aiven

Scaling Kafka in the Cloud

Meetup, Credorax Group

Heikki Nousiainen
2019-09-03
Tel Aviv

Speaker

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

@hnousiainen

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

- Managed Kafka Offering since 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

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

Our Challenges & Constraints

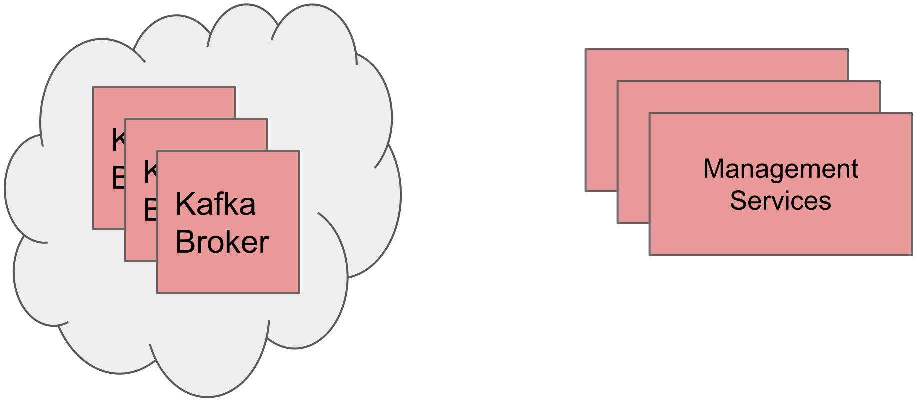
- Six supported cloud providers
- Immutable infrastructure
- Keeping software up-to-date
- Adjustable cluster size and broker specs
- Migration between regions and clouds
- High availability, SLAs

- Changing instances, IP addresses
- Ephemeral disks
- Roll-forward updates

Aiven Architecture

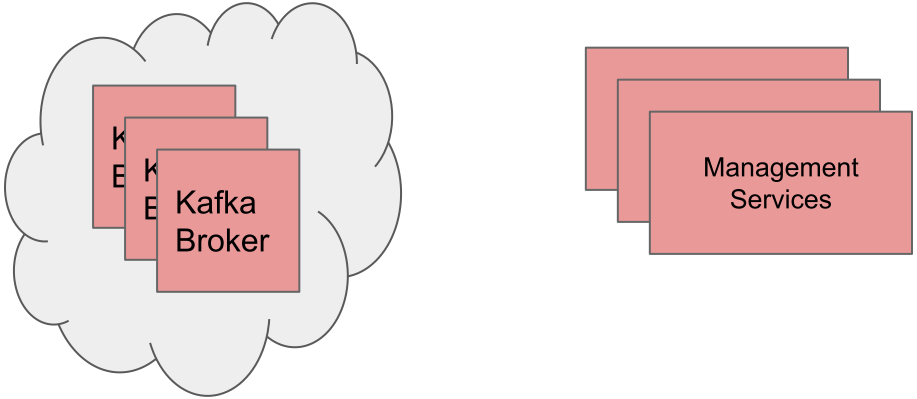
Aiven Architecture

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



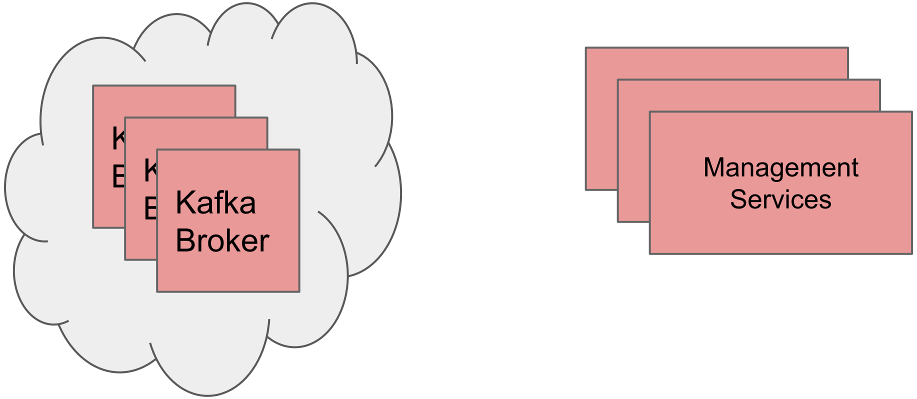
Aiven Architecture

- 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



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

- Intra-cluster IPv6 overlay network
- Built using IPSec tunnels over either public or private instance IP addresses

- Consistent addressing regardless of the VM location / networking capabilities

Building Blocks

ZooKeeper

- Distributed, consensus based synchronization & coordination service
- Used by Kafka for controller election and cluster state management

- ZooKeeper 3.5 branch supports dynamic reconfiguration
- Released May 20th, 2019
- In use in Aiven for 3+ years

ZooKeeper

- Dynamic reconfig requires quorum

- Fault recovery adds new node(s), removes disappeared ones
- Roll-forward upgrade adds new node(s), removes obsoleted nodes

- Both failure and positive cases handled using the same procedure

Kafka 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

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

Kafka Corrective Actions - Broker

- Kafka broker sometimes falls off sync with the cluster
- Also, thread deadlocks

- Restart forces full state reload
- Can be triggered automatically

- For availability, must not to restart more than one broker at a time

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 - 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

- Our placement logic is part of our per VM management agent
- My hope is to separate it & publish it as Open Source

Kafka Corrective Actions - Cluster

- 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

Kafka Cluster Operations

- 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 new VM
- Partition manager drains an existing broker - data migration
- Old broker 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

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

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

Thank you!

@hnousiainen
htn@aiven.io

https://aiven.io