Elastic Data Streaming

Autoscaling Apache Kafka

Jakub Scholz

KubeCon NA 2024

About me

- Senior Principal Software Engineer @ Red Hat
- Maintainer of Strimzi project (https://strimzi.io)
- Occasional Apache Kafka contributor

@scholzj

https://github.com/scholzj

https://www.linkedin.com/in/scholzj/

STRIMZI

What is Strimzi?

- CNCF Incubating project
- Open Source community (Apache License 2.0)
- Focuses on Apache Kafka on Kubernetes
  - Based on the operator pattern
  - Provides operators for running and managing Apache Kafka and its components
  - Additional tools to make Apache Kafka easier to use on Kubernetes

STRIMZI

Why auto-scale Kafka brokers?

- Kafka is often a big workload
  - Consumes a lot of resources
  - Big potential to optimize costs, energy consumption, ...
- Apache Kafka is not very elastic
  - Ability to scale Kafka brokers up and down according to immediate demand
  - **=> We tried to improve this!**

STRIMZI

Demo

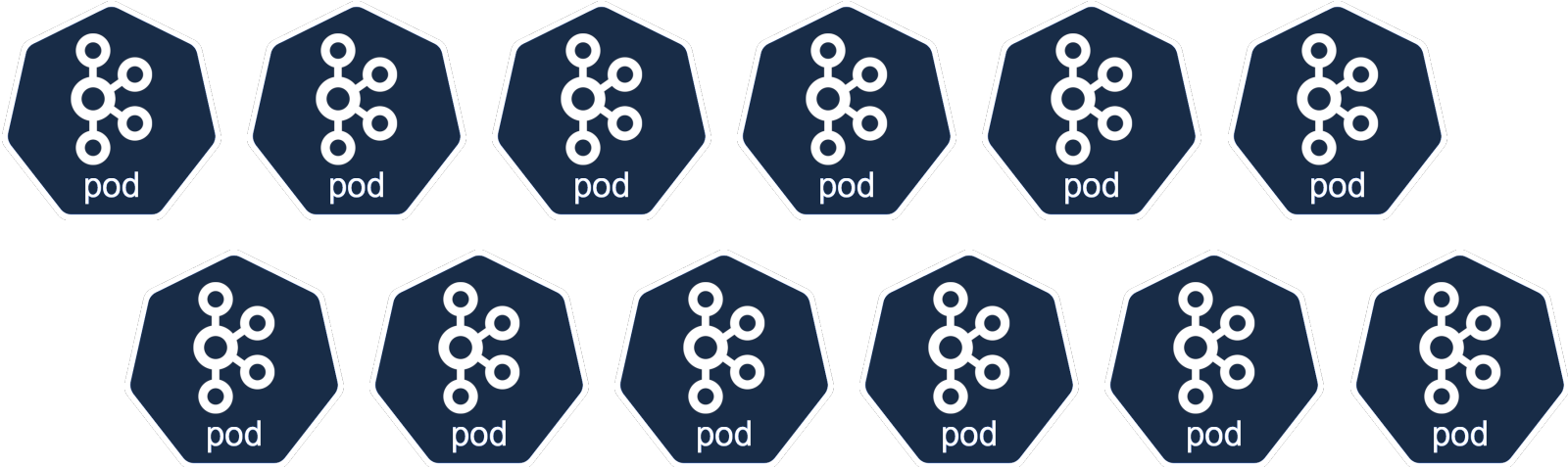
Start the demo ...

STRIMZI

Kafka and Scalability

STRIMZI

Kafka and Scalability



STRIMZI

Kafka and Scalability

STRIMZI

Kafka and Elasticity

STRIMZI

Kafka and Scalability

- Apache Kafka is a highly scalable distributed data streaming platform
  - Scalability = Ability to scale Kafka brokers with your project/company
- Apache Kafka is not very elastic
  - Ability to scale Kafka brokers up and down according to immediate demand
  - **=> We tried to improve this!**

STRIMZI

Auto-scaling

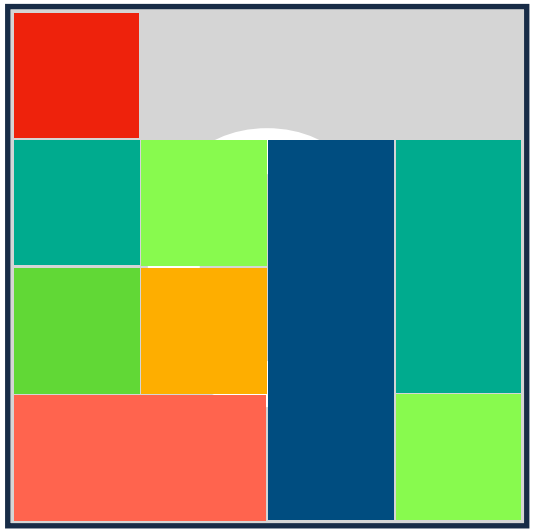
- Requires the Kubernetes scale subresource
  - Allows to scale the custom resource without a deep understanding of the structure
    - Can be used with `kubectl scale`
    - Or with Horizontal Pod Autoscaler

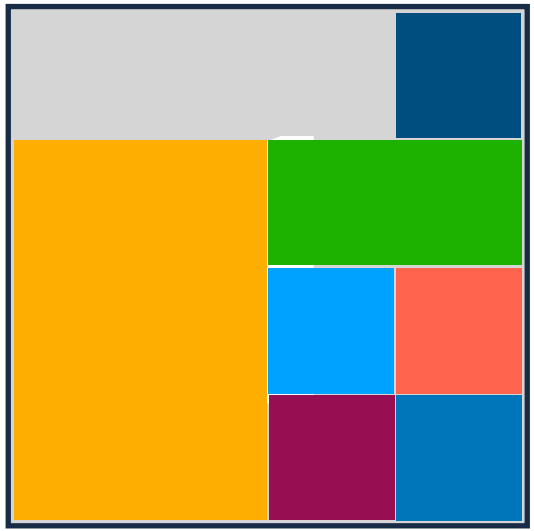
STRIMZI

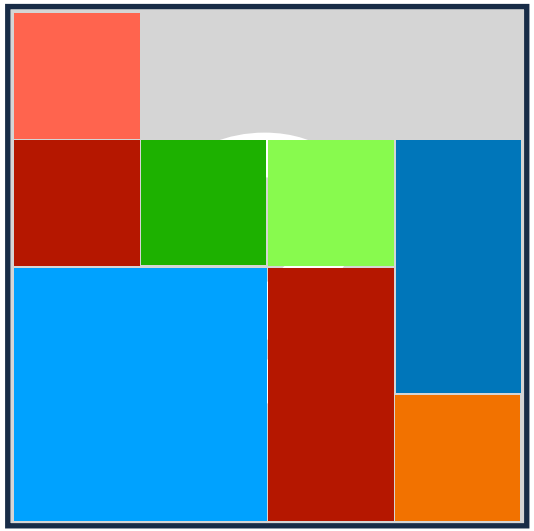
What happens ...

... when you scale-up?

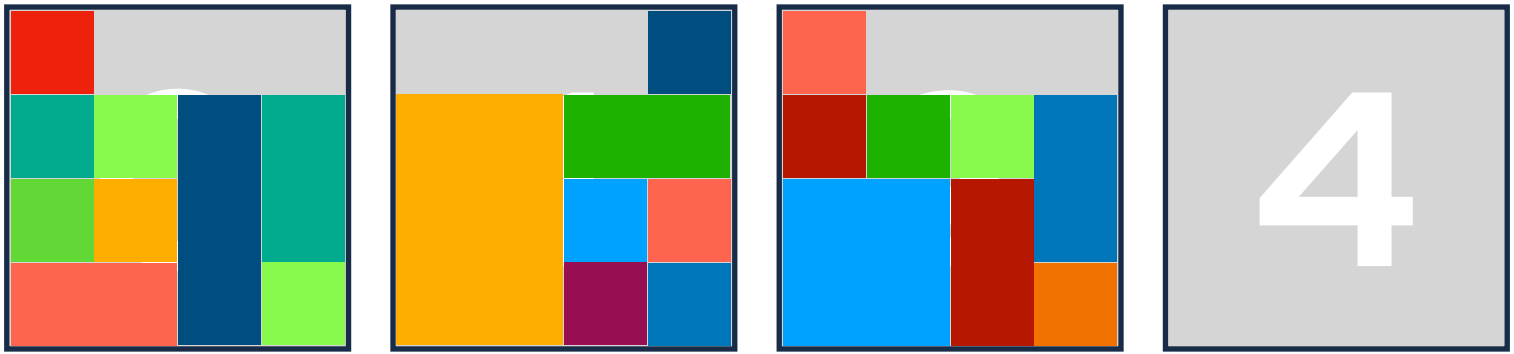
STRIMZI







STRIMZI

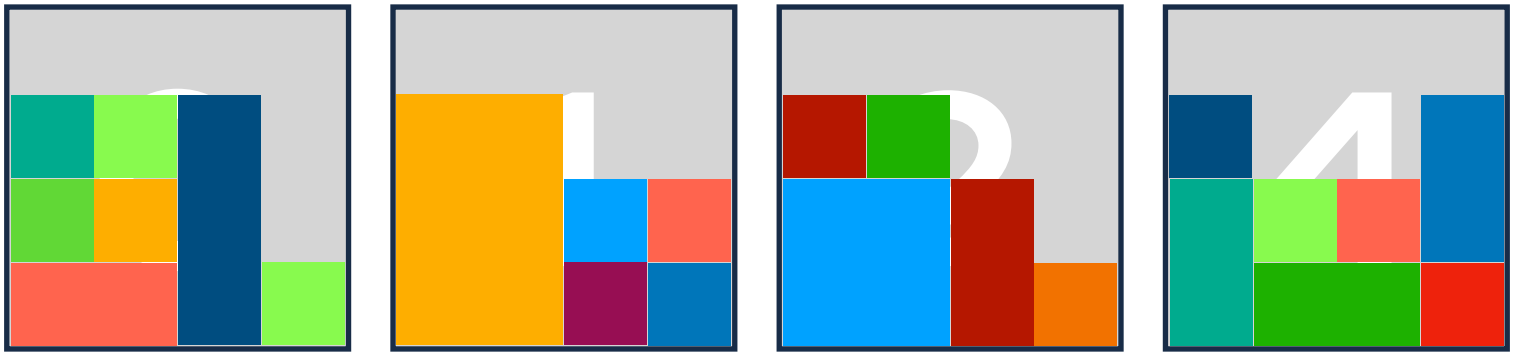


STRIMZI

What happens ...

... when you scale-down?

STRIMZI



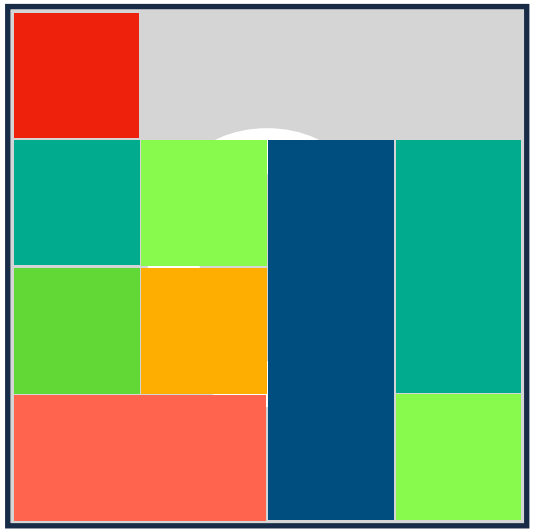
STRIMZI

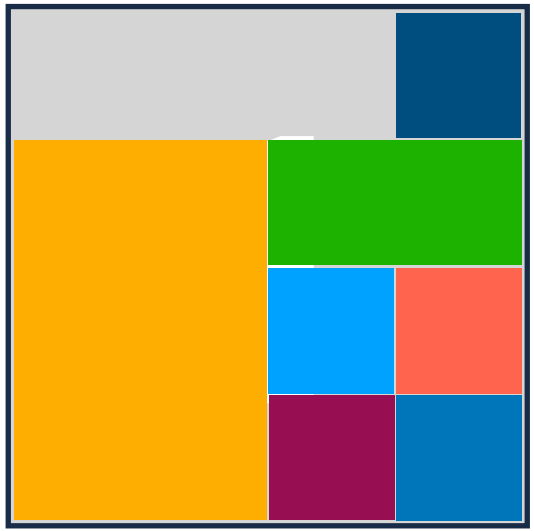
Auto-rebalancing

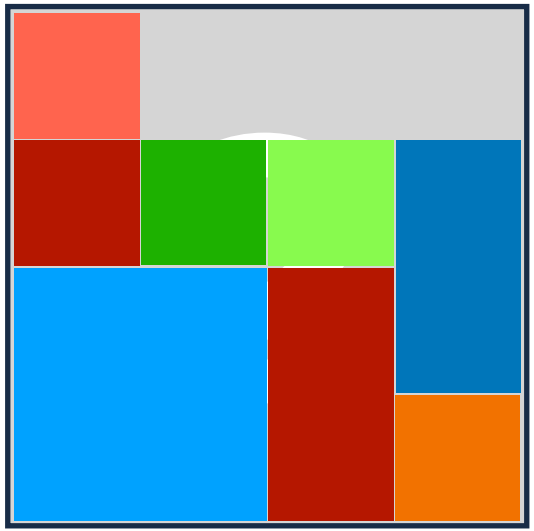
- We have to automatically rebalance the cluster after it scales
  - Configurable in the Kafka custom resource
    - Users can configure the rebalancing details (what to optimize for)
  - At scale-up, Strimzi automatically rebalances the brokers to shift the load to them
  - At scale-down, Strimzi will move the data before removing the nodes

What happens now?

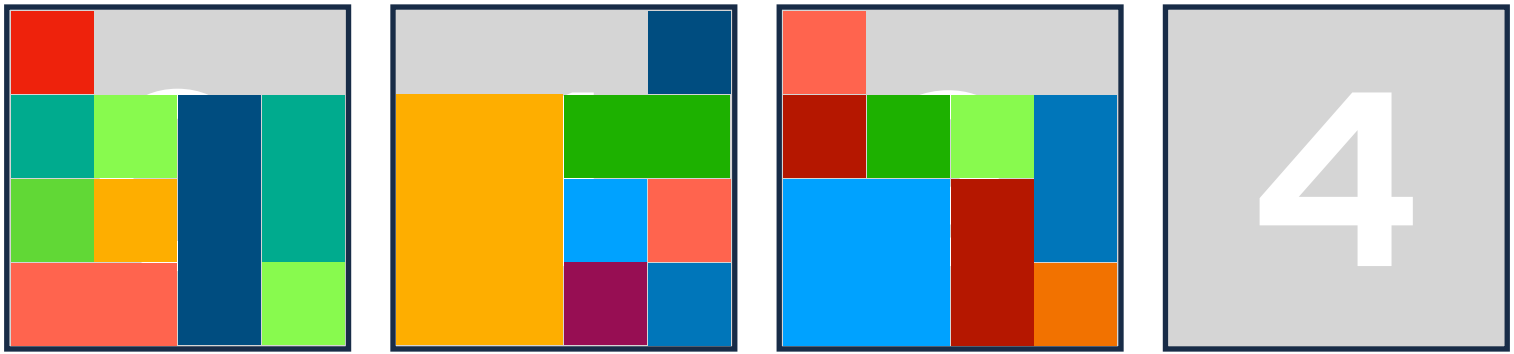
STRIMZI



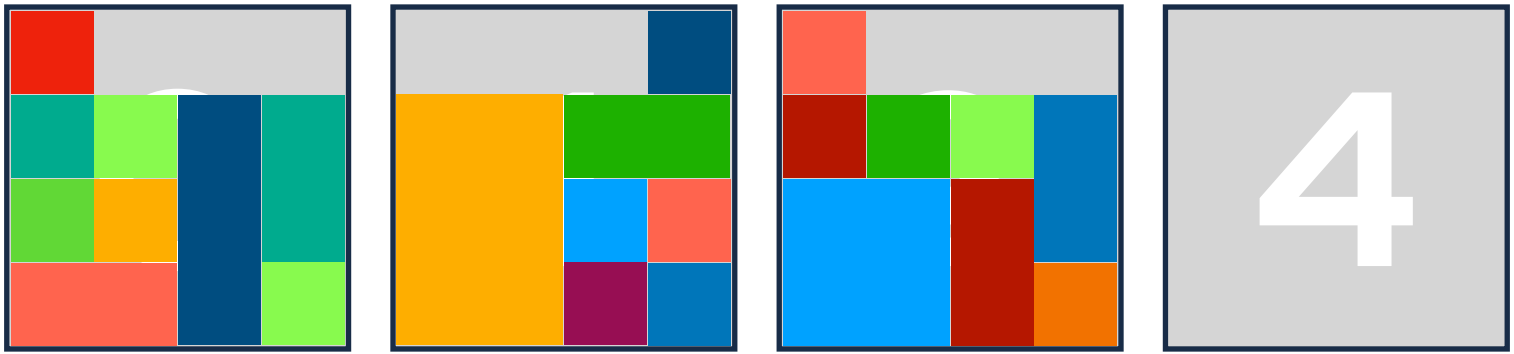




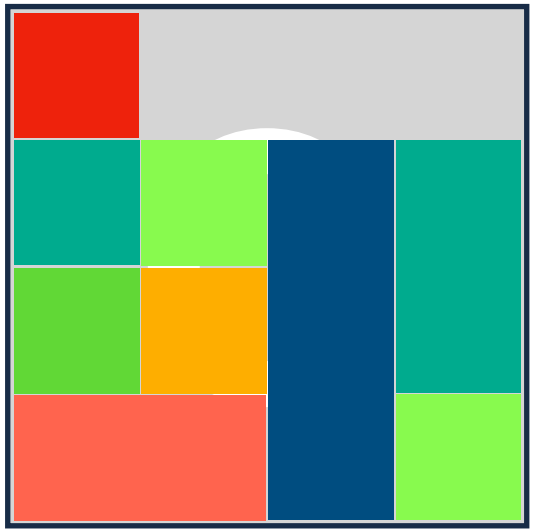
STRIMZI

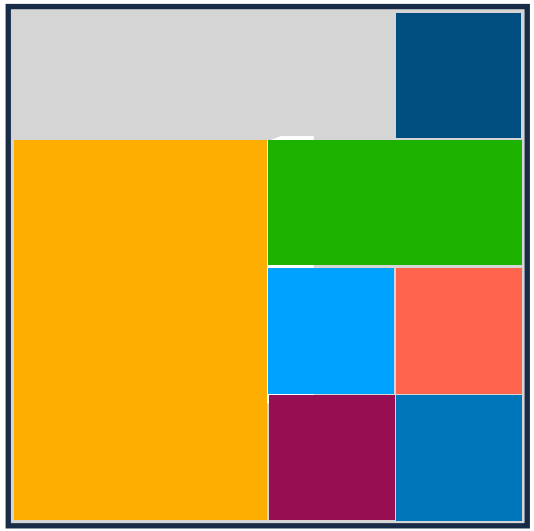


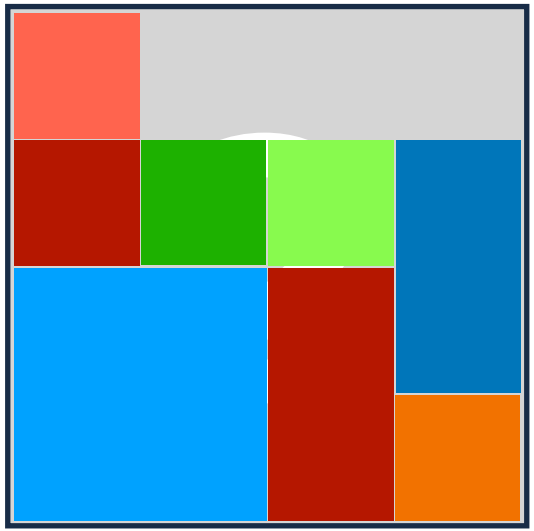
STRIMZI



STRIMZI







STRIMZI

What is happening under the surface?

STRIMZI

STRIMZI

Tiered storage

- Stores different data in different storage tiers
  - The latest data and data that are often accessed are kept in block storage
  - Older and less important data are offloaded to another tier
    - This tier is typically cheaper, has higher capacity, and lower performance
    - Typically object storage or shared file storage
- Reduces the amount of data stored on the broker
- Less data need to be moved while rebalancing during scale-up/scale-down

300Gi

300Gi

300Gi

300Gi

STRIMZI

Demo

... did it scale?

STRIMZI

Demo

http://jsch.cz/autoscaling

STRIMZI

Remaining challenges

- Scaling is expensive
  - After the brokers are scaled (replicas count is changed), data needs to be moved
  - This requires additional resources => the load increases before it settles
  - For example:
    - You scale down because the load is small
    - Moving the data between brokers increases the load right away
  - This requires careful timing of the auto-scaler

Remaining challenges

- Moving data is slow
  - And even with tiered storage, you might need to move a lot of data
  - Full compute/storage separation might help, but it works well only for some use cases
- Tiered storage is not for everyone
  - Not every Kafka use case is a good fit for tiered storage

Remaining challenges

- What is the right metric to scale on?
- Do we need a Kafka-aware autoscaler?
  - Does the cluster have enough partitions and is the load well distributed for the scaling to be effective?
- Do you have free capacity?
  - Kafka brokers often run on dedicated nodes
  - The dedicated nodes can be auto-scaled as well, but it adds additional time needed to scale-up

Result

- You can auto-scale Kafka, but …
  - Scale early to be ready to handle the load
    - Don’t scale up when you have 100% utilization, but already much earlier
  - More suitable for mid- and long-term scaling
    - Scale down for weekends/nights or as your project grows
- Sizing Kafka is hard …
  - Auto-scaling can help to reduce overprovisioned Kafka clusters

Interested in auto-scaling your Kafka cluster?

Let us know and help us continue this journey!

STRIMZI

Other talks

- Strimzi and the Future of Apache Kafka on Kubernetes
  - Project Lightning Talks, Tue 12th November

- Strimzi: Data Streaming on Kubernetes with Apache Kafka
  - Maintainer Track, Fri 15th November

STRIMZI

Join Us

https://strimzi.io/join-us/

STRIMZI

Thank you


Website: https://strimzi.io


GitHub: https://github.com/strimzi


Twitter: @strimziio


YouTube: https://youtube.com/c/Strimzi


LinkedIn: https://www.linkedin.com/company/strimzi

STRIMZI