Azure Service Bus

Martyn Coupland

Principal Technical Consultant

Agenda

- Compare messaging services
- High level architecture
- Optimizing performance
- Recovery and replication
- High availability
- Handling outages

Event Grid vs Event Hubs vs Service Bus

Event Grid

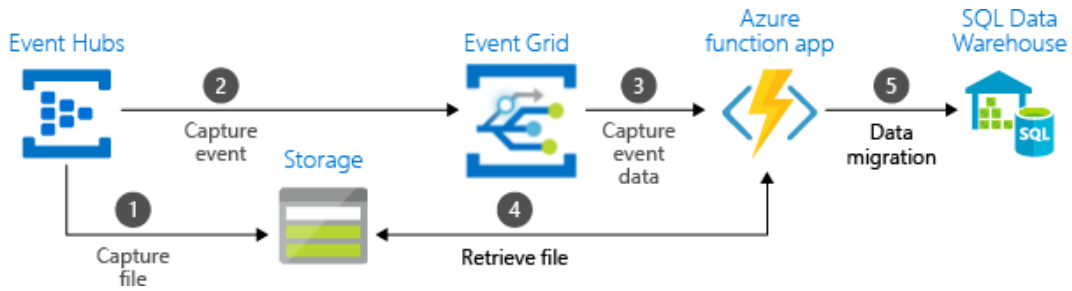
- Publish-subscribe
- Scalable
- Low cost
- Serverless
- ALO delivery

Service Bus

- State transition
- Broker based
- Asynchronous
- ALO delivery
- In-order delivery

Event Hubs

- Big Data pipeline
- Streaming
- Low latency
- Millions/second
- ALO delivery



High level architecture

- Common scenarios:
  - Messaging – Transfer business data, such as sales orders, inventory movements
  - Decouple applications – Improve reliability and scalability of applications (anti-corruption layers)
  - Topics and subscriptions – Enable 1:n relationships between publishers and subscribers
  - Message sessions – Workflows that require message ordering or deferral

High level architecture: Namespaces

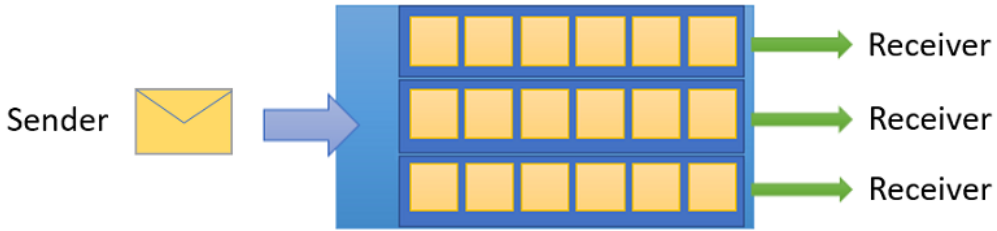
- Container for all components
- Multiple queues and topics
- Often containers for applications
- Maximum 200 per subscription (100 basic/standard, 100 premium)
- Max 1,000 TCP concurrent connections per namespace

High level architecture: Queues

- Messages sent to and received from queues
- Stored until receiving application is available
- Ordered and timestamped on arrival
- Once accepted, held in storage
- Delivered in pull mode, only delivering when requested

High level architecture: Topics

- Only used to send and receive messages
- Queue is for point-to-point communication
- Topics are publish/subscribe scenarios
- Subscriptions are independent
- Filters and rules define conditions that trigger optional actions



Topic with three Subscriptions with Messages

Advanced features

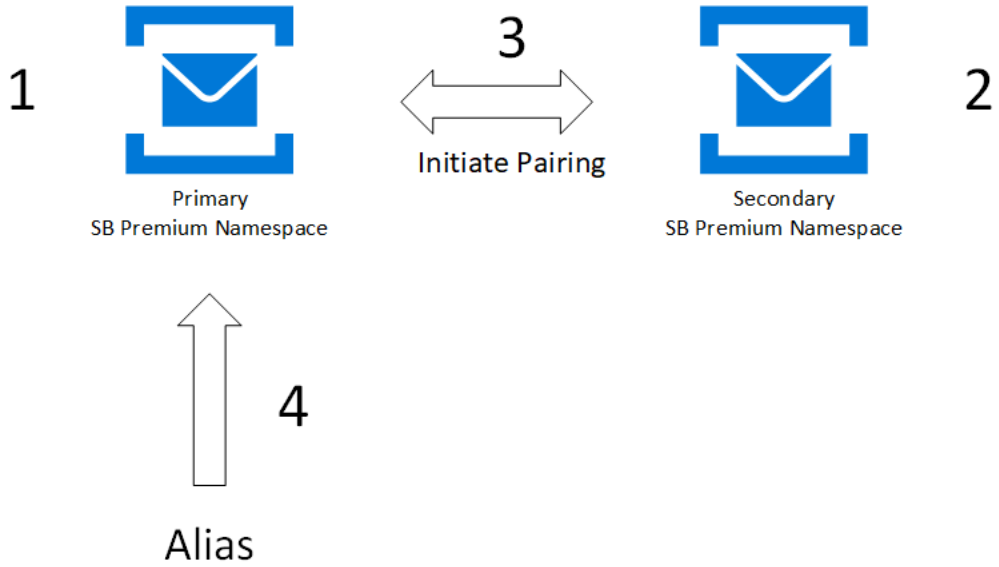
- Message sessions
- Auto-forwarding
- Dead-letter queue
- Scheduled delivery
- Message deferral
- Batching
- Transactions
- Filtering and actions
- And More!!

Optimizing performance

Goal: Maximize the throughput of a single queue. The number of senders and receivers is small

- To increase the overall send rate into the queue, use multiple message factories to create senders. For each sender, use asynchronous operations or multiple threads.
- To increase the overall receive rate from the queue, use multiple message factories to create receivers.
- Use asynchronous operations to take advantage of client-side batching.
- Set the batching interval to 50 ms to reduce the number of Service Bus client protocol transmissions. If multiple senders are used, increase the batching interval to 100 ms.
- Leave batched store access enabled. This access increases the overall rate at which messages can be written into the queue.
- Set the prefetch count to 20 times the maximum processing rates of all receivers of a factory. This count reduces the number of Service Bus client protocol transmissions.

Recovery and replication



Asynchronous and high availability

- Reliability in Service Bus
- Throttling
- Azure dependencies
- Single subsystem failures

Handling outages

- Protecting against outages in Standard namespaces
- Protecting against outages in Premium namespaces
- Protecting against data center outages

Useful links

- https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-quotas
- https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview
- https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-metrics-azure-monitor
- https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-authentication-and-authorization

- https://www.linkedin.com/in/martyncoup/
- Twitter: @mrcoups