aiven

Operating Kafka in the Cloud

Chicago Area Kafka Enthusiasts

Heikki Nousiainen
2019-04-06
Chicago, IL

Agenda

1. Introduction
2. Aiven & our architecture
3. Monitoring
4. Automation
5. Partition Placement
6. Kafka Versions
7. Q & A

Speaker

- Heikki Nousiainen
- CTO, co-founder @ Aiven
- First contact with Apache Kafka in 2014

@hnousiainen

Google Cloud Platform

amazon web services

Microsoft Azure

UpCloud

DigitalOcean

packet

Aiven

- *Your data cloud*
- Based in Helsinki and Boston
- 8 data engines now available in 6 clouds and 80 regions, virtual and bare metal instances
- Launched a fully-managed Kafka Cloud service in 2016
- Other services: PostgreSQL, MySQL, Elasticsearch, Cassandra, Redis, InfluxDB

Google Cloud Platform

amazon web services

Microsoft Azure

UpCloud

DigitalOcean

packet

Kafka @ Aiven

- We utilize Kafka as core message bus between all of our components
- Signaling, Metrics, Logs
  - Configuration/state communication
  - Logs from VM to Kafka, from Kafka to ES
  - Stats from VM to Kafka, from Kafka to InfluxDB / M3
- Why Kafka? Needed a fault tolerant signaling for Postgres failovers

Google Cloud Platform

amazon web services

Microsoft Azure

UpCloud

DigitalOcean

packet

Kafka @ Aiven

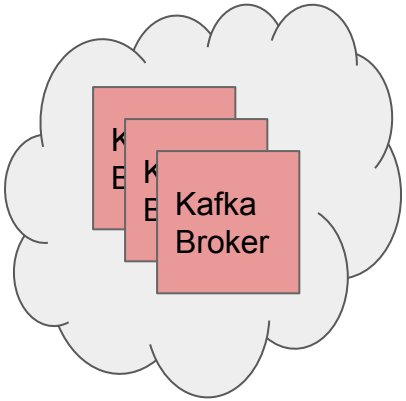
- Managed Kafka Offering in 2016

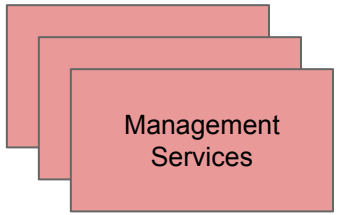
- 100s of Kafka Clusters
- 1000s of Brokers
- Median cluster 600 messages / second
- Median message size just over 200 bytes
- Typical cluster cost: $660 / month

Aiven Architecture

Aiven Architecture

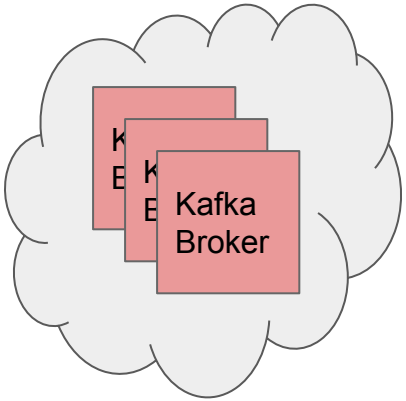
- Kafka Brokers implemented as dedicated VMs
- Backed by management layer responsible for provisioning the cluster resources

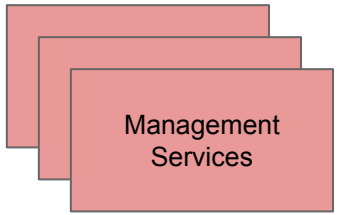




Aiven Architecture

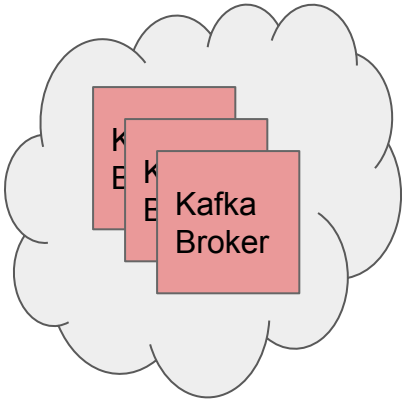
- Kafka Brokers managed by agent software running within VM
- Per cluster ZK co-hosted on same VMs
- Agent responsible for setting up and managing both ZK and Kafka
- Immutable; software updates via VM replacement

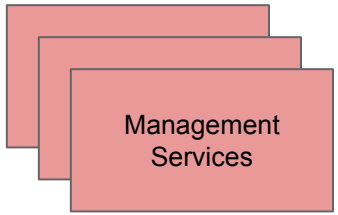




Aiven Architecture

- Management layer provisions and provides resources to the cluster.
- Management layer monitors and ensures cluster meets the specification: number of brokers, size of brokers, correct software version, broker health
- Availability Zones for fault tolerance





Monitoring

Monitoring

- We monitor Kafka clusters by reading JMX via Jolokia
- Collected by telegraf and sent to centralized store
- Key metrics for alerts:
  - UnderReplicatedPartitions (*)
  - IsrShrinksPerSec
  - RequestHandlerAvgIdlePercent / NetworkProcessorAvgIdlePercent
  - LogCleanerManager:time-since-last-run-ms
- If you have stable workloads, consider low/high message rates
- Implement paging and/or automatic corrective responses

Monitoring

- Metrics generated from Kafka log / journald
- Exceptions and Errors
  - ReplicaFetcher errors
  - LeaderAndIsr requests
- Can be noisy!

- We're looking to contribute in this area

Monitoring

- ZooKeeper
  - /brokers/topics/...
  - /admin/reassign_partitions
  - /admin/delete_topics/ - if you're using ZK for topic management
- Health checks to API
  - metadata call
  - response times

Automated Actions

Applying Configuration Changes

- There are more and more settings that can be configured dynamically
- Some Kafka settings require broker restart
  - auto.create.topics.enable
  - offsets.retention.minutes

- Local agent restarts Kafka process as needed
- Interlocking / synchronization via ZK
  - Avoid restarting multiple brokers at once

Corrective actions

- Kafka broker sometimes falls off sync with the cluster.
- Restart forces full state reload, and is thus often one tool in bringing misbehaving broker back

- Again, careful not to restart more than one broker at a time

Active Partition Placement

Active Partition Placement

- Agent performing continuous monitoring for partition placement
- Adjusts placement automatically:
  - To satisfy redundancy requirements & data distribution
  - To balance storage usage
  - To balance partition leaders

Process:

- Create and maintain a continuously updated plan in ZK
- Correct deviations from the plan
- Throttle the amount of inflight partition movements
- Partition management instructions via ZK /admin/reassign_partitions

Partition Placement - Monitoring

- Monitor progress via /admin/reassign_partitions
- Sometimes gets stuck:
  - Ever tries to implement reassignment on a broker that's gone
  - Failed partition leadership assignments

- On timeout, force controller re-election
- Hoping to investigate and contribute fixes in Kafka

- Beware of UnderReplicatedPartitions
- We filter UnderReplicatedPartitions vs. ISR >= number of desired replicas

Partition Placement - Cluster Changes

- Drain nodes that are about to be decommissioned
  - Maintain the specified level of redundancy
  - Maintain correct AZ spread
- Ensure clients get negative reply (not_leader_for_partition) and reload metadata
- Building block for cluster level actions

- Our placement logic is part of the overall management agent
- My hope is to separate it & publish it as Open Source

Cluster Level Corrective Action

Broker failures

- Management services detect loss of Kafka brokers
- Creates new VM resource for the cluster
- Assigns a new Broker ID

- Agent adjusts the ZK cluster memberships
- Partition placement uses the new resource and restores replication levels

Positive cases

- Management services detect:
  - Required security software updates
  - Requested Kafka version updates
  - Mismatching VM types - scaling
  - Mismatching VM number - scaling
  - Mismatching VM locations - migration

- Create a single VM
- Partition manager drains an existing VM
- Old VM recycled
- Loop until complete

Chaos Engineering

- Practice makes perfect
- Kafka is critical component in our software stack
- We initially run our Kafka clusters using Google Cloud Platform preemptible VM instance types
- The instances can be terminated at any time, and will be terminated after 24 hours

Kafka Versions

Kafka Versions

- 0.8.2:
  - no TLS
  - ZK based Consumer Groups
  - Regular Kafka Broker restarts required
- 0.9.0: Aiven goes live with Postgres Service
  - TLS support
  - Consumer group management over Kafka protocol
  - Still rather unstable
- 0.10.2: Aiven starts to offer managed Kafka service
  - Aiven Kafka service into production
  - Authentication and ACL suppot
  - Relatively stable when handled with care

Kafka Versions

- 0.11.0:
  - new more efficient message record format
  - record headers
  - idempotent produce & transactions
  - 0.11.0 did come with a host of quality and stability problems
  - Stabilized with 0.11.1 and finally 0.11.2.
- 1.0.0:
  - SASL authentication
  - Memory and resource leaks, fixed completely in 1.0.2
  - With 1.0.2, solid & stable
- 1.1.0:
  - Stability & performance under load improves

Kafka Versions

- 2.0:
  - A lot of fixes in the replication protocol in error conditions
  - Significant performance update
  - Quota management
- 2.1:
  - TLA+ model used for hardening the replication protocol
  - Java 11 support

- General rule of thumb, If you have any issues with stability / performance, upgrade your Kafka version

Thank you!

@hnousiainen
htn@aiven.io

https://aiven.io