



AWS Black Belt Online Seminar

Amazon MemoryDB for Redis

耐久性のあるRedis互換インメモリデータベースで
超高速なパフォーマンスを実現をする

堤 勇人

Senior Database Specialist Solutions Architect

2023/12

自己紹介

名前：堤 勇人 (Tsutsumi, Hayato)

所属：アマゾンウェブサービスジャパン合同会社
データ事業本部
サービススペシャリストソリューションアーキテクト本部
シニアデータベーススペシャリストソリューションアーキテクト

好きなAWSサービス：
Amazon MemoryDB for Redis,
Amazon ElastiCache, Amazon DynamoDB



本セミナーの対象者

- Amazon MemoryDB for Redisとは何かを知りたい方
- Amazon MemoryDB for Redisをご利用予定の方
- Amazon ElastiCacheやRedisをご利用中で、より耐久性を高めたい方

アジェンダ

1. Why Amazon MemoryDB for Redis?
2. Overview of Amazon MemoryDB for Redis
3. Amazon MemoryDB for Redis Technical deep dive
4. Amazon MemoryDB for Redis の始め方
5. まとめ

Why Amazon MemoryDB for Redis?



最新のリアルタイムアプリケーション

パフォーマンス、拡張性、可用性



ユーザー	1M+
データ量	テラバイト-ペタバイト
局所性	グローバル
パフォーマンス	マイクロ秒遅延
リクエスト率	ミリオン/秒
アクセス	モバイル、IoT、デバイス
スケール	アップダウンアウトイン
経済	従量課金制
開発者アクセス	オープン API



Eコマース



メディア
ストリーミング



ソーシャル
メディア



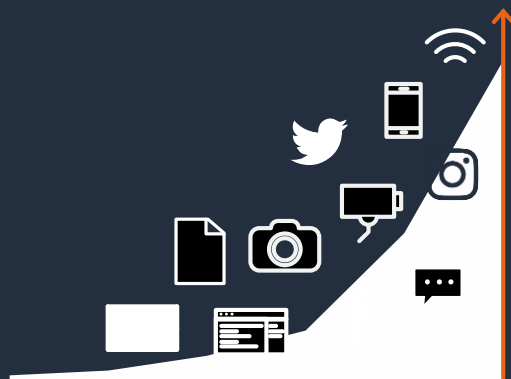
オンライン
ゲーム



シェアエコノ
ミー

データの生成方法と消費方法に影響する傾向

データの爆発的増加



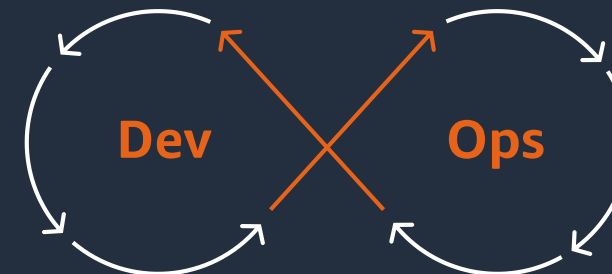
ネットワークに接続されたスマートデバイスにより、データは5年ごとに10倍に増加します

マイクロサービスはデータと分析の要件を変える



マイクロサービスアーキテクチャにより、汎用のデータベースの必要性が減り、リアルタイムの監視と分析の必要性が高まります。

DevOps がもたらす急速な変化



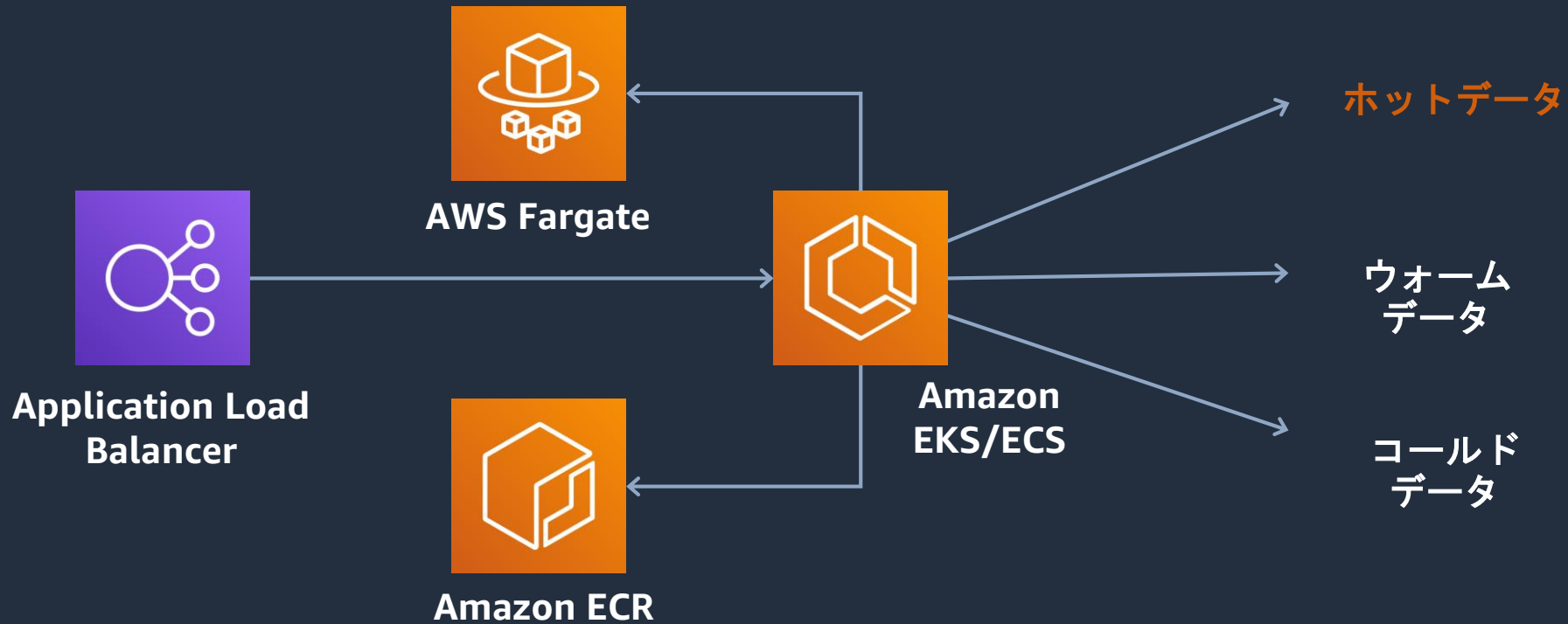
IT から DevOps への移行で加速

マイクロサービスアプリケーション

ホットデータの重要性

コンテナサービス

データサービス



インメモリデータベース： ホットデータのための高速で柔軟なデータベース

1. 高速

- 1つのユーザーアクションで多数のマイクロサービスが関与し、それぞれにわずかなレイテンシ猶予しかない
- 数百万人のユーザーが毎秒数百万件のリクエストを行うスループット
- 個々のサービスがより多くの計算を行い、より多くのデータにアクセスする

2. 柔軟

- アプリケーションの速度が速く、開発にはデータベースの俊敏性が必要
- 予期せぬビジネス要件をサポートするアプリケーションの進化
- アプリケーションモデルとデータベースモデル間のインピーダンスミスマッチを最小化

セルフマネージド型インメモリデータストアの課題



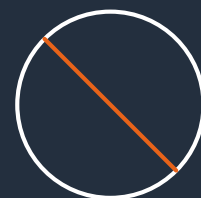
管理の難しさ

サーバのプロビジョニング、ソフトウェアパッチ適用、セットアップ、構成、バックアップを自分で管理



堅牢性の問題

多くのインメモリデータベースにはデータの耐久性がないため、システム停止やフェイルオーバー時にデータが失われる可能性があります。また、高速エラー検出と修復を実装する必要があります。



スケーラビリティ

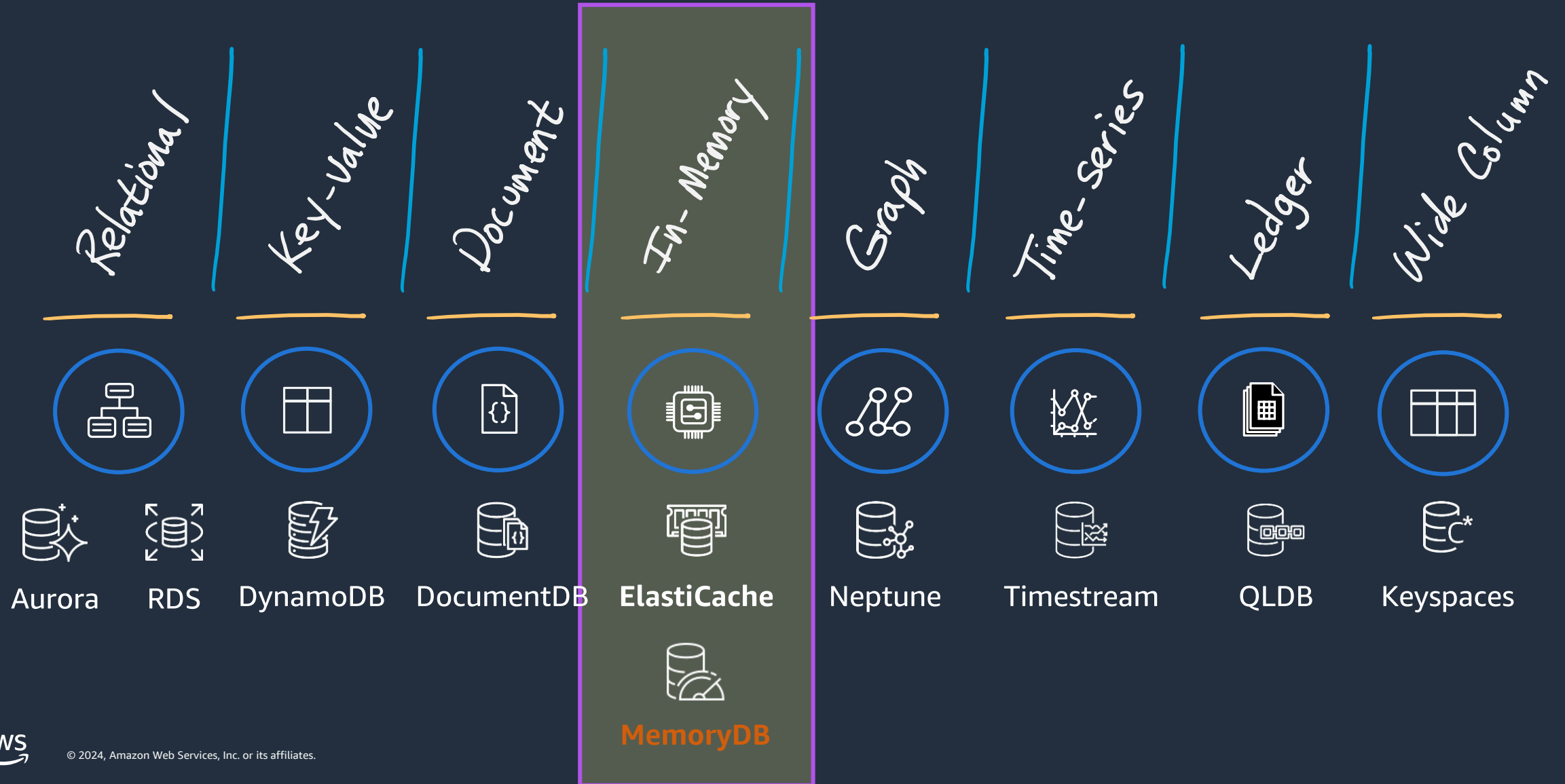
オンラインスケーリングはエラーが発生しやすく、スケーリング操作中にレプリケーションのパフォーマンスを監視する必要があり、問題が発生するとデータが失われる可能性があります。



高価

人材、プロセス、ハードウェア、ソフトウェアへの投資

Purpose-built Databases



AWS インメモリ耐久性選択

パーシステンス
なし

スナップショット
非同期レプリケー
ション

マルチ AZ トランザ
クションログ

揮発性

半耐久性

耐久性



ElastiCache
for memcached



ElastiCache
for Redis



MemoryDB
for Redis

MemoryDB for Redis vs ElastiCache

MemoryDB for Redis のユースケース

- 超高速なパフォーマンスを提供する堅牢なデータベースを必要とするアプリケーション
- Redis の API とデータ構造を使用するが、データ損失のリスクを回避するための**耐久性**を求めるアプリケーション
- キャッシュとデータベースを使用して低レイテンシーを実現し、**アーキテクチャを簡素化**してコストを削減したいアプリケーション

ElastiCache for Redis のユースケース

- 既存のプライマリデータベースでデータアクセスを高速化したいワークロードをキャッシュする場合
- Redis のデータ構造と API を使用して、プライマリデータベースまたはデータストアに格納されているデータにアクセスする場合



シンプルなアーキテクチャとアクセス



Overview of Amazon MemoryDB for Redis



Amazon MemoryDB for Redis

マルチ AZ の耐久性を備えた REDIS 互換のインメモリデータベース



超高速パフォーマンス

数百万 RPS でのマイクロ秒の読み取り
レイテンシーと 1 桁ミリ秒の書き込み
レイテンシー



Redis 互換

使いやすい Redis API と
豊富なデータ構造



耐久性と可用性

Multi-AZ トランザクションログと
99.99% SLA



フルマネージド

AWS が管理するハードウェアとソフト
ウェアのセットアップ、設定、監視、
スナップショット



スケーラビリティ

クラスターあたり最大500ノードと
最大105 TBのインメモリストレージ



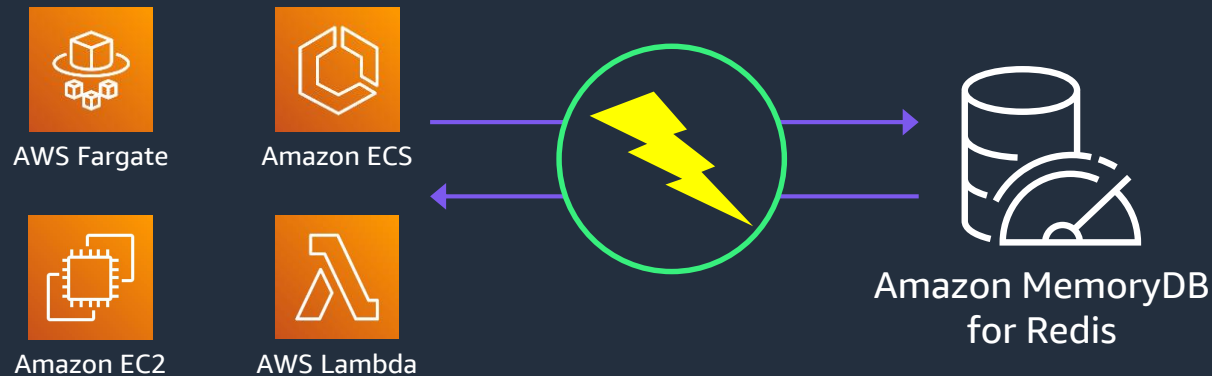
セキュリティと統制

IAM 認証
Access Control Lists (ACLs)
SOC, ISO, PCI-DSS, HIPAA, FedRAMP





Amazon MemoryDB: 超高速パフォーマンス



Applications

書き込み:マルチ AZ トランザクションログに保存

- 1桁ミリ秒という低いレイテンシーの書き込み
- シャードあたり最大 10 万回/秒の書き込み
- シャードあたり 100 MB のスループット/秒

読み取り:すべてのデータをメモリ上に展開

- マイクロ秒のレイテンシー読み取り
- ノードあたり最大 47万回/秒の読み取り
- ノードあたり 1.3 GB のスループット/秒

極限のスケール

- クラスタあたり 500 ノード
- 高可用性構成では 105 TB のデータ
- 数百万の RPS

Amazon MemoryDB for Redis のコア機能



Durability



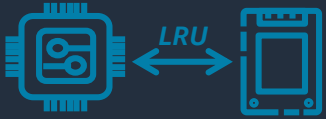
μ sec reads



Fully managed



Multi AZ with automated failover



Data tiering for optimal price/performance



Encryption in-transit and at-rest



Up to 99.99% SLA



Redis OSS compatibility



Scale horizontally and vertically



Secure & compliant, HIPAA, PCI-DSS, SOC



Automatic backups



Vector Search (preview)

MemoryDB integration with other AWS services



Amazon EKS



Amazon IAM



AWS CloudTrail



Amazon S3



Amazon CloudWatch



Amazon KMS



Amazon SNS



NEW

PREVIEW

Vector search for Amazon MemoryDB for Redis

インメモリのパフォーマンスでベクトルの保存、
取得、インデックス作成、検索



高いスループットと再現率を必要とするアプリケーション向けの AWS で最も速いベクトル検索エクスペリエンスの一つ



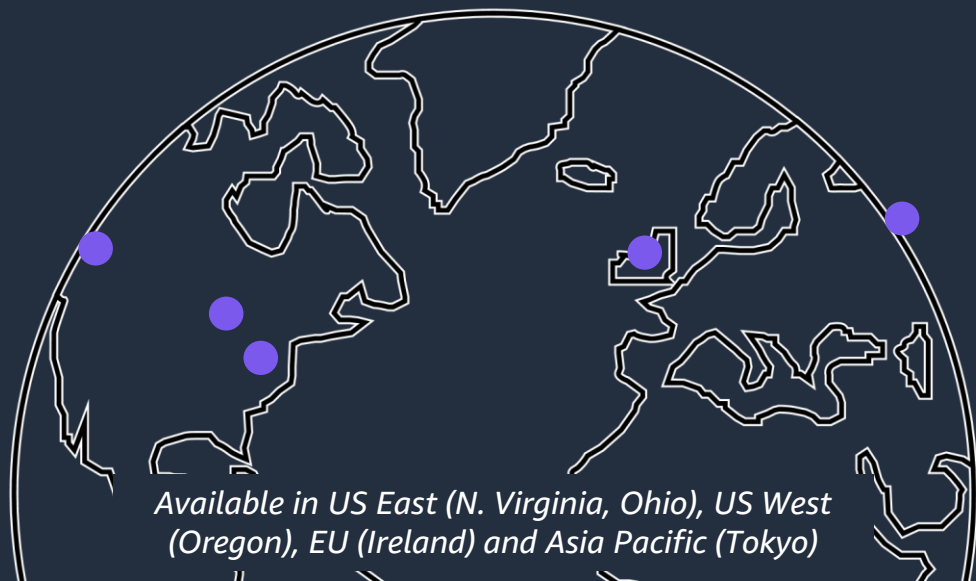
1 桁のミリ秒単位のクエリと更新の応答時間



1 秒あたり数万件のクエリ、99% 以上の再現率 (関連結果)



MemoryDB ベクトル検索 – preview



Available in US East (N. Virginia, Ohio), US West (Oregon), EU (Ireland) and Asia Pacific (Tokyo)

Node types: R7g, R6g, T4g

Available with MemoryDB for Redis v7.1



ミリ秒単位の検索と更新のレイテンシー



最大 419 GB のメモリと 32,768 次元



Redis JSON と Hash を使う
数百万の embedding



FLAT and HNSW
インデックス



ユークリッド距離、コサイン類似度、ドット積類似度検索



MemoryDB のユースケース



マイクロサービス

小規模で疎結合型のサービスを、超高速の読み取りと書き込みで強化



リアルタイムデータとソーシャルメディア分析

広告ターゲティング、パーソナライゼーション、IoT 向けのリアルタイムデータの取り込み、処理、分析



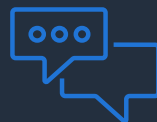
地理空間

IoT、ゲームなどのリアルタイムの地理空間データを大規模かつ迅速に管理



メディアストリーミング

ユーザープロファイルのメタデータや認証情報を含むライブストリーミングデータ



チャットとメッセージング

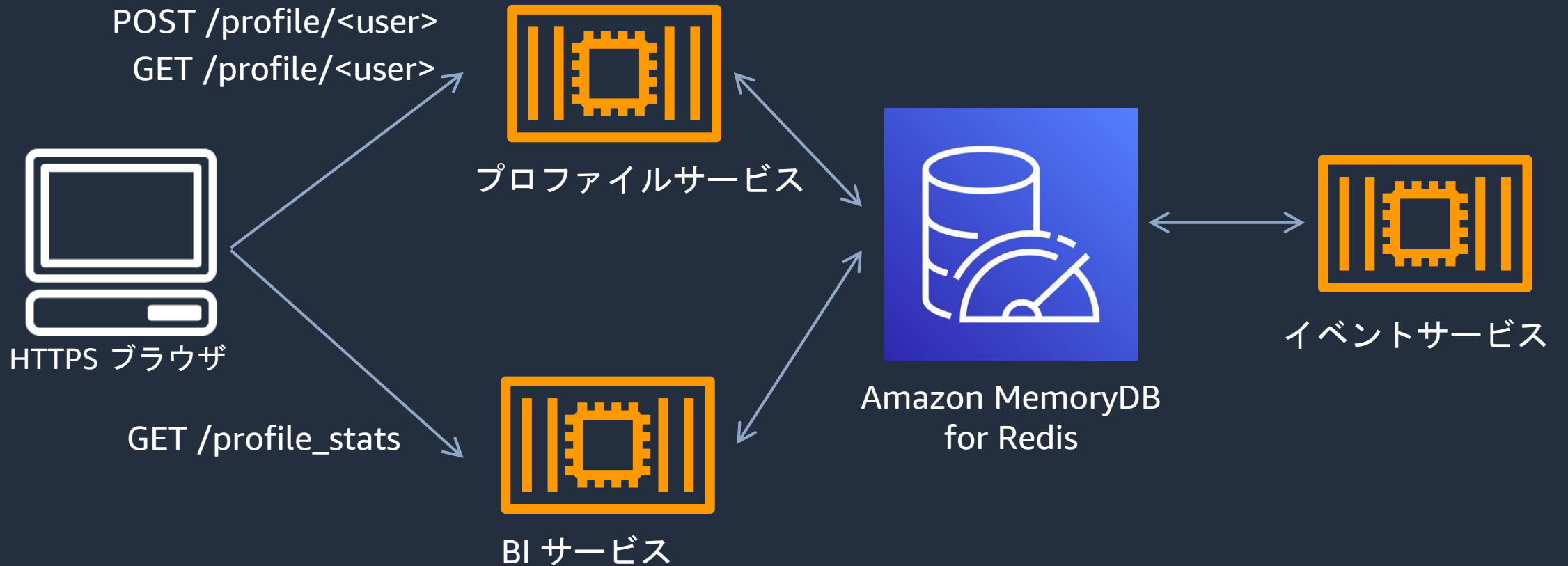
Redis Streams または PUS/SUB を使用して、高性能なチャットルームやリアルタイムのコメントをサポート



機械学習とジェネレーティブ AI

検索・拡張生成、異常検知、文書検索、リアルタイムのレコメンデーションにはベクトル検索 (プレビュー) を使用

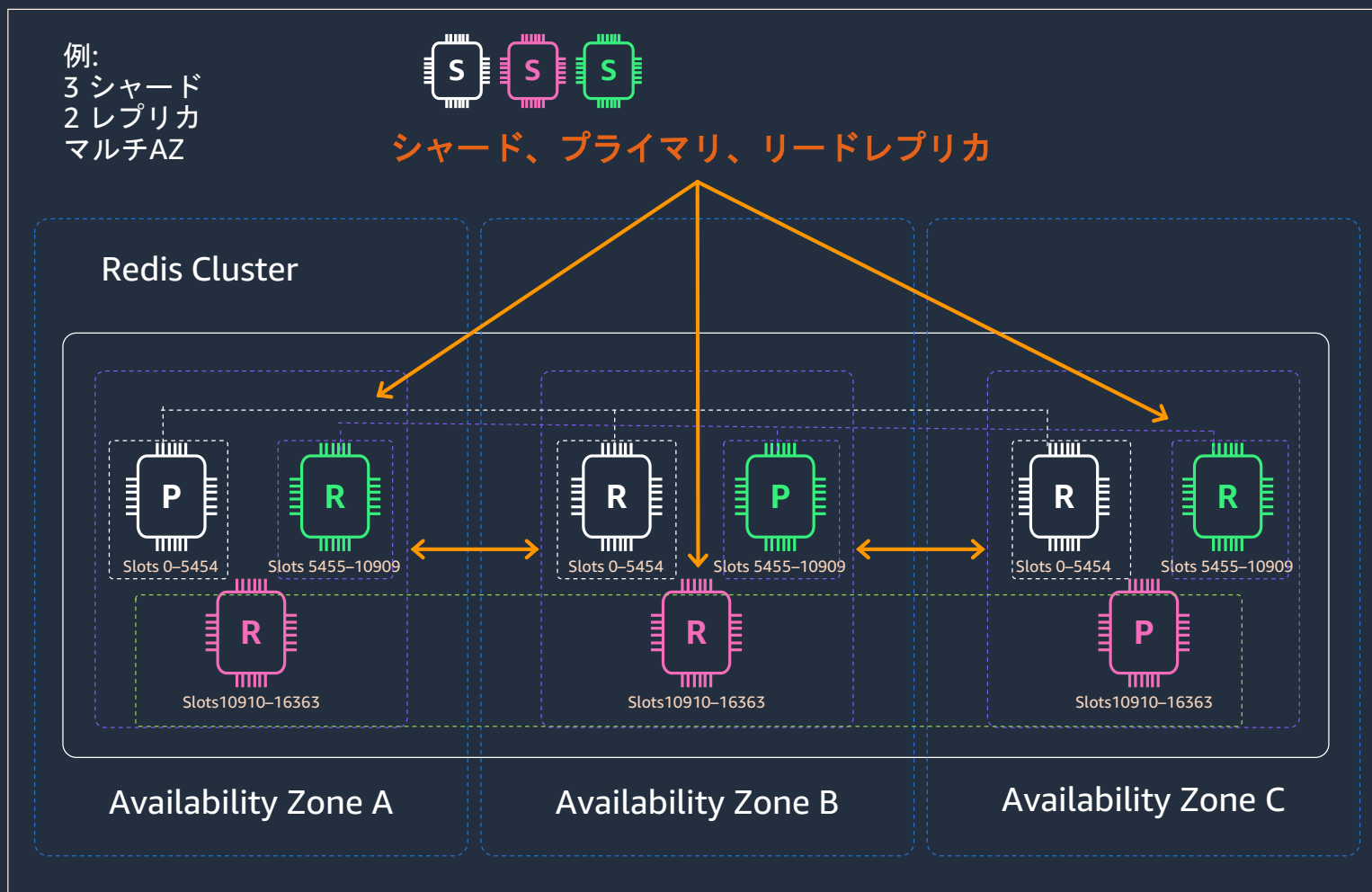
マイクロサービス間のメッセージング



Amazon MemoryDB for Redis Technical Deep Dive



高可用性とスケーラビリティ:マルチ AZ とレプリケーション



- シャードを追加して読み取り/書き込みをスケーリングし、インメモリ容量を増やす
- レプリカを追加して読み取りをスケーリングし、可用性を向上

高可用性とスケーラビリティ: クラスタ化アーキテクチャ

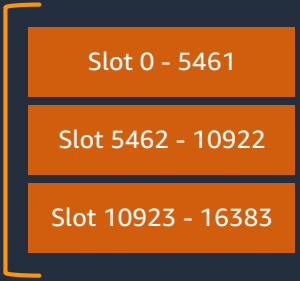
特徴

Redis クラスタ

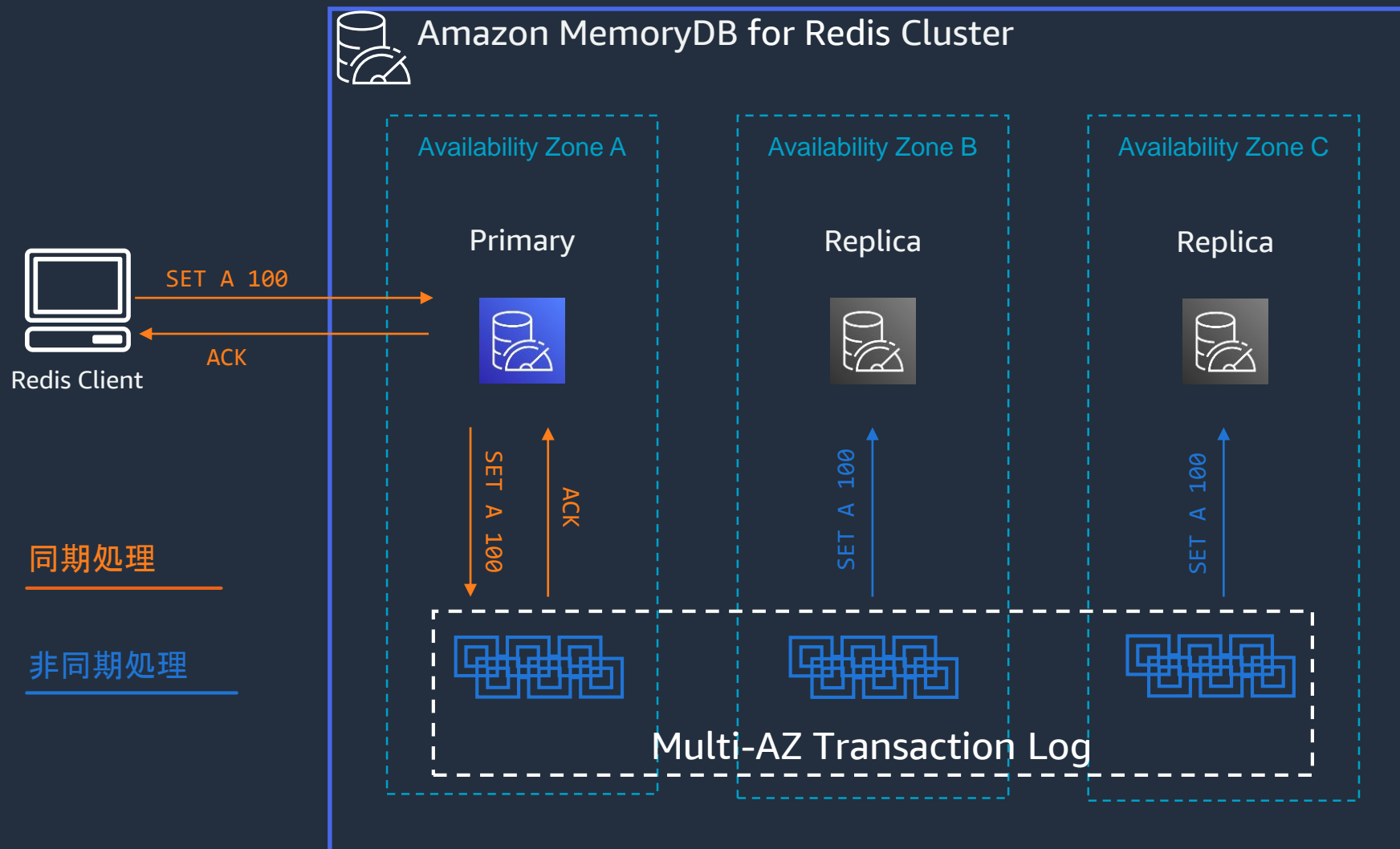
リカバリタイム	通常、計画フェイルオーバーは200ミリ秒未満、予定外のフェイルオーバーは10秒未満
フェイルオーバー影響	書き込みはフェイルオーバー中のシャードに関連した場合影響 読み取りはフェイルオーバー中も可能
ノードスケール	最大 500 ノード (83 ~ 500 シャード) シャードごとに1つのプライマリ、0 ~ 5 つのレプリカ
ストレージ	シャードごとに1つのレプリカで 100 TB 以上
オンラインスケールアップ	シャード、プライマリノード、リードレプリカ
マイグレーションパス	スナップショットのバックアップ/リストア、サードパーティツール
スケーラビリティとパフォーマンス	<ul style="list-style-type: none">水平スケールリングによりスループットを向上水平/垂直スケールリングをサポート
スケールアップ操作	クラスタサイズ変更 <ul style="list-style-type: none">シャードを追加/削除する水平スケールリング読み取りスケーラビリティによるレプリカの追加/削除



Configuration Endpoint



MemoryDB の耐久性

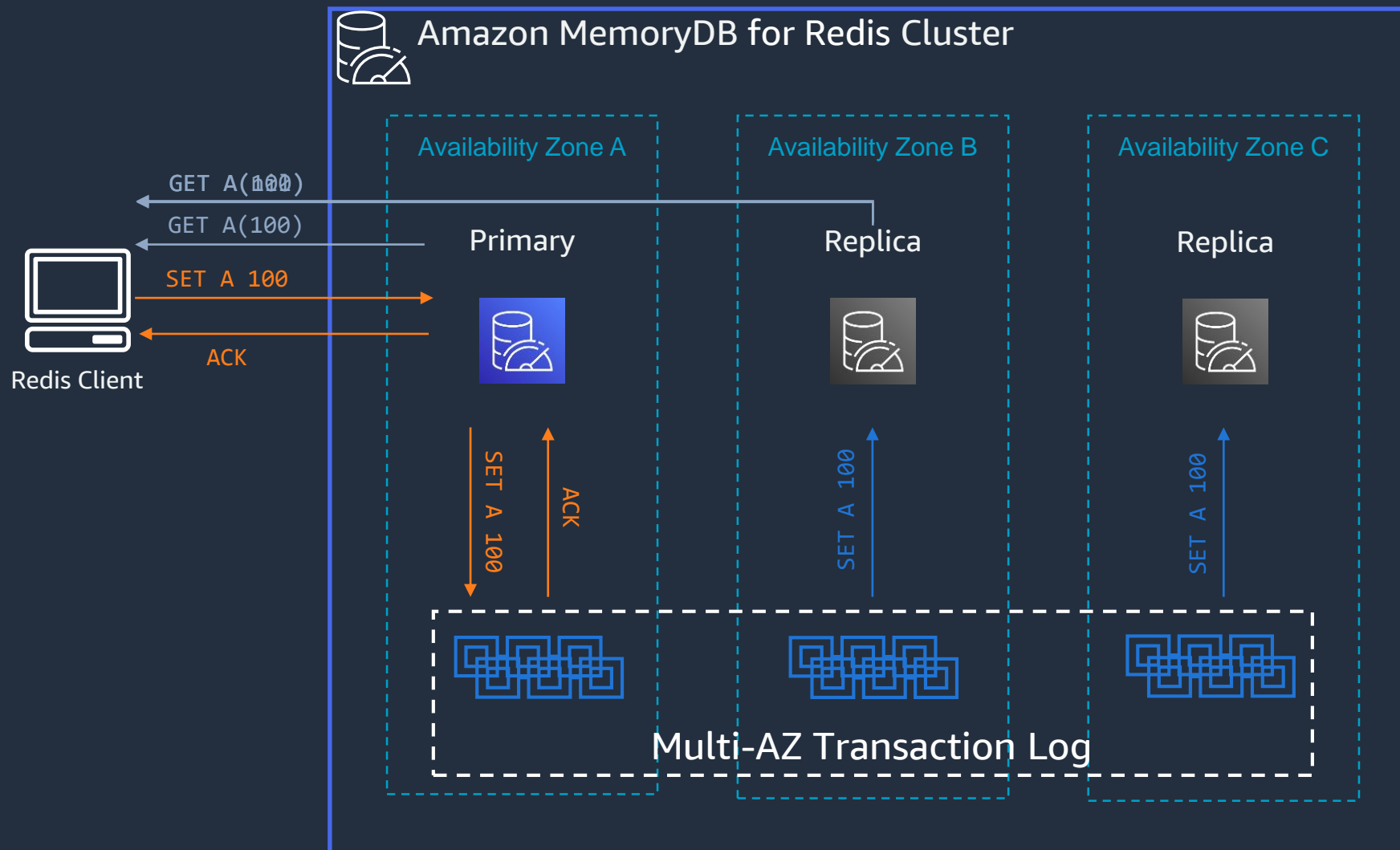


トランザクションを保存後にクライアントにACKを返す

トランザクションログにより、データレプリケートされ、レプリカで読み取られる

レプリカへの書き込み保証

MemoryDB の整合性

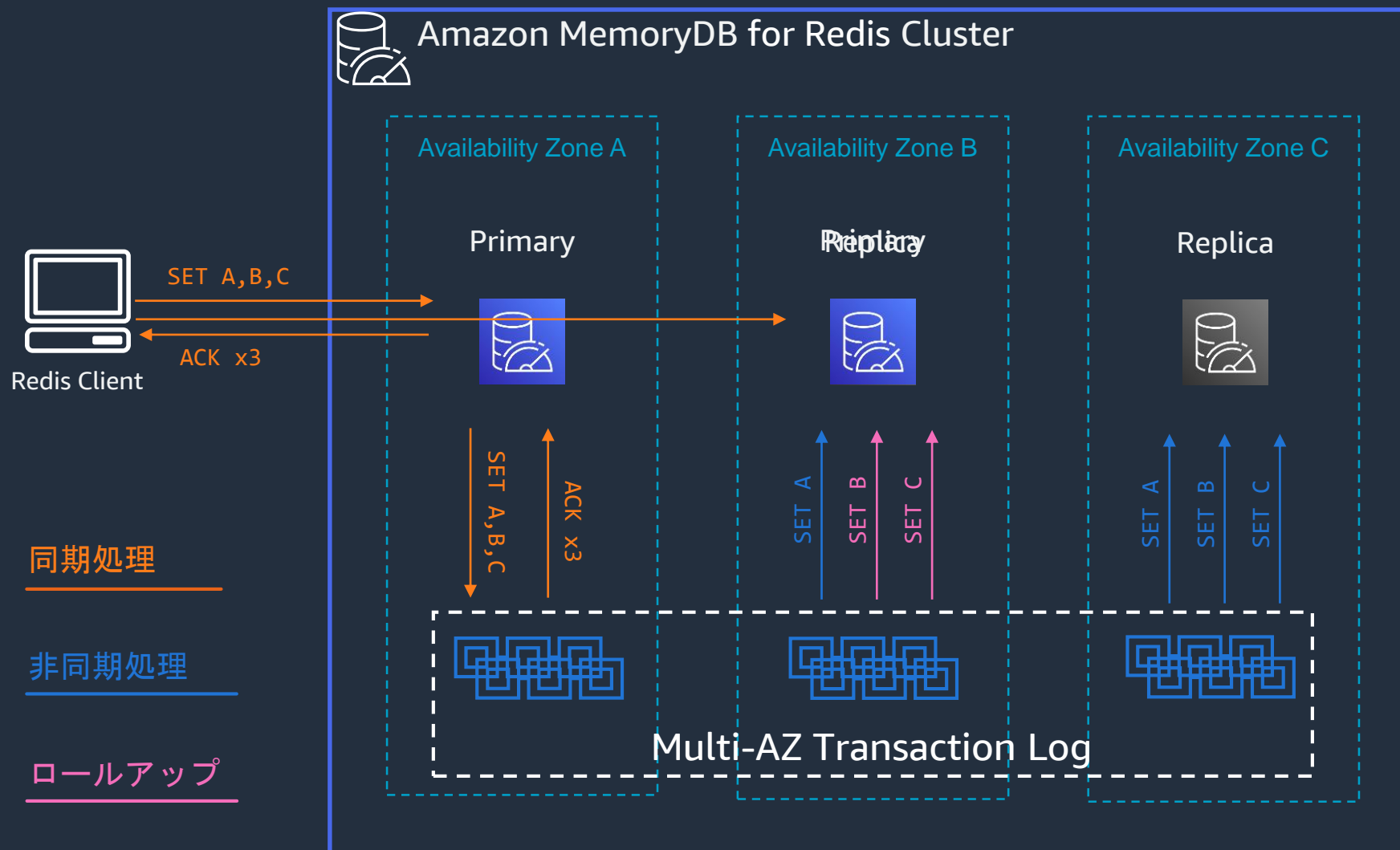


特記:
データは常にメモリ上に展開され、読み取りは超高速です。

プライマリからは強い整合性で読み取れます。

レプリカからは結果整合性で読み取れます

MemoryDB の耐久性 - フェイルオーバー

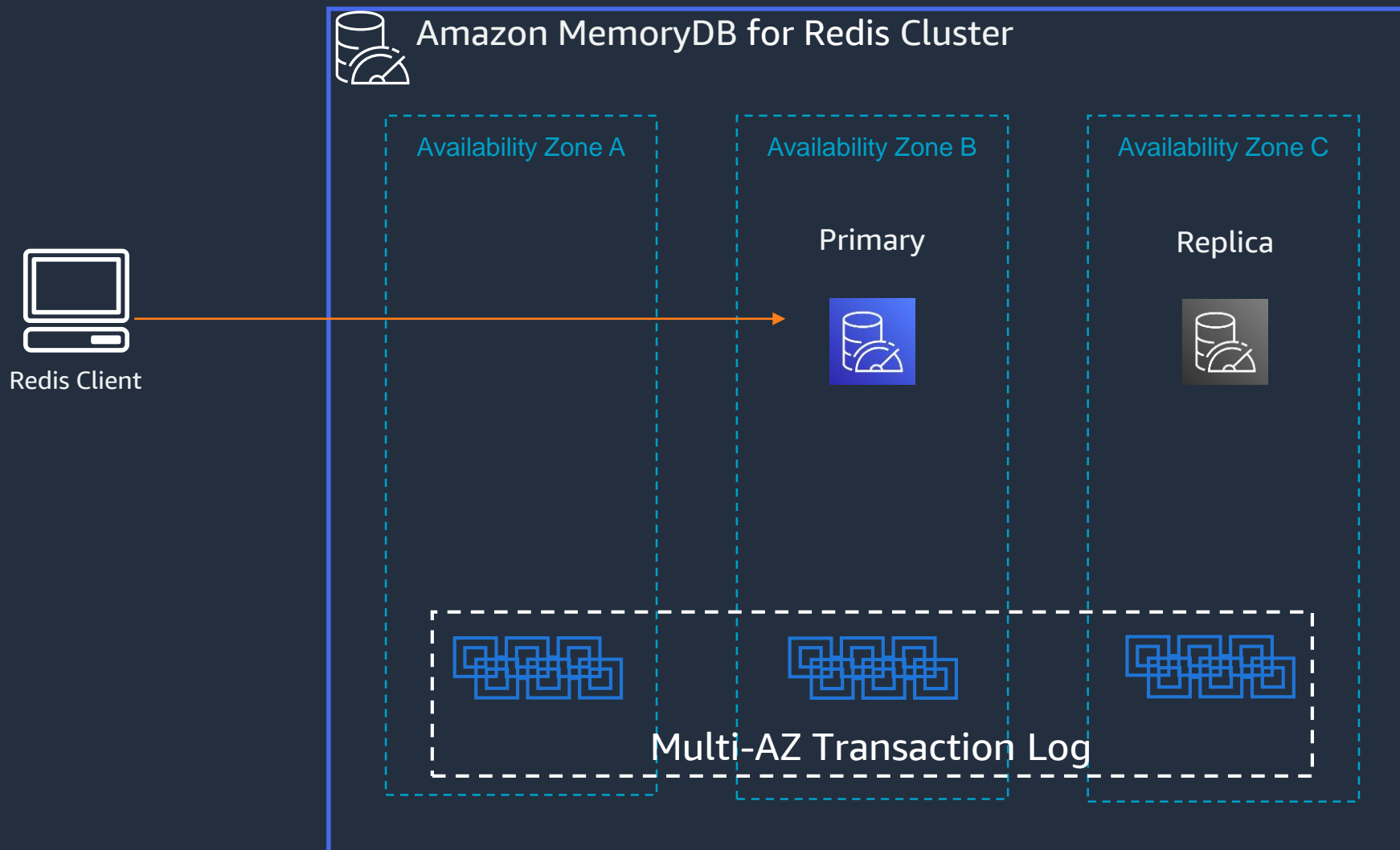


レプリカはログから全てのトランザクションを読み込むとプライマリに昇格します。

クライアントは全ての書き込みを新しく昇格したプライマリにリダイレクトされます。

レプリカは引き続きログを非同期的に読み込みます。

MemoryDB の耐久性 - フェイルオーバー

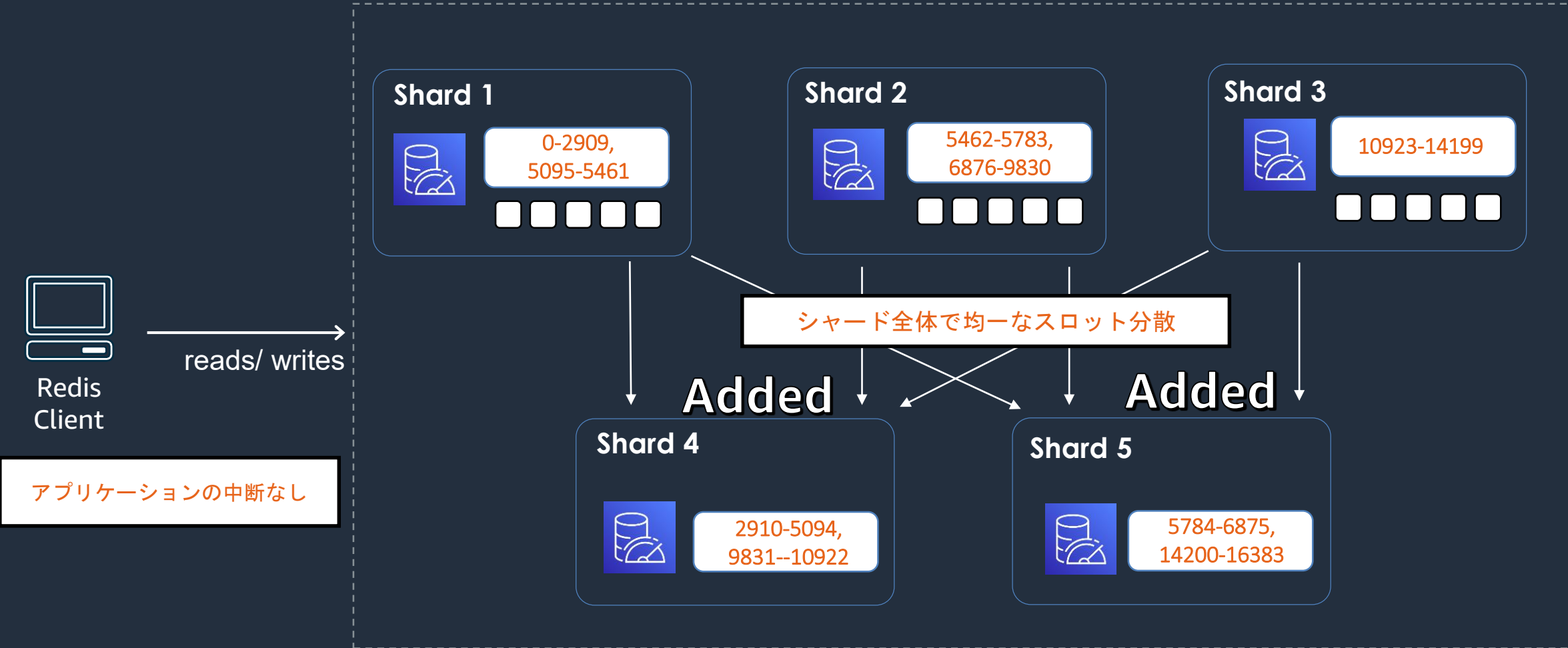


フェイルオーバー時の
データ損失はゼロ

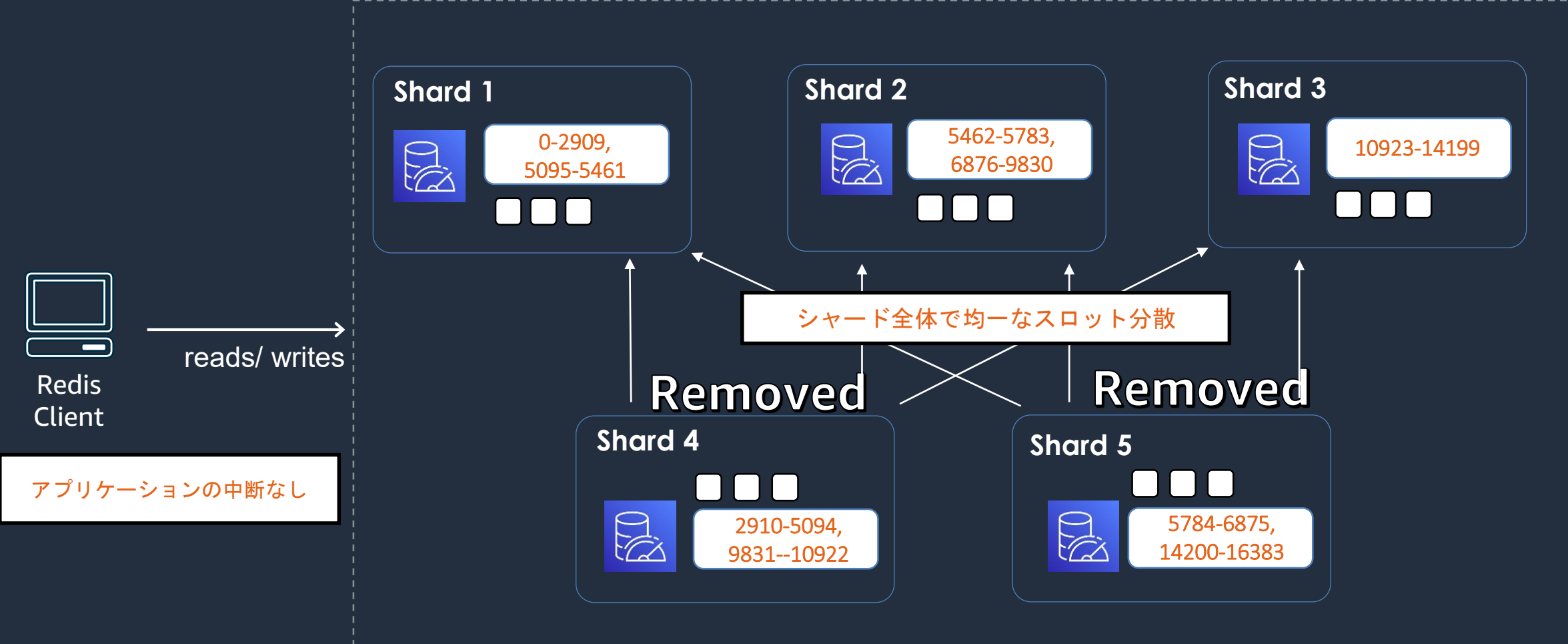
高可用性とスケーラビリティ:オンラインリシャードイング



高可用性とスケーラビリティ:オンラインスケールアウト



高可用性とスケーラビリティ:オンラインスケールイン



Redis Data Structures

Data Type	Description	Example Use Cases	Associated Commands
String	バイトシーケンス	キャッシュ	APPEND,GET,SET,INCR,DECR,GETSET...
List*	Stringのリスト	スタック	LSET,LLEN,LPUSH,LPOP,LTRIM,RPOP...
Set*	ユニークな順序無しのコレクション		SADD,SCARD,SDIFF,SUNION,SINTER,SMEMBERS...
Sorted Set*	ユニークな順序有りのコレクション	リーダーボード	ZADD,ZCARD,ZCOUNT,ZRANK,ZSCORE...
Hash*	Key-Valueペアのマップ		HGET,HGETALL,HKEYS,HVALS,HMSET,HMGET...
Streams	ログデータストラクチャ	メッセージキュー	XADD,XRANGE,XREAD,XACK,XCLAIM,XLEN...
Geospatial	緯度経度	地図、近傍検索	GEOADD,GEODIST,GEOPOS,GEORADIUS...
Bitmaps	Stringの特殊な用法		GETBIT,BITCOUNT,SETBIT,SETRANGE,GETRANGE...
HyperLogLogs	近似的カウント		PFADD,PFCOUNT,PFMERGE
JSON	JSONフォーマットの保存とネイティブアクセス		JSON.ARRAPPEND,JSON.ARRINDEX,JSON.CLEAR...

* These data types are capable of storing over 4 billion sub elements



MemoryDB: セキュリティ機能

暗号化

転送中の暗号化 (パフォーマンスに影響する可能性)

保存中の全データの暗号化

認証

Redis ACL を使用して、パスワードとアクセス文字列を持つユーザーを作成する
IAMを利用して認証することも可能

転送中の暗号化が有効になっている場合のみ使用可能

認可

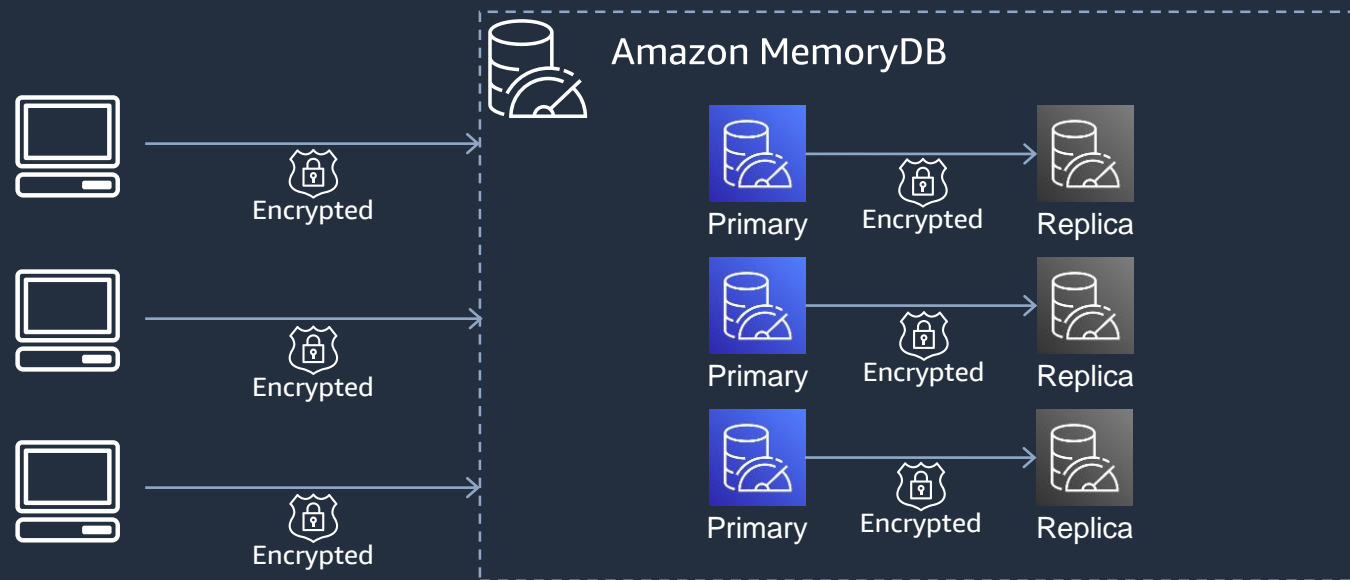
ACL を使用してコマンドとキーを制限する機能

Amazon VPC 統合

承認された IP、AWS リソースの VPC セキュリティグループ
ネットワーク分離のための VPC サブネットグループ



セキュリティ-転送中の暗号化



クライアントからクラスタへのトラフィックとノード間のトラフィックは暗号化される
クライアントは正しいサーバーに接続していることを認証できる

セキュリティ機能:認証

Redis v6



User: readapp

Account password



User: marketing




Account password



User: analytics

Account password

セキュリティ機能:認可

User	Authorized	Key space	Access string
 readapp	GET command	App keys	<u>on</u> <u>~app:*</u> <u>~@all</u> <u>+get</u>
 marketing	All commands	Marketing keys	on +@all ~mktg:* ~marketing:*
 analytics	All read-only commands	All keys	on ~* +@read

セキュリティ機能:オープンソース Redis セキュリティコマンド

ACL LOG コマンドを実行して詳細を確認する

```
> acl log
1) 1) "count"
   2) (integer) 1
   3) "reason"
   4) "command"
   5) "context"
   6) "toplevel"
   7) "object"
   8) "lrange"
   9) "username"
  10) "dbuser"
  11) "age-seconds"
  12) "562.7359999999999"
  13) "client-info"
  14) "id=15 addr=172.31.43.79:60584 fd=11 name= age=25
      idle=0 flags=N db=0 sub=0 psub=0 multi=-1 qbuf=42
      qbuf-free=32726 obl=0 oll=0 omem=0 events=r
      cmd=lrange user=dbuser"
```

セキュリティ機能:CloudWatch モニタリング

Amazon CloudWatch の重要なクラスターセキュリティメトリクス

AuthenticationFailures = Redis への認証の試行の失敗

KeyAuthorizationFailures = ユーザーが許可なくキーにアクセスしようとして失敗した

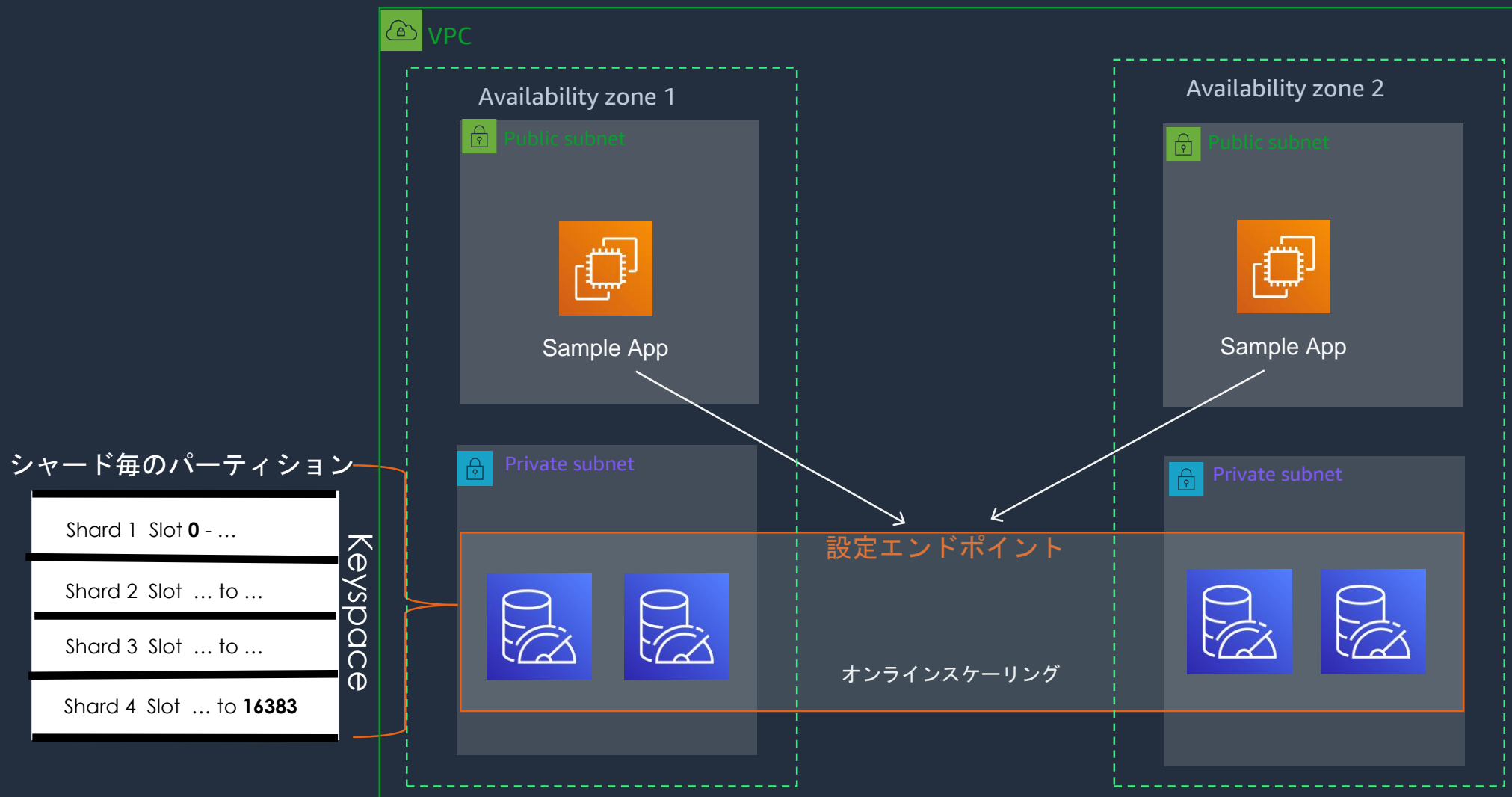
CommandAuthorizationFailures = ユーザーが許可なくコマンドを実行しようとして失敗した



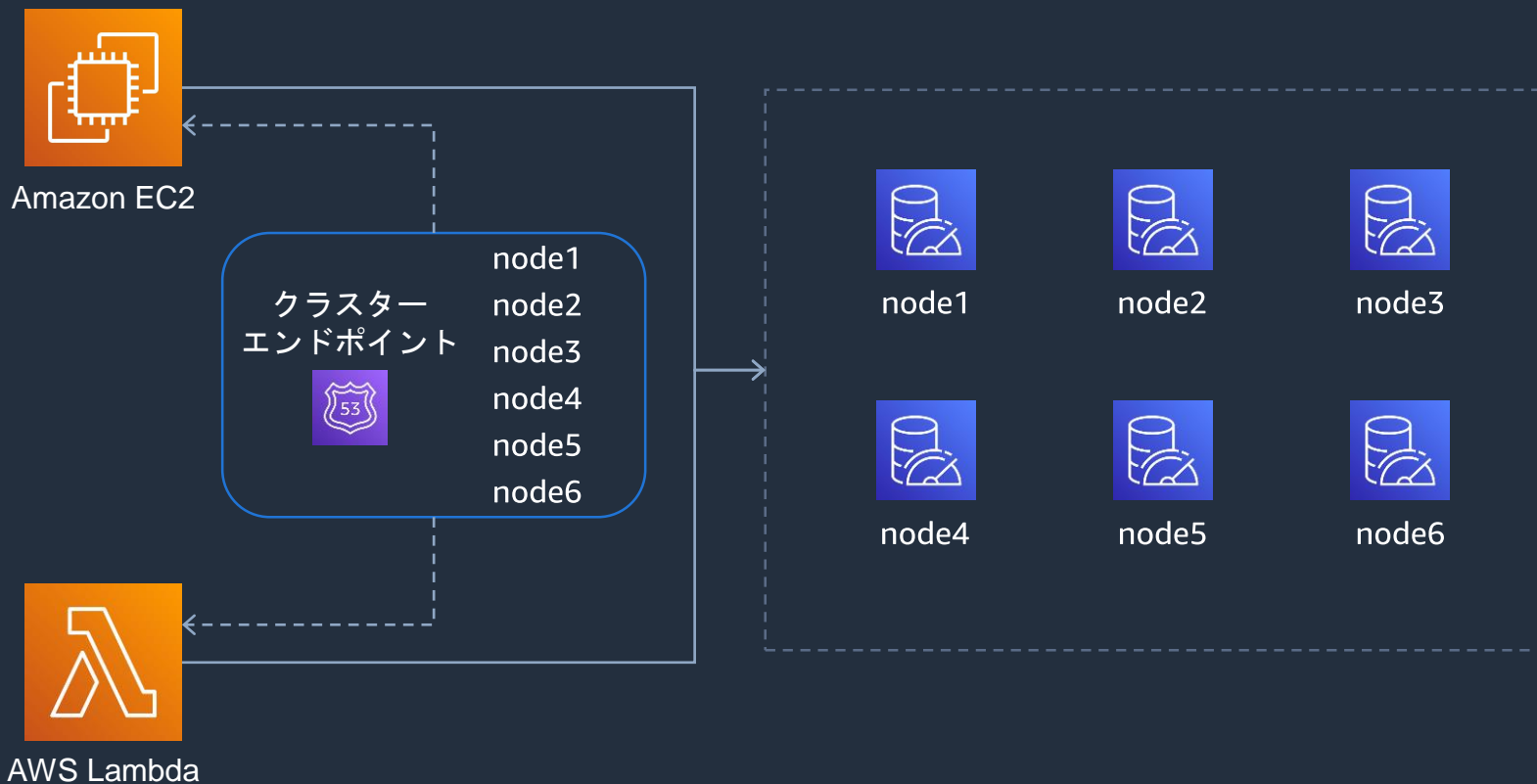
Amazon MemoryDB for Redis の始め方



MemoryDB に接続する



コネクション



オープンソース接続コード

```
startup_nodes = [  
  {"host": "node1", "port": "6379"},  
  {"host": "node2", "port": "6379"},  
  {"host": "node3", "port": "6379"},  
  {"host": "node4", "port": "6379"},  
  {"host": "node5", "port": "6379"},  
  {"host": "node6", "port": "6379"}]  
rc = RedisCluster(startup_nodes=startup_nodes)
```

クラスターの変更時にコードの更新が必要

MemoryDB 接続コード

```
startup_nodes = [  
  {"host": "cluster_endpoint", "port": "6379"}]  
rc = RedisCluster(startup_nodes=startup_nodes)
```

コード変更不要

MemoryDB に接続する

既存のオープンソース Redis クラスター対応クライアントの使用 — Python の例

```
import rediscluster
import certifi

MemoryDBEndpoint = "MEMORYDB_ENDPOINT"
startup_nodes = [{"host": MemoryDBEndpoint, "port": "6379"}]

cluster = rediscluster.RedisCluster(startup_nodes=startup_nodes, username='zach',
    password='REDISAUTH12345678910', skip_full_coverage_check=True, decode_responses=True,
    ssl=True, ssl_ca_certs=certifi.where())

cluster.set("hello", "memorydb")

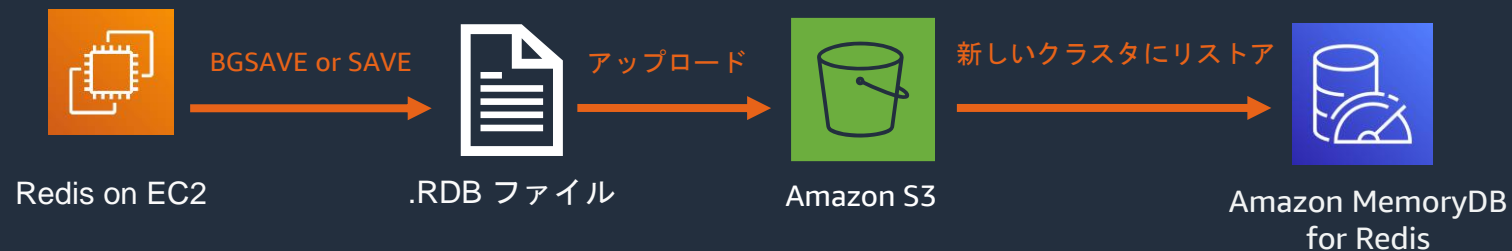
print(cluster.get("hello"))
```

クラスター対応の
オープンソースRedis
クライアントを使用
する

ACLによる認証

オープンソースの
Redis 6.2 API コマン
ドを使用する

自己管理型からの移行：バックアップ/リストア



推奨事項:計画されたメンテナンスウィンドウを活用する

1. BGSAVE または SAVE を使用して Redis バックアップを作成する
2. Amazon S3 バケットとフォルダを作成する
3. Amazon S3 にバックアップをアップロードする
4. MemoryDB に.RDB ファイルへの読み取りアクセス権を付与する
5. MemoryDB クラスターに.RDB ファイルデータをシードする

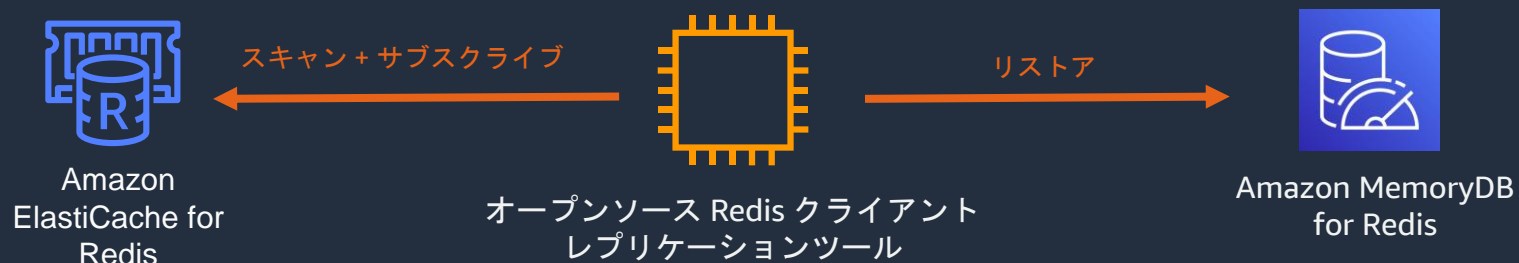
ElastiCache for Redis からの移行 : バックアップ/復元



推奨事項:計画されたメンテナンスウィンドウを活用する

1. Amazon S3 バケットを作成する
2. ElastiCache に Amazon S3 バケットへのアクセス権を付与する
3. コンソールか、コピースナップショットコマンドなどの API/CLI を使用して ElastiCache バックアップをエクスポートする
4. MemoryDB に.RDB ファイルへの読み取りアクセス権を付与する
5. MemoryDB クラスターに.RDB ファイルデータをシードする

論理レプリケーション - オープンソースクライアント



推奨事項: ダウンタイムを最小限に抑えるシナリオ

1. MemoryDB クラスターを作成
2. ソース Redis (ElastiCache またはオープンソース) と MemoryDB for Redis 間のクライアント接続を確保
3. ElastiCache for Redis からスキャンとサブスクライブを使用し、MemoryDB for Redis にキーを復元

まとめ



まとめ

- Amazon MemoryDBは**耐久性**を持つ**インメモリ**データベースです。
- データはメモリに展開され、**超高速**のパフォーマンスを実現します。
- プライマリデータベースとして利用できるよう設計され、これまでキャッシュ+プライマリデータベースと複数データベースが必要だった**アーキテクチャ**を**簡素化**できます。
- Redisと**互換性**があり、豊富なデータ型でリーダーボード、マイクロサービス、機械学習等、**様々なユースケース**を実現できます。

MemoryDB関連リソース

項目	URL
MemoryDB 製品ページ	https://aws.amazon.com/memorydb/
memoryDB よくある質問	https://aws.amazon.com/memorydb/faqs/
MemoryDB 価格設定	https://aws.amazon.com/memorydb/pricing/
MemoryDB の概要 (ビデオ)	https://youtu.be/uClbpz5LWPg
マニュアル	https://docs.aws.amazon.com/memorydb/latest/devguide/what-is-memorydb-for-redis.html
はじめに	https://docs.aws.amazon.com/memorydb/latest/devguide/getting-started.html

AWS Black Belt Online Seminar とは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン 合同会社が提供するオンラインセミナーシリーズです
- AWS の技術担当者が、AWS の各サービスやソリューションについてテーマごとに動画を公開します
- 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
 - <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
 - <https://www.youtube.com/playlist?list=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



ご感想は X (Twitter) へ！ハッシュタグは以下をご利用ください
#awsblackbelt

内容についての注意点

- 本資料では2023年12月時点のサービス内容および価格についてご説明しています。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (<https://aws.amazon.com/>) にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっております。日本居住者のお客様には別途消費税をご請求させていただきます
- 技術的な内容に関しましては、有料の [AWS サポート窓口](#)へお問い合わせください
- 料金面でのお問い合わせに関しましては、[カスタマーサポート窓口](#)へお問い合わせください (マネジメントコンソールへのログインが必要です)



Thank you!