Building Modern Microservices Architectures with Amazon ElastiCache

Zach Gardner, Sr. Solutions Architect

aws

Agenda

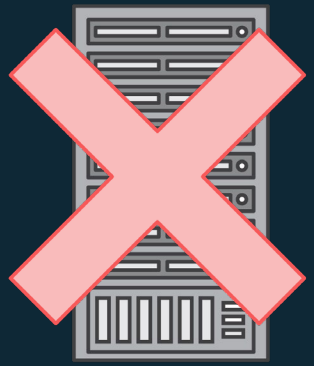
- Monolith to Microservices
- Architectural Components
- Purpose Built Databases
- Example Patterns:
    - Stateful Sessions
    - Caching – Write Through
    - Event Sourcing
    - Service to Service Communication
- Demo
- Additional Use Cases
- Getting Started

aws

Microservices

Monolithic applications - limitations

Hard to Scale



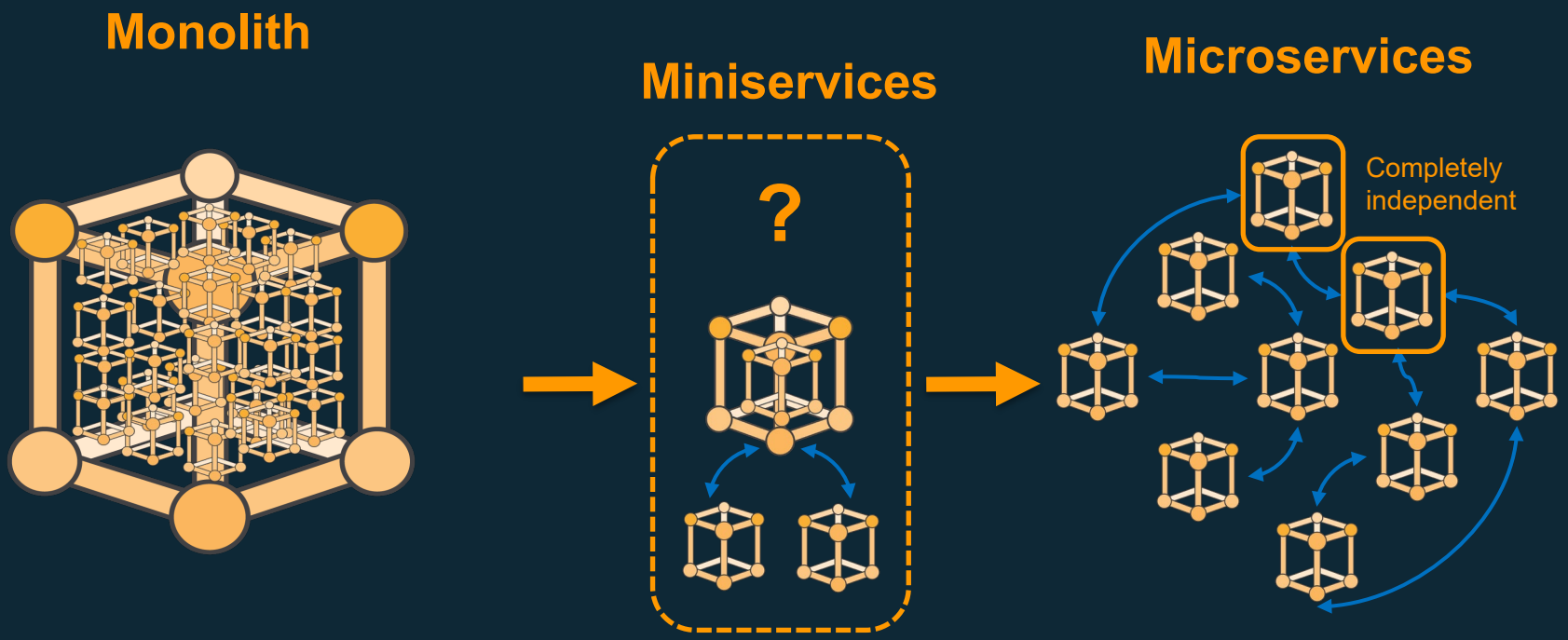
Can’t Handle Component Failures

Limited options

Slow Deployment Process

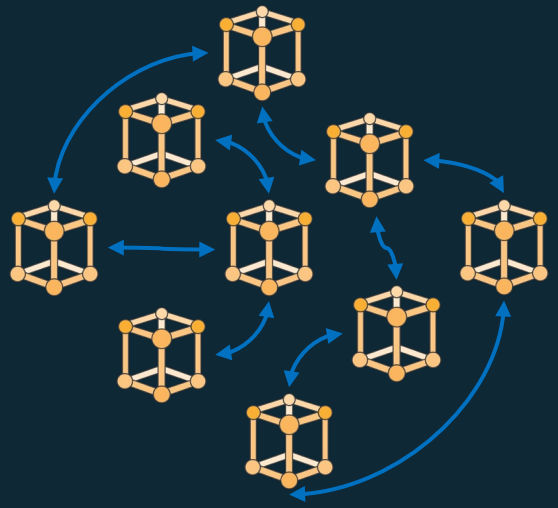
aws

Concepts and Definitions



Drivers to switch to microservices

- Time to Market
- Time to Repair
- Enabled Hyperscaling
- Technologically Independent



Amazon ElastiCache for Redis

Purpose-built databases

aws

Amazon ElastiCache for Redis

Fully managed

AWS manages all hardware and software setup, configuration, monitoring.

Extreme performance

In-memory data store and cache for sub-millisecond response times.

Scalable

Write and memory scaling with sharding.
Non-disruptive scaling.
Read scaling with replicas.

OSS compatible

Fully compatible with open source.

aws

What is Redis?

Initially released in 2009, it provides:

- In-memory data structures:
  Strings, Lists, Sets, Sorted Sets, Hash Tables, HyperLogLog, Geospatial, and Streams
- High-availability through replication
- Scalability through online sharding
- Persistence via snapshot / restore
- Multi-key atomic operations
- LUA scripting
- Open source

A high-speed, in-memory, non-relational data store.

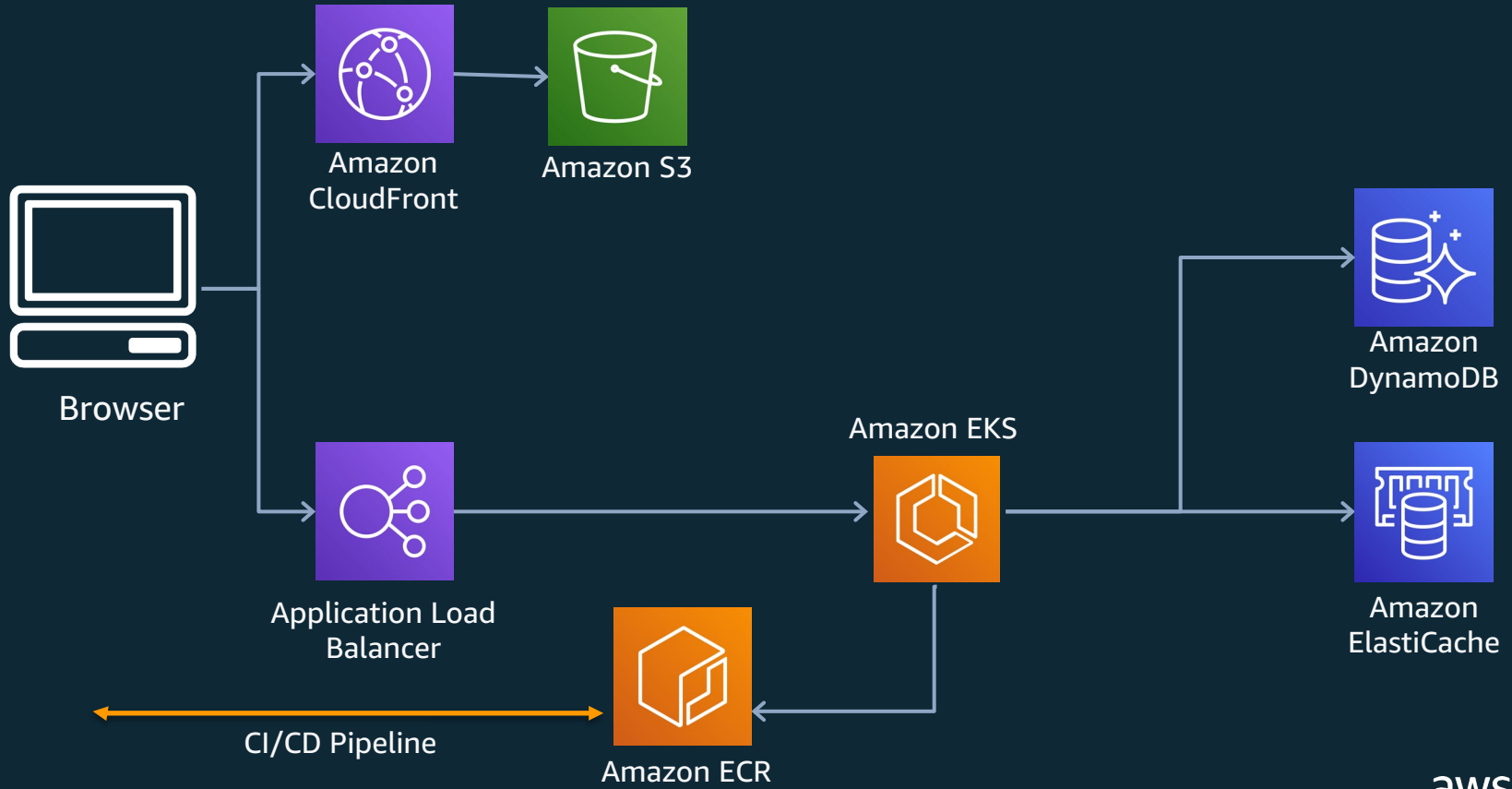
Customers love that Redis is easy to use.

aws

Microservices Patterns

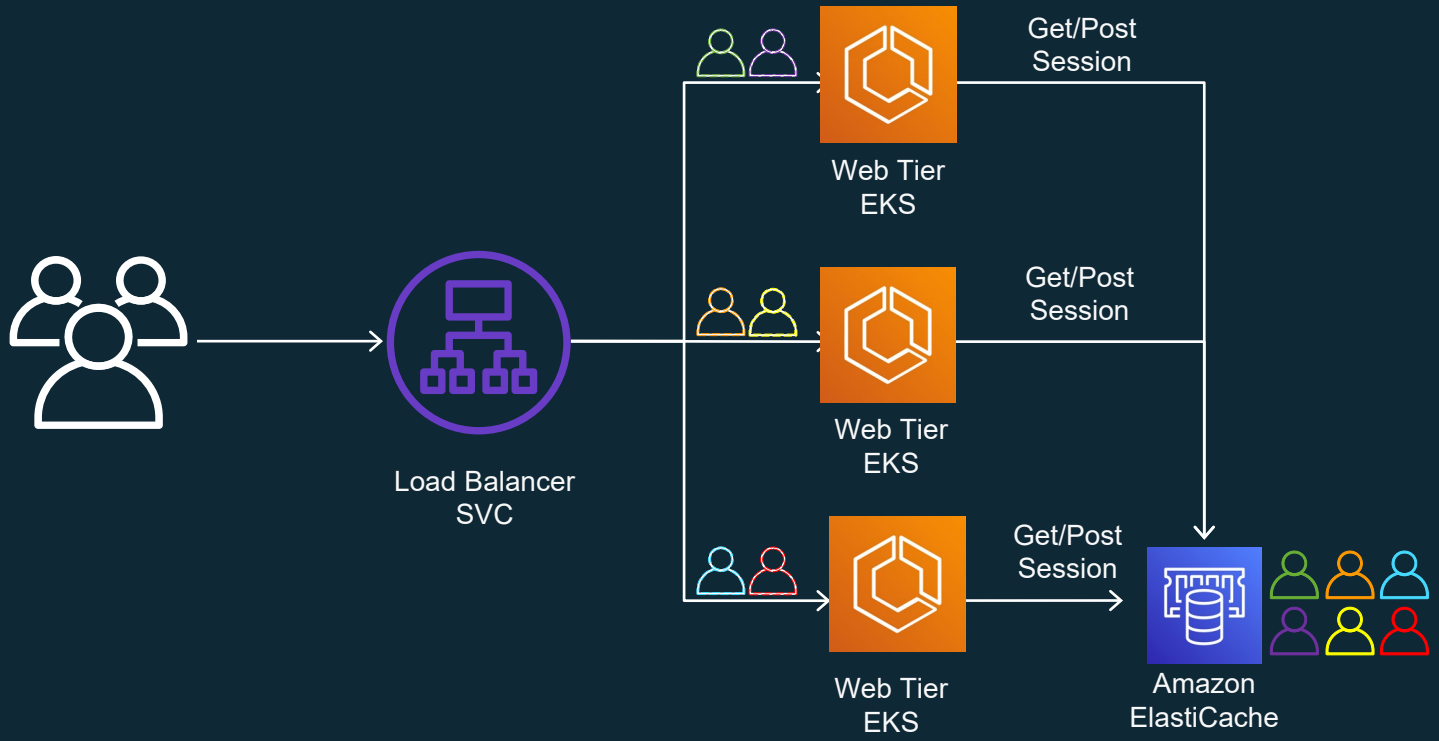
aws

Example Microservices Architecture

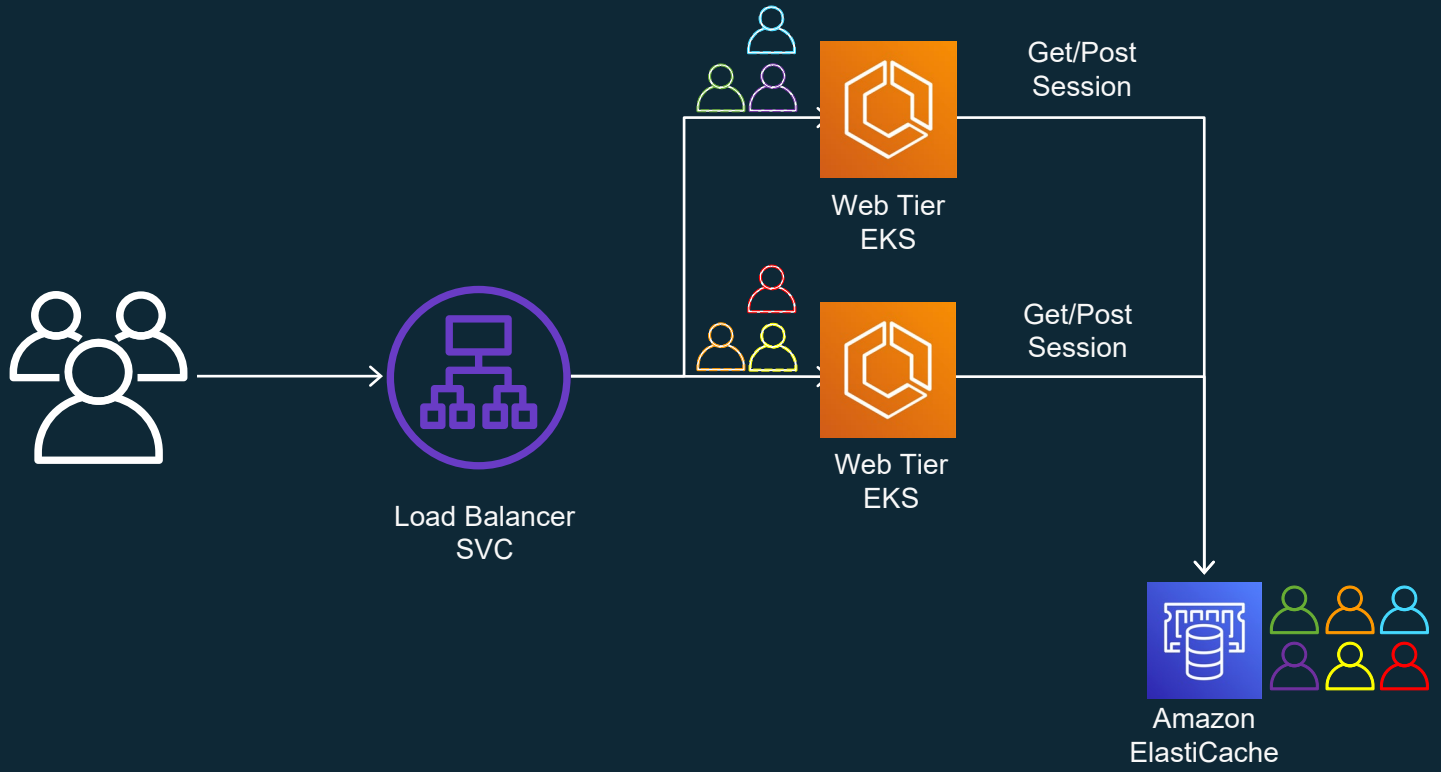


Stateful Services

Session Store – Stateful Services



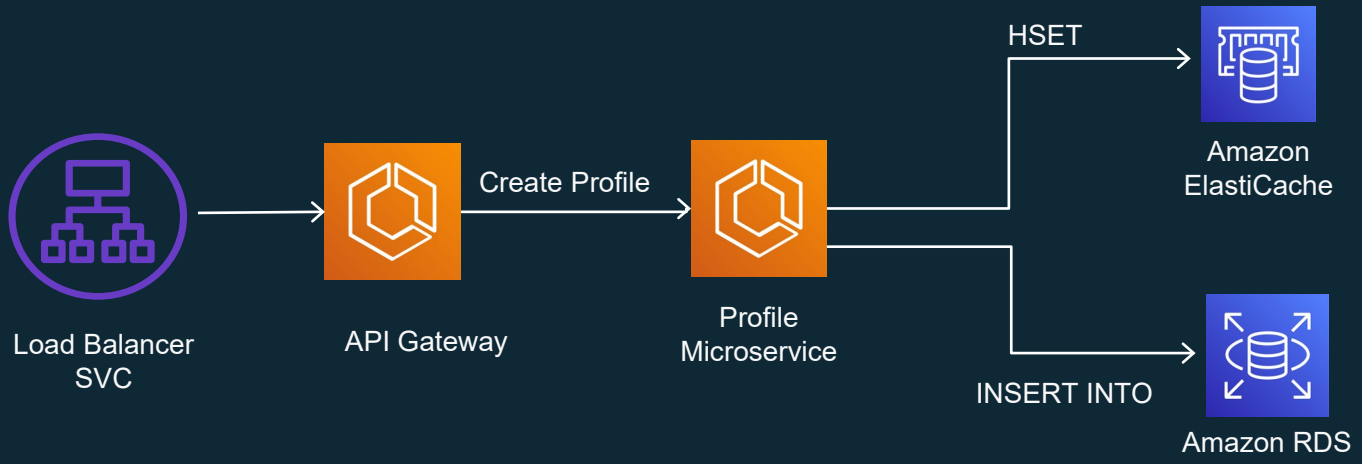
Session Store – Stateful Services



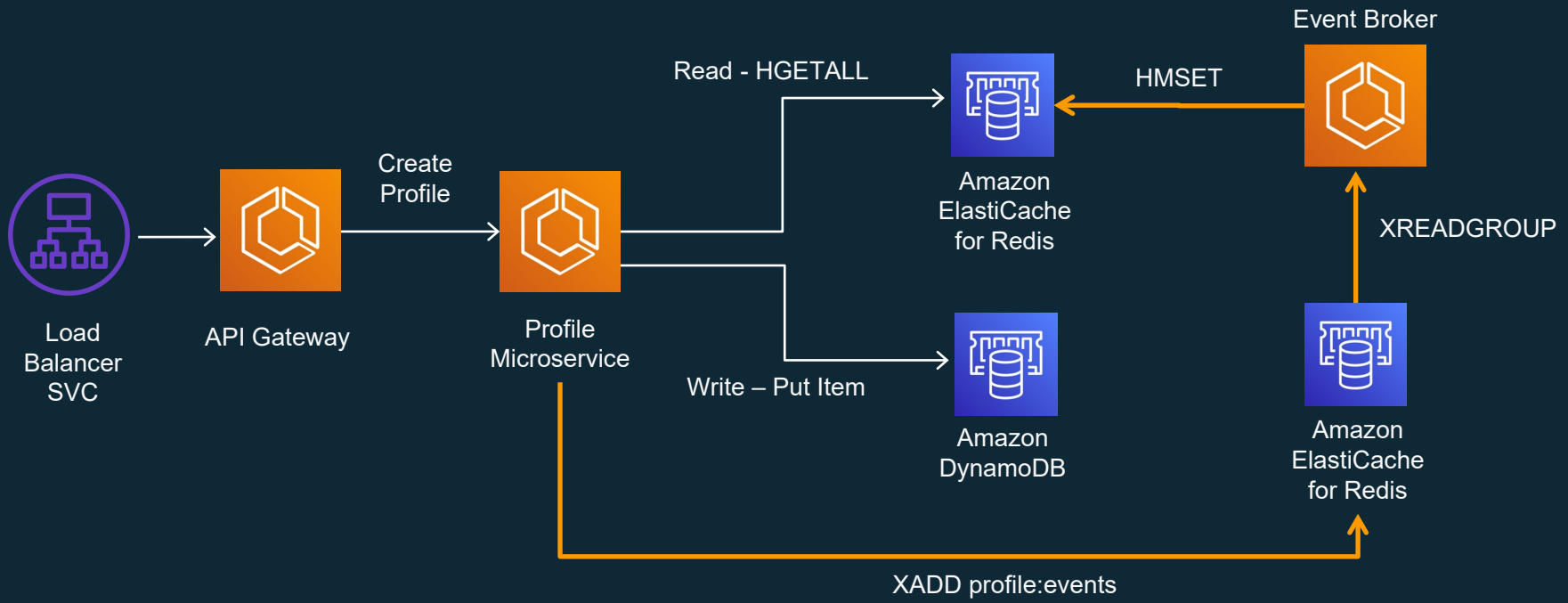
Caching and Datastores

aws

Caching – Write Through

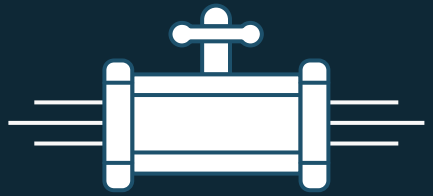


Caching – Event Sourcing



Service to Service Communication

Redis Streams



Redis Streams

Append-only log data structure

(Redis 5.0 and above)

Extreme performance

Collect large volumes of data arriving in high velocity

Communicate at Scale

Message brokers, message queues, website activity tracking, event streams

aws

Concepts

STREAMS DATA TYPE

Log of events in sequential order

Supports lookup by ID

IMPLEMENTED WITH RAX

A Radix library tree optimized for fast lookups and range queries

PRODUCERS

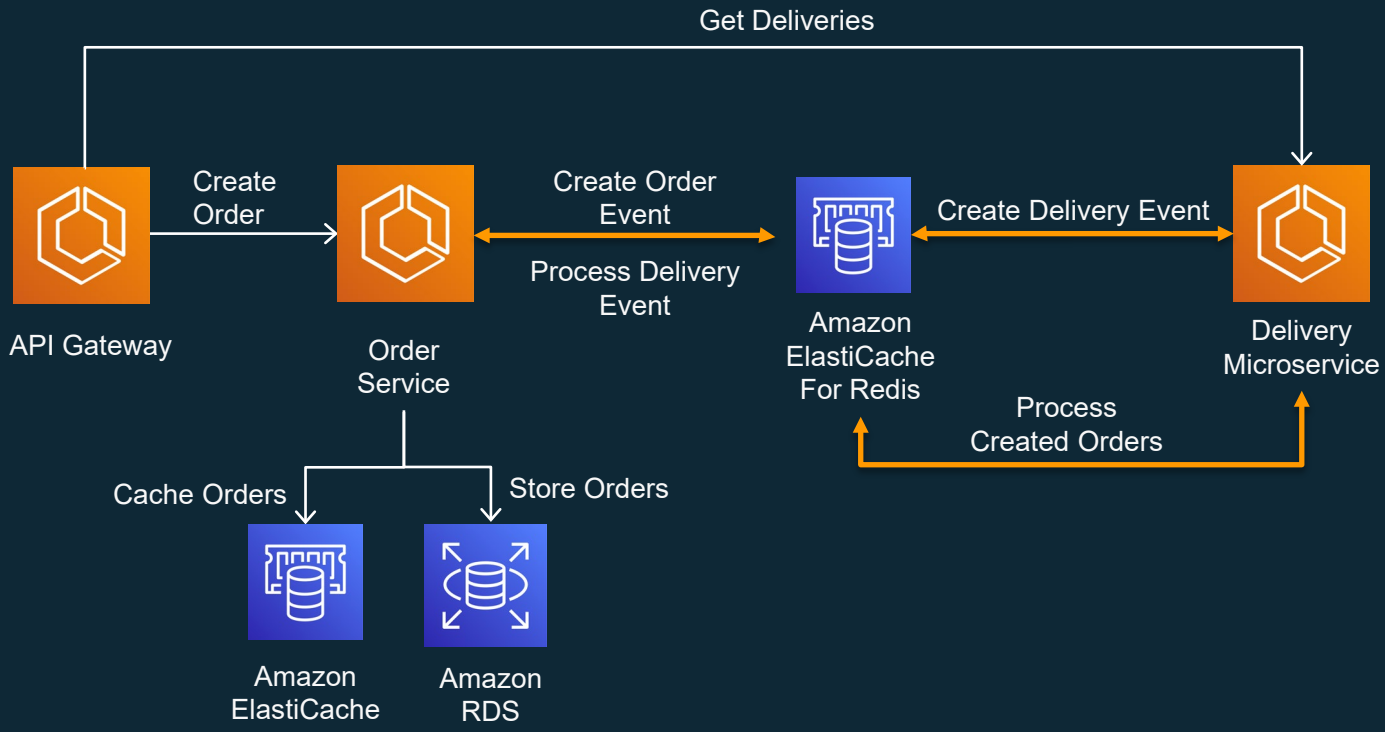
Push data records into the stream

CONSUMERS AND CONSUMER GROUPS

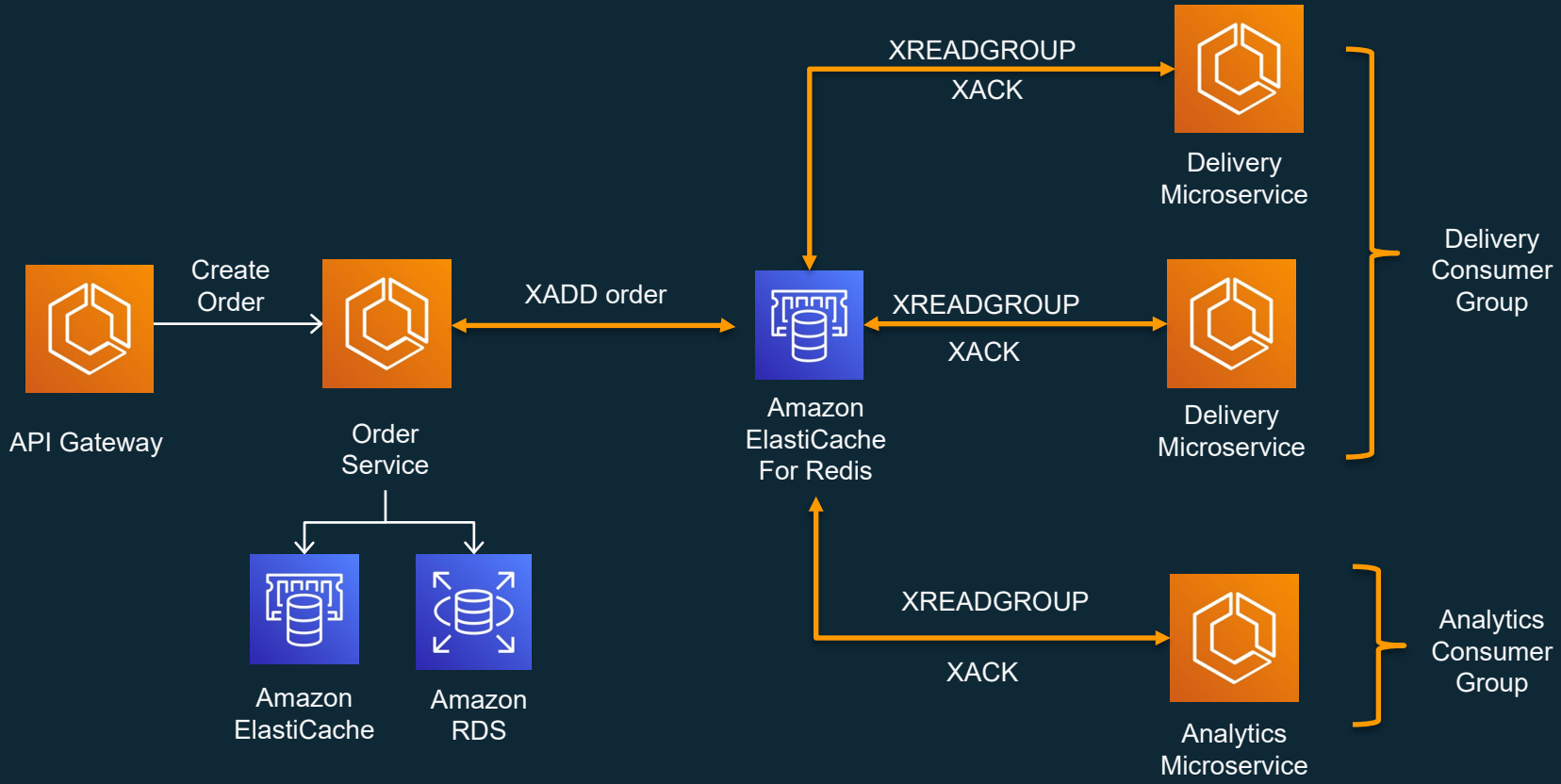
Mechanism to split a stream of messages among multiple consumers that are grouped together

aws

Use Case – Event Store Communication

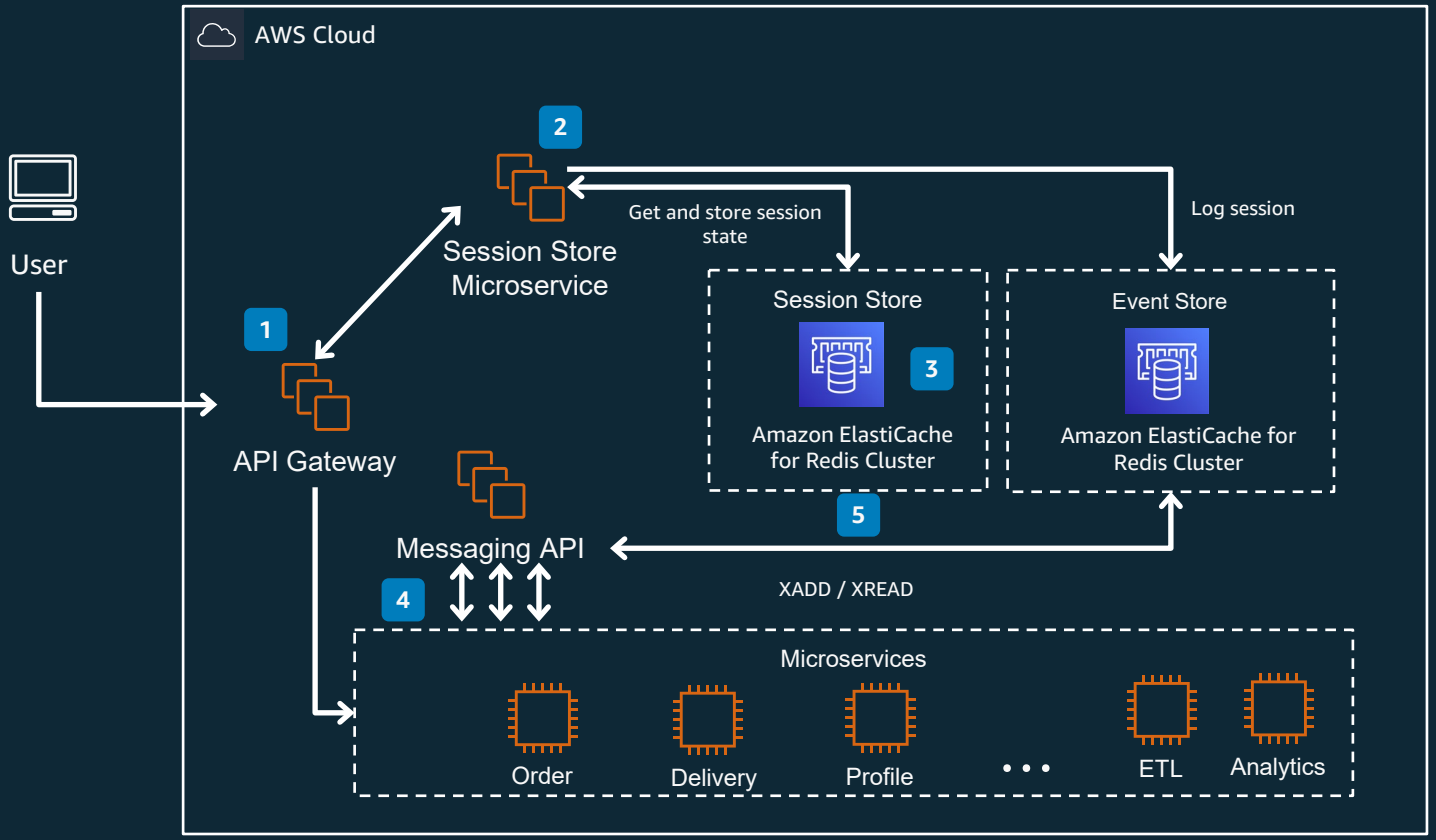


Use Case – Event Store Fan-out



Demo

Demo Architecture



Additional Use Cases

- API Rate Limiting
- Service Discovery
- Configuration Data Store
- Real-time Analytics
- Leaderboards
- Job Queues

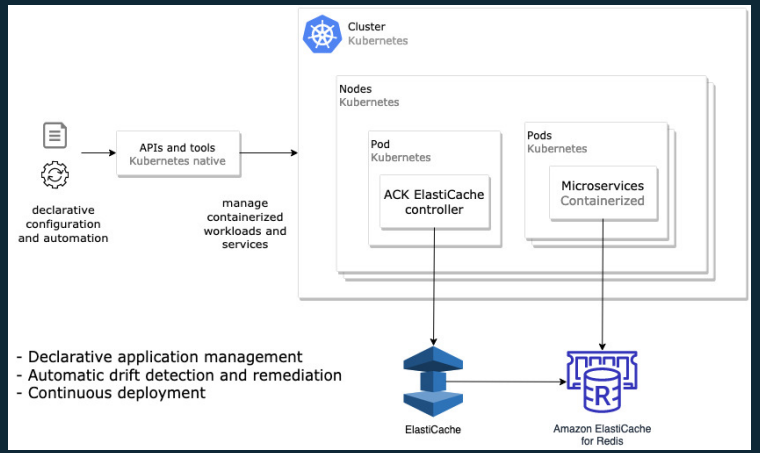
Developer Preview: AWS Controllers for Kubernetes (ACK) and Amazon ElastiCache for Redis

aws

AWS Controllers for Kubernetes (ACK) ElastiCache Controller

Kubernetes native experience to:

- Create and manage resources
- Perform Cluster scaling
- Automatic drift detection and remediation
- Provides Kubernetes access to Amazon ElastiCache for Redis resources using Custom Resource Definitions



Getting Started with Amazon ElastiCache for Redis

aws

Resources

FAQ	https://aws.amazon.com/elasticache/faqs/
Blog	https://aws.amazon.com/blogs/database/tag/elasticache/
Forum	https://forums.aws.amazon.com/forum.jspa?forumID=127
Getting started	https://aws.amazon.com/elasticache/redis/resources/
Learning Path	https://pages.awscloud.com/GLB-WBNR-AWS-OTT-2021_LP_0003-DAT_AmazonElastiCache.html

aws

Thank you!