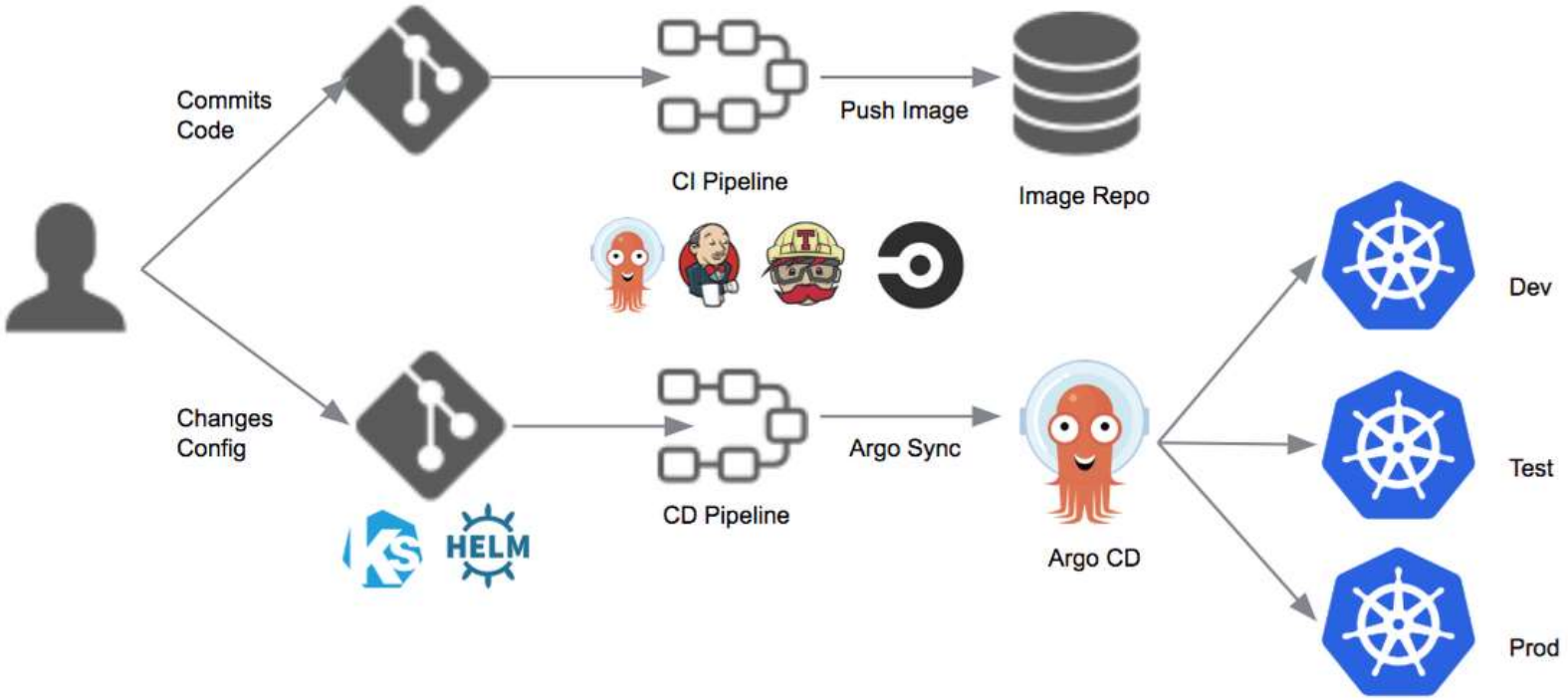



```
kafka:
  replicas: 1
  configurationOverrides:
    num.partitions: 1
    offsets.topic.replication.factor: 1
    log.retention.hours: -1
    log.retention.bytes: -1
    default.replication.factor: 1
    min.insync.replicas: 1
    listeners: PLAINTEXT://:9092,EXTERNAL://localhost:9093,SSL://:9091
    advertised.listeners: EXTERNAL://localhost:9093,SSL://:9091
    listener.security.protocol.map: EXTERNAL:PLAINTEXT,PLAINTEXT:PLAINTEXT,SSL:SSL
    inter.broker.listener.name: SSL
    ssl.client.auth: required
    ssl.enabled.protocols: TLSv1.2
    ssl.keystore.filename: kafka.keystore.jks
    ssl.truststore.filename: kafka.truststore.jks
    ssl.key.credentials: pw.txt
    ssl.keystore.credentials: pw.txt
    ssl.truststore.credentials: pw.txt
  kafkaHeapOptions: "-Xmx512M -Xms512M"
  secrets:
  - name: kafka-jks
    keys:
      - kafka.keystore.jks
      - kafka.truststore.jks
      - pw.txt
    mountPath: /etc/kafka/secrets
  resources:
    requests:
      memory: 50Mi
      cpu: 50m
    limits:
      memory: 1Gi
      cpu: 500m
  zookeeper:
    replicaCount: 1
    resources:
      requests:
        memory: 50Mi
        cpu: 50m
      limits:
        memory: 500Mi
        cpu: 500m
    env: {ZK_HEAP_SIZE: "512M"}
```

CI/CD workflow



GitOps Workflow




Q
A