Production Cluster Monitoring and Remediation for High Reliability at eBay

钱世俊, Cloud Software Engineer, ebay @danielqsj
刘应科, MTS1, Cloud Software Engineer, ebay @keyingliu
Agenda

Growing Clusters

Monitoring

Remediation

Q&A
Growing Clusters

30+ Clusters

8K+ Nodes (BMs+VMs)

100K+ Pods
Monitoring Goals

- Control Plane Management
  - Apiserver
  - ETCD
  - Scheduler
  - Controller

- Data Plane Management
  - Node Lifecycle Management
  - Pod Lifecycle Management
  - Daemonset / Deployment / Service / Ingress ...

- Alert Management

- AIOps
Monitoring Overview

- Alerts
- Logging
- Metrics
- Automation
- AIOps
How we logging

Daemonset: Filebeat Logrotate

Kube API Server

Ingress Service

Resource Manager

Flink

Kafka

Egress Service

Storage SDK

elastic

HDFS
How we collect metrics

Cluster Prometheus A

- K8S Key Components
- Exporter

Control Plane Management
- Apiserver Latency
- Scheduling Latency
- IP Allocation Latency
- ETCD Latency
- ETCD disk usage
- Namespace Resource Usage
- ... (omitted)

Cluster Prometheus B

- Node Problem Detector

Node Lifecycle Management
- NotReady Nodes: Amount and Timestamp
- SchedulingDisabled Nodes: Amount, Timestamp and Reason
- Cpu, Memory, Disk usage
- Network Status
- PID, FD status
- ... (omitted)

Cluster Prometheus C

- Kube State Metrics
- Assertion

Pod Lifecycle Management
- Pod Creation Latency
- Pod Terminating Latency
- Pod Restart Times
- Pod Resource Usage
- Container Creation Latency
- Container Terminating Latency
- Container Exit Status
- ... (omitted)
Assertion

simulate workloads

real time analysis of workloads

Cluster Health

- api-server
- node
- etcd
- ipam
- ecr
- netperf
- vip-usage
- kubeproxy-healthz
- pod-connectivity
- pod-failure
- cross-cluster
- more...

Assertion
How we build dashboards

Cluster Dashboards
- Apiserver
- ETCD
- Node
- Namespace
- Pod
- Service
- Ingress
- Storage
- Network
- Capacity
- ...

Global Dashboards
- Global Health
- Global Cluster Capacity
- Global Alerts
- Components Version
- ...

Cluster Prometheus

Federated Prometheus
Global Health Dashboard
Cluster Health Dashboard
Node Health Dashboard

Uptime: 8.5 weeks
Node Status: Ready

Latest Status Change Time:
- Current: 2018
- Month: 10
- Day: 15
- Hour: 2
- Minute: 36

Latest Status Change Duration: 2.032 weeks
NTP Clock Offset: 269 µs

Basic CPU / MEM / DISK USAGE:

- CPU Busy: 4%
- Used RAM Memory: 22%
- Used Root FS: N/A
- CPU System Load (1m avg): 4
- CPU System Load (5m avg): 4
- CPU System Load (15m avg): 4
How we manage alerts

Alert Rules
- Labels
  - Component
  - Severity
- Annotations
  - Description
  - Summary
  - Runbook
  - Cluster

Cluster Alert Dashboard
Global Alert Dashboard

Pagerduty

Jira
- RCA
- Execution Plan
- Time Consumption

Grafana

KubeCon CloudNativeCon China 2018
Global Alert Dashboard

- Current Alerts: 92
- Critical Alerts: 0
- High Alerts: 0

Alerts:
- Assertions Alerts: 49
- DNS Alerts: 0
- DNSLoad Balancer Alerts: 5
- IngressGateway Service Alerts: 1

Alerts:
- Ingress Alerts: 12
- Networking Alerts: 1
- Nodes Alerts: 24

Alert Trends

Alert List:
- Horizontal| Beta, Kubernetes| IngressGateway| DNSLoad Balancer
- Node| Assert
- IngressGateway| Assert
- DNSLoad Balancer| Assert
- Node| Assert
- IngressGateway| Assert
- DNSLoad Balancer| Assert
How to audit across clusters?

How to execute complex queries quickly?
KubeWatch Architecture

API Server

Resource Update

Changes

kubeWatch

Storage Provider

Elasticsearch

Ceilometer UX

Browser

PostgreSQL

pg UI Client

Query
Query Example:

1. **Get all pods for namespace kube-system**
   ```sql
   SELECT
     name AS podName,
     data->'metadata'->>'namespace' AS namespace,
     data->'spec'->>'nodeName' AS nodeName,
     meta,
     data
   FROM pods
   WHERE data->'metadata'->>'namespace' LIKE 'kube-system'
   AND deleted = FALSE;
   
2. **Get all services of type load balancer**
   ```sql
   SELECT
     name AS serviceName,
     data->'spec'->>'type' AS type,
     data->'metadata'->>'namespace' AS nameSpace
   FROM svcs
   WHERE data->'spec'->>'type' LIKE 'LoadBalancer';
   ```
Monitoring Automation

- Rollout alert rules
- Rollout monitoring configurations
- Git Driven

1. Get newest rules / configuration
2. Get runtime rules / configuration
3. Compare rules / configuration
4. Update rules / configuration and Reload
AIOps

- Real-time analysis and alerts
- Reducing MTTD and MTTR
We have

- More than 30 clusters
- Thousands of nodes
- Including both BMs and VMs
Overall

- Node Problem Detector
- Memory Status
- Disk Status
- Docker Status
- Compute Node
- Remediation Controller
- Hardware
- Software
- Reparo
- Nodes
Hardware Failures

Sensors (Non intrusive detection)
- TEMP
- CPU
- PSU
- MEMORY
- VOLT
- HDD
- FAN

- Define each pattern to correspond with each known failure
- More patterns can be defined if new failure found
- Check if the failure can be tolerant
- Mark the hardware as failures
- Get notified if hardware issues have been fixed

In OS (intrusive detection)
- Kernel message
- MCE message
- Disk check
Software Failures

Health check for key components:

Kube*

Key Services

Configurations

Kernel soft lockup

Container Runtime

etc.
Thank You !