Q 演習問題

1 **ある企業では、アプリケーションで使用するデータベースとして、NoSQLデータベースをAmazon EC2上に構築しています。Amazon EBSにはスループット最適化HDD（st1）を使用しています。最近、ストレージのI/O性能が原因の不具合が増えているため、改善したいと考えています。NoSQLデータベースのパフォーマンスを発揮するのに最も適したボリュームタイプは、以下のうちどれですか。**

A　EBSマグネティックボリューム

B　汎用SSDボリューム（gp2）

C　プロビジョンドIOPS SSDボリューム（io2）

D　コールドHDD（sc1）

2 **ある企業では、汎用SSDボリュームがアタッチされた単一のEC2インスタンス上でアプリケーションが実行されています。利用しているEC2インスタンスは、EBS最適化インスタンスを有効化できるインスタンスタイプです。この環境でAmazon EBSのパフォーマンスが低下した場合に、EBS最適化の有効化に加えてパフォーマンスを向上させる手段として、適切な措置は次のうちどれですか。なお、アプリケーション要件を満たすインスタンスタイプが利用されているものとします。**

A　汎用SSDボリュームをプロビジョンドIOPS SSDに変更する

B　インスタンスタイプを別のタイプに変更する

C　他リージョンで汎用SSDボリュームを利用する

D　汎用SSDボリュームをマグネティックボリュームに変更する

A 解答

1 C

最適なEBSボリュームタイプを選択する際の知識を問う問題です。
問題文の要件に「ストレージのI/O性能が原因の不具合が増えている」とあるので、I/O性能の高いボリュームタイプを選択する必要があります。これら選択肢のなかで高いI/O性能を期待できるボリュームタイプはCのため、**C**が正解になります。
AとDおよびBは、CほどのI/O性能を発揮することが難しいため不正解です。

2 A

EBS最適化インスタンスとAmazon EBSのパフォーマンスに関連する知識を問う問題です。
EBS最適化を有効にすると、EBS用に専用のネットワーク帯域幅が用意されるため、EBSのI/Oやトラフィック競合を最小限に抑えてパフォーマンスを向上させることができます。
問題文に「要件を満たすインスタンスタイプが利用されている」と記載されていることから、Bのインスタンスタイプの変更は不要のため不正解です。
EBSのパフォーマンスを向上させるには、より性能の高いボリュームタイプに変更する必要があるため、**A**が正解です。CおよびDは、汎用SSDボリュームよりもパフォーマンスを向上させられないため不正解になります。

4-5 データベースサービスにおけるパフォーマンス

AWSでは複数のデータベースサービスが提供されています。ユースケースに適したデータベースサービスを利用することで、システムのパフォーマンスを向上させ、最適化することができます。
本節では、パフォーマンスの観点から見たデータベースサービスと、各サービスのユースケースについて説明します。

1 Amazon Relational Database Service（RDS）

Amazon RDSは、マネージド型のリレーショナルデータベースサービスです。RDSのユースケースと特徴は次のとおりです。

●RDSのユースケース

リレーショナルデータベース（RDB）はデータを表形式で管理し、表と表の関係（リレーション）を定義することができます。複数の表でデータを管理することで、複雑なデータの関連性を扱うことが可能になります。

業務システムなど大量のデータを扱う場合は、従来ではリレーショナルデータベースにデータを格納するのが一般的でした。現在では、NoSQLのデータベースが台頭してきていますが、依然としてリレーショナルデータベースが各所で利用されており、企業の基幹システムや情報システムをはじめ多様な分野のシステムで使われているデータベースの1つであることに変わりはありません。

Amazon RDSは、そうしたリレーショナルデータベースの機能を必要とする場合に利用します。

●RDSのパフォーマンス

Amazon RDSのパフォーマンスは、主にスペック（インスタンスやストレージの種類）とRDS固有の機能である**リードレプリカ**（97ページを参照）によって高速化することができます。

試験対策

データベースのデータ参照が多い場合は、リードレプリカを利用することでパフォーマンスを向上させることができます。

2 Amazon DynamoDB

Amazon DynamoDBは、キーバリュー型のNoSQLデータベースサービスです。DynamoDBのユースケースと特徴は、次のとおりです。

●DynamoDBのユースケース

DynamoDBを利用することの大きなメリットの1つは、高い信頼性とパフォーマンスを維持できる点にあります。そのため、**可用性**が求められ、大量データの処理が必要な場合に向いています。たとえば、**モバイルゲーム**や**アドテクノロジー**（広告配信システム）、**IoTでのセンサーデータ**用のデータベース、Webアプリケーションのセッション管理データベース、などで利用できます。

●Amazon DynamoDB Accelerator（DAX）

Amazon DAXは、Amazon DynamoDB用のインメモリキャッシュです。

DynamoDBは高性能なデータベースであり、数ミリ秒のレイテンシーで動作しますが、DynamoDBとDAXを一緒に利用することで、1秒あたり100万回単位のリクエストの応答時間を数ミリ秒で処理できるようになります。

たとえば、Amazon EC2上で動作するアプリケーションが頻繁にDynamoDBのデータを参照するような場合、EC2とDynamoDBの間にDAXを配置し、そこにDynamoDBのデータをキャッシュすることで、DynamoDBへのアクセスを低減して、より高速に応答できるようになります。

【DynamoDBとDAXを一緒に利用するイメージ】

試験対策

Amazon DynamoDBの特徴とユースケースを覚えておきましょう。

●Amazon DynamoDB Auto Scaling

データベースにかかる負荷を事前に予測することは困難です。また、その予測できない負荷を想定して、手動でリソースのスケーリングを設計するという方法も現実的ではありません。

Amazon DynamoDB Auto Scalingは、DynamoDBへの負荷に応じて書き込みキャパシティや読み込みキャパシティを動的に調節します。これにより、急激な負荷の増減に自動的に対応し、データベースのパフォーマンスを低下させることなく、効率よくコストを抑えることができます。

●Amazon DynamoDBグローバルテーブル

DynamoDBグローバルテーブルは、複数リージョンに配置しているデータベーステーブルでデータの整合性を確保するフルマネージド型のソリューションです。

DynamoDBのテーブル作成時にグローバルテーブルを選択し、テーブルを作成したいリージョンを指定するだけで、複数リージョンに同一テーブルを作成することができます。

このテーブルは、リードレプリカのように1つがプライマリテーブルで、そのデータをほかのレプリカテーブルに同期しているわけではありません。すべてのテーブルがプライマリテーブルのため、どのリージョンのテーブルに対してデータ変更をしても、他リージョンのテーブルに対してデータの同期が行われます。

試験対策

Amazon DynamoDBのグローバルテーブルのユースケースについて確認しておきましょう。

参考

DynamoDBのグローバルテーブルのユースケースは、以下を参照してください。

https://docs.aws.amazon.com/ja_jp/amazondynamodb/latest/developerguide/GlobalTables.html

3 Amazon Redshift

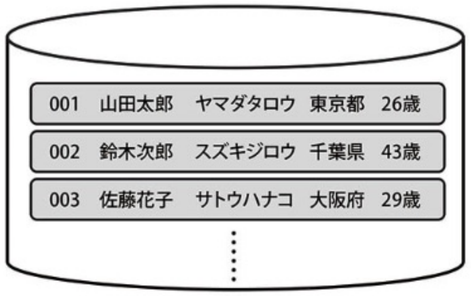
Amazon Redshiftは、ペタバイト規模のデータを標準SQLで分析可能なマネージド型のデータウェアハウスサービスです。

SQLでデータにアクセスできるという点は一般的なリレーショナルデータベースと同じですが、次のような特徴を備えています。

Oracle DatabaseやMySQL、PostgreSQLなどのRDBは行指向データベースといいます。行指向データベースは、頻繁にデータベースにアクセスして、行単位でデータを追加・更新・削除したり、一部のデータを検索したりすることに適した構造になっています。

ただし、RDBではデータ量が増加すればするほど、応答速度(レスポンス)が低下することが一般的です。たとえば、大量に格納されたレコードの金額をすべて参照して、その合計金額を集計するような処理も遅くなります。これでは、ビッグデータを集計・分析するような処理には利用することができません。

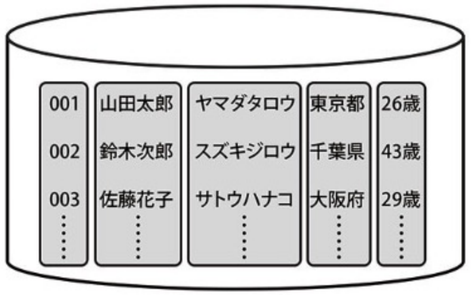
【行指向データベース】



一方、近年に新たなアプローチとして登場したのが**列指向データベース**です。列指向データベースは、データ分析を高速に処理できるように最適化されています。

列単位でデータを扱うため、対象データのある列に格納された値をすべて集計するような処理を高速に実行できます。Redshiftは列指向のアーキテクチャで、データの集計・分析に向いています。

【列指向データベース】



●Redshiftのユースケース

Amazon Redshiftは、大量データの保持やデータウェアハウス、BIツールなどによる分析に向いています。

●Amazon Redshift Concurrency Scaling

Redshiftは、大量の業務データをBIツールで分析したり、リアルタイムのストリーミングデータを分析する業務で使用されます。このRedshiftへの同時実行クエリが大量に発行されると、処理しきれないクエリはキューに格納され、Redshiftでの処理が可能となるリソースが確保されるまで待機状態になり、待ち時間が発生します。

このような場合の選択肢として登場したのが、Amazon Redshift Concurrency Scalingです。この機能を有効にすると、バースト性のある(突発的に負荷が高まる可能性のある)ユースケースが発生したときに、事前に設定した範囲で自動的にスケーリングしてクエリ処理を行います。

4　Amazon ElastiCache

Amazon ElastiCacheは、キーバリュー型のNoSQLデータベースサービスです。インメモリデータベースのMemcachedとRedisをデータストアとして利用することができますが、この両者には内容に違いがあります。

それぞれのデータストアの特徴と選択基準は、次のとおりです。

【MemcachedとRedisの特徴と選択基準】

データストア	Memcached	Redis
特　徴	● 一時的なデータのキャッシュ用として使用 ● ノード間の複製は行われない ● データベースを別途用意	● プライマリ／レプリカ型の構成 ● データストアとしても利用可能
選択基準	● シンプルなデータ構造が必要 ● 複数のコアまたはスレッドを持つ大きなノードの実行が必要 ● スケールアウトおよびスケールインの機能が必要 ● データベースなどのオブジェクトのキャッシュが必要	● ハッシュ型やリスト型などの複雑なデータ構造を使用 ● プライマリノードに障害が発生した場合は、自動的にフェイルオーバーされることが必要 ● 読み取るデータ量が多いため、プライマリノードから複数のリードレプリカに複製 ● 永続性が必要

●ElastiCacheのユースケース

ElastiCacheは、一般的にRDBのパフォーマンス向上のために利用されます。

RDBでは、データ量の増加やアクセス集中によってレスポンスが遅くなる可能性があるため、クエリの結果をElastiCacheに保存しておき、同じクエリが実行された場合にElastiCacheから結果を返すことで、RDBの負荷低減とシステムのレスポンス向上を実現できます。

たとえば、Webアプリケーションのセッション情報や、データベースに対して頻繁にアクセスするSQLクエリをキャッシュするのに利用します。

MemcachedとRedisの詳細な違いは、以下を参照してください。
https://docs.aws.amazon.com/ja_jp/AmazonElastiCache/latest/mem-ug/SelectEngine.html

5 データベースサービスの比較

本節で説明したAWSのデータベースサービスの特徴をまとめ、次の比較表に示します。

【各データベースサービスの比較】

項目＼サービス		RDS	DynamoDB	Redshift	ElastiCache
製品タイプ	RDB	○	―	○	―
	NoSQL	―	○	―	○
特　徴		● 一般的なRDB ● 頻繁なデータ更新、トランザクション処理などに適している ● 複雑なSQLを発行してデータを抽出可能	● キーバリュー型データ構造 ● 半構造化データを格納することができる	● 大容量データの集計、分析を高速に処理できる	● メモリ上にデータを格納するため高速
用　途		企業の基幹システム	モバイルゲーム、IoTデータ管理	BIツールなどでの集計・分析	データベースのデータキャッシュ

Q 演習問題

1 **ある企業では、毎週BIツールを使用してビジネスレポートを作成しています。BIツールで使用されるデータはオンプレミス環境のサーバー上に格納されていましたが、そのデータをすべてAWSに移行する計画が企画されています。より効率的にデータ分析を行うためのデータ格納ソリューションは、次のうちどれですか。**

A　Amazon RDS

B　Amazon DynamoDB

C　Amazon Redshift

D　Amazon Aurora

2 **ある企業のアプリケーションチームは、スケーラブルなWebアプリケーションを設計しており、ユーザーセッション情報を管理するサービスを検討しています。これらの要件を満たすことができるサービスは、次のうちどれですか。（2つ選択）**

A　Amazon Redshift

B　Amazon RDS

C　Amazon ElastiCache

D　Amazon DynamoDB

3 **ある企業が提供しているコンシューマー向けのアプリケーションでは、ユーザー情報の管理にAmazon DynamoDBを使用しています。組織の方針で、今後は他のリージョンでもアプリケーションを展開することになり、そのためのソリューションを検討する必要があります。最小の労力でAmazon DynamoDBのデータを展開し同期するには、次のうちどの方法が適切ですか。**

A　現在のDynamoDBのバックアップを取得し、各リージョンで復元する

B　各リージョンにDynamoDBテーブルを作成し、現在のDynamoDBテーブルとDynamoDB Replication機能で同期する

C　DynamoDBグローバルテーブルを使用して、各リージョンに展開する

D　ユーザー情報の管理をDynamoDBからAmazon Auroraに移行し、Auroraグローバルテーブルを使用する

A 解答

1 C

Redshiftについての知識を問う問題です。
A、Dは行指向データベースのため、効率のよいデータのクエリには向かないため不正解です。BはRDBではなくNoSQLデータベースで、複雑なリレーションを持つデータには適さないため不正解です。CのRedshiftは列指向データベースで、分析のために効率よくデータをクエリするのに適しています。したがって、**C**が正解になります。

2 C、D

サービスのユースケースについての知識を問う問題です。
選択肢のうち、AとBはRDBまたは分析用途に提供されるデータベースのため、セッション情報を格納し頻繁に読み書きするアプリケーションには向いていません。
CのAmazon ElastiCacheとAmazon DynamoDBはシンプルなデータ構造と頻繁な読み書きに対応したサービスです。したがって、**C**と**D**が正解になります。

3 C

DynamoDBのグローバルテーブルについての知識を問う問題です。
Aは別のリージョンへの展開までは可能ですが、データ同期はできないため不正解です。BはDynamoDB Replicationという機能はないため不正解です。
Cは最小限の労力でデータを展開、同期できる構成です。
Dは移行に時間がかかりすぎるため不正解です。したがって、**C**が正解になります。

Column

AWS認定試験の全冠と学習観点

2022年12月現在、AWSの認定試験は全部で12種類が存在しており、基礎レベルから専門レベルまで幅広いAWSの知識が求められる構成になっています。

これらの資格を取得した人は、棋士のタイトルホルダーのように「○冠」と呼ばれ、なかでもすべてのAWS認定資格を取得した人は「全冠」と呼ばれています。

AWSでは、パートナー企業に所属する全冠取得者を表彰しており、2020年は14名、2021年は91名、2022年は340名が表彰されています。これらの取得者以外にも、取得を申請していない人やパートナー企業に所属していない受験者もいるため、実際にはさらに多くの全冠取得者がいると思われます。

もし、あなたが全冠を目指そうとした場合、おそらく各AWS資格について、まず調べるのではないでしょうか。現在では、AWS認定試験の学習方法は多岐にわたり、AWS公式ドキュメントや本書のような資格試験の対策書籍など、さまざまな方法が選択できます。最適な学習方法は人によって異なるため、これをやれば全冠制覇できるという特定の方法はありませんが、学習に際して意識しておくと良い観点はあります。

個人的には「AWSが推奨している設計(ベストプラクティス)を理解しながら学習する」ことが、特に重要だと考えています。AWS認定試験は12種類もあるため、出題されるすべてのAWSサービスの機能を1つずつ覚えるには、相当な時間と労力が必要になります。

そうした個々のAWSサービスの機能を覚えることも重要ですが、「各サービスをどのように使えば推奨の設計にできるのか」を正しく理解することで、各試験に共通した観点だけでなく、実際のAWS上でのシステム設計・構築に必要な知識を身に付ける最短距離になると考えています。

AWS
Solutions Architect–Associate

第5章

AWSにおけるコスト最適化

5-1 AWSにおけるコスト最適化の考え方
5-2 コスト効果が高いリソースの選定
5-3 需要と供給のマッチングによるコスト最適化
5-4 コストの管理

5-1 AWSにおけるコスト最適化の考え方

AWSをはじめとするクラウドサービスでは、従量課金でITリソースを柔軟に活用できる反面、コストの適切な管理や最適化を行わなければ、想定外の料金になる場合があります。
本節では、AWSにおけるコスト最適化の考え方について説明します。

1 クラウドにおけるコスト最適化の重要性

近年、日本においてもクラウドサービスの利用は増加傾向にあります。たとえば、総務省が公表している『令和3年版 情報通信白書』によると、2020年時点で約7割の日本企業が何らかのクラウドサービスを利用しており、導入率は年々増加傾向にあることがわかっています。

その一方で、クラウド利用の増加と比例して、利用コストに関する問題も多く指摘されています。たとえば、Enterprise Management Associates(EMA)の調査によると、クラウドを活用している企業が抱えるクラウド関連の課題のうち、コスト管理に関する項目が最も優先度が高い[※1]ことが報告されています。

具体的には、実際にクラウドサービスの利用で課金されている内容のうち、約4〜5割は不用意に使われている、つまりコスト最適化の余地があるということです。

次に示す状況は、クラウドの利用コストが増加する要因の一例です。

- 仮想マシンやストレージがほとんど使われていない
  例：約3割の仮想マシンが、実際にはあまり利用されない状況でも起動している

- 不必要に24時間365日稼働している
  例：稼働が不要な時間帯でも仮想マシンが起動している

※1 Optimize Cloud Costs Through Governance and Collaboration, EMA Report 2017/03

- 適切なスペックのサイジングを割り当てていない
  例：オーバースペックなサイズのまま仮想マシンを稼働させている

- 割引オプションを利用していない
  例：仮想マシンの割引オプションを有効活用できていない

2 AWSにおけるコスト最適化の指針

では、AWSではどのようにコスト最適化に取り組むべきでしょうか。

AWS Well-Architectedフレームワークの**コスト最適化**の柱では、コストを最小限に抑えるための指針として、次に示す5つの設計原則が掲げられています。

●必要なリソースを必要なときに必要な分だけ利用する

従来のオンプレミス環境では、あらかじめ想定された量のトラフィックを処理できるように、将来を見据えて余裕を持ったスペックのサーバーを購入・構築するため、初期費用がかさんでいました。

一方でクラウド環境では、「必要に応じてリソースを柔軟に拡張」できるため、最初は必要なリソースに合わせてスモールスタートの環境を構築し、将来的にはビジネスの拡大に合わせて環境を増築することで、コスト抑制が可能です。

また、「使った分だけ課金」されるため、たとえば、夜間など「開発環境やテスト環境を利用していない時間帯は仮想マシンを停止する」といった運用を行うことで、コストを削減できます。

●全体的なコスト効果を測定する

ビジネスから生まれる成果や価値と、それに費やしたクラウドのコストの両面を測定しながら、「ビジネス価値÷コスト」が最大化するようなコスト最適化を行うことが重要です。

たとえば、AWSで利用しているサービスの単なるコスト削減によりビジネス価値が大きくなるのか、あるいは、コストが増大してもAWSサービスの活用を増やすことでビジネスをスピーディに展開し、ビジネス価値が改善するのかなど、単なるコスト削減の視点ではなく、全体的な利用価値・コスト効果の視点から最適化を進めることが重要です。

●データセンター運用への投資を不要に

クラウド活用により、企業の競争力に直接影響しないデータセンター内のITインフラ構築・運用(たとえば、サーバーのラッキングなど)はAWSが担当するため、企業は競争力の源泉となるビジネスの推進や開発に集中することができます。

●投資効果の要因分析

AWSで利用したITリソースの使用量と、それを使った組織、チーム、プロジェクトが簡単に把握できるため、投資効果(ROI：Return on Investment)を組織やチーム、プロジェクトなどの細かい単位で局所化して把握しながら、問題の要因分析やコスト最適化などの改善が実施できるようになります。

●マネージドサービスの活用による所有コストの削減

電子メールやデータベースのサーバーなど、従来では自社で構築・運用していたITインフラをまるごとAWSのマネージドサービスに代替することで、サーバー機器の保有や保守を含めた所有コストが削減できます。

3 コスト最適化の4つのベストプラクティス

AWSには、前項で説明したコスト最適化の指針を具体的に実現する手段として、次に説明する4つのベストプラクティスが提示されています。これらのベストプラクティスを参考にすることで、効果的にコスト最適化を進めることができます。

●コスト効果が高いリソースの選定

適切なサイジングを割り当てる、割引オプションを使用する、マネージドサービスを利用するなど、AWSサービスを活用するうえで最もコスト効果の高いリソースの選定方法を説明します。詳細は、次節の「5-2　コスト効果が高いリソースの選定」を参照してください。

●ITリソースの需要とAWSサービスの適切な供給によるコスト最適化

AWS Auto Scalingなどを活用し、需要に合わせてAmazon EC2インスタンスを供給するというように、需要と供給を適切にマッチングさせることでコストを最適化する方法を説明します。詳細は「5-3　需要と供給のマッチングによるコスト最適化」を参照してください。

●全体的なコストの管理

コストの深掘り調査やコストを超過した場合の通知方法など、コスト管理全般の方法論を説明します。詳細は「5-4　コストの管理」を参照してください。

●継続的なコスト最適化の活動

コストの測定基準や継続的な監視・改善など、コスト最適化のサイクルをどのように進めていくかを説明します。詳細は「5-4　コストの管理」を参照してください。

次節以降から、各ベストプラクティスの内容を説明していきます。AWSにおけるコスト最適化の概念(5つの設計原則と4つのベストプラクティス)は、具体的なコスト最適化の手段を理解する前提として重要です。

Q 演習問題

1 **ある企業では、AWS上で構築したシステムが継続的に正しく動作し、かつ必要なときに必要な分だけリソースを利用できるよう、無駄のない全体最適なクラウド アーキテクチャを設計したいと考えています。次のAWS Well-Architectedフレームワークのうち、どのベストプラクティスを指していますか。正しいものを1つ選びなさい。**

A　パフォーマンス

B　コスト最適化

C　運用上の優秀性

D　セキュリティ

A 解答

1 B

AWS Well-Architectedフレームワークでは、「コスト最適化」の柱で、コストを最小限に抑えるための指針として5つの設計原則が掲げられています。
したがって、**B**が正解です。

5-2 コスト効果が高いリソースの選定

AWSでは、コスト効果が高い各種のリソースや購入オプションを提供しています。
本節では、どのような種類のリソースや購入オプションが提供されているか、また、コスト最適化の観点からそれらをどのように選定するかを説明します。

1 AWSサービスの購入オプション

AWSでは、サービス利用時にさまざまな購入オプションを提供しています。企業がAWSサービスを利用する際には、ビジネス計画やシステム特性に合わせたコスト効果の高い購入オプションを選定することができます。

では、具体的にどのような選択肢があるのでしょうか。

Amazon EC2やAmazon RDS、Amazon Redshiftなど、主に使用した時間に対して課金されるコンピューティングリソースを利用するサービスでは、次の表に示す3つの購入オプションが提供されています。

【AWSにおける購入オプションの一覧】

購入オプション	主な特徴	主なサービス
オンデマンドインスタンス（デフォルト）	初期費用なしで使用した分だけの従量課金	Amazon EC2、Amazon RDS、Amazon Redshiftなど
リザーブドインスタンス	長期利用権（1年または3年）による購入割引（最大で75%オフ）がある。また、ある特定期間（日次、週次、月次など）を指定できる「スケジュールされたリザーブドインスタンス」も購入が可能。	Amazon EC2、Amazon RDS、Amazon Redshiftなど
スポットインスタンス	AWSの余剰リソースを安価に利用（最大で90%オフ）。ただし、強制終了される場合がある。	Amazon EC2、Amazon EMRなど

次に、3つのオプションの特徴と選定方法について説明します。

●オンデマンドインスタンス(デフォルト)

Amazon EC2などのインスタンスを利用する場合、標準では**オンデマンドインスタンス**が適用されます。

オンデマンドインスタンスは、あらかじめ決められた一定レートの料金で使用した時間に対して課金されるため、使用期間(たとえば、1カ月間など)の制約はありません。

主な用途として、開発・テスト環境のような定時(平日の日中など)の時間帯しか使用しないサーバー群や、キャンペーン時の一時的なWebサイトなどが考えられます。

> 参考
> オンデマンドインスタンスの最新の料金一覧は、AWSのサイト[※2]に掲載されています。たとえば、東京リージョンでvCPUが2コア、メモリが8GBのAmazon EC2インスタンス(m5.large)を利用する場合は、1時間あたり0.124ドルです(2022年12月時点)。

> 参考
> Amazon EC2やAmazon EMRなどのサービスでは、秒単位で使用した分が課金されます。ただし、AWSのサービスによって課金単位は異なります。

●リザーブドインスタンス

オンデマンドインスタンスと異なり、あらかじめ決められた使用期間(1年または3年)分を購入することで、最大72%オフの割引価格が適用されます。

支払い方法は「全額前払い」「一部前払い」「前払いなし」の3種類のオプションから選択できます。ただし、「全額前払い」では、途中で利用をキャンセルしても払い戻しされません。

リザーブドインスタンスの主な用途としては、最低限必要な台数があらかじめ決まっているWebサーバーやアプリケーションサーバー、また常時起動しておく必要があるデータベースサーバーなどが考えられます。

Amazon EC2ではリザーブドインスタンスがよく利用され、「Standardタイプ」と「Convertibleタイプ」の2種類が提供されています。

※2　https://aws.amazon.com/ec2/pricing/on-demand/

Standardタイプは、リージョンやアベイラビリティーゾーン(AZ)を指定してインスタンスを購入できます。

同じインスタンス構成(ファミリー、OSなど)であれば、購入時に指定したリージョン内またはAZ内でインスタンスの配置を変更できますが、それ以外の場所に変更する場合は手続きが必要になります。

Convertibleタイプは、あらかじめ指定したインスタンス構成(ファミリー、OSなど)に依存せず、柔軟に構成変更(ただし、作成時の価格と同等以上)が可能です。その分だけ割引率は、Standardタイプより低く設定されています。

●スケジュールされたリザーブドインスタンス

リザーブドインスタンスの重要な機能の1つに、**スケジュールされたリザーブドインスタンス**があります。

従来よりAWSで提供されているリザーブドインスタンスは、購入すると24時間×365日の継続・常時稼働を前提として割り引きが適用されています。この購入オプションは、たとえば、外部向けWebサーバーのように常時稼働が前提となっているAmazon EC2インスタンスには適していますが、「営業時間中にのみ稼働する」「週末や月末のバッチ処理でのみ稼働する」といった、ある特定の期間だけEC2インスタンスを稼働させたい場合には適しません。

このような特定の期間に限定して稼働したいというユーザーのニーズに応え、AWSでは「スケジュールされたリザーブドインスタンス」が提供されるようになりました。これによりユーザーは、日次、週次、月次のようにある特定の時間や期間で、リザーブドインスタンスを購入することができます。

3つのオプションの特徴を次にまとめます。

- Standardタイプは、割引率が最も高く、定常状態の仕様に最適
- Convertibleタイプは、インスタンス構成を柔軟に変更可能(ただし、割引率は低い)
- スケジュールされたリザーブドインスタンスは、特定の期間に限定して購入

試験対策

リザーブドインスタンスの主な用途と3種類のタイプの違いを理解しましょう。

参考

Amazon EC2などの期間購入のリザーブドインスタンスと同様の考え方で、Amazon CloudFrontやAmazon DynamoDBでは「リザーブド キャパシティ」と呼ばれる、あらかじめ決められた期間(1年など)分のキャパシティ(10TBなど)を購入する割引オプションが提供されています。

●Savings Plans

Amazon EC2のインスタンスを購入するリザーブドインスタンスに加えて、EC2のインスタンス使用量を購入することで割引が受けられるSavings Plansという新しいプランも登場しています。

Savings Plansでは、1年または3年の期間で特定の使用量(USD/時間で測定)を契約すると、オンデマンドインスタンスよりも安価になる料金モデルです。

このプランには、Compute Savings PlansとEC2 Instance Savings Plans、Amazon SageMaker向けのAmazon SageMaker Savings Plansの3種類が用意されています。

● Compute Savings Plans

リージョンやインスタンス構成に関係なく、すべてのEC2インスタンスを単位時間あたりの使用量を指定して購入できます。

Amazon EC2だけでなく、AWS LambdaやAWS Fargateも対象に含まれます。

● EC2 Instance Savings Plans

リザーブドインスタンスのように、リージョンやインスタンス構成と単位時間あたりの使用量を指定して購入できます。

● Amazon SageMaker Savings Plans

Amazon SageMaker向けのSavings Plansです。

●スポットインスタンス

スポットインスタンスは、Amazon EC2の余剰リソースを入札形式で安価に利用する方法で、割引率が最も高い(最高で90%オフ)購入オプションです。

変動するEC2インスタンスのスポット価格(マーケット価格)に対し、希望の価格で入札します。スポット価格を上回っていればインスタンスが起動し利用できますが、AWS内でインスタンスの需要が高まりスポット価格が高騰して入札価格を上回ると、インスタンスが強制終了されます。

スポットインスタンスは、処理が中断されても特に支障なく再実行が可能なシステムに適用されます。たとえば、開発環境やメディアのエンコード処理を並列化するために利用されるEC2インスタンス、分析のため分散処理を行うAmazon EMRのタスクノードなどが考えられます。

スポットインスタンスは、必要に応じて「スポットブロック」と「スポットフリート」の2種類のオプションを選択することができます。

スポットブロックは、継続期間を意味します。スポットブロックが設定されたスポットインスタンスでは、インスタンス起動後、設定された時間に対して継続して動作することを保証するオプションとなります(最大6時間まで)。

割引率は通常のスポットインスタンスよりも低いですが、処理途中における想定外の強制終了を回避できます。

スポットフリートは、あらかじめ指定した性能キャパシティを満たすように、スポットインスタンスを構成するオプションです。

たとえば、スポット価格の高騰で指定したスポットインスタンスが使用不可になった場合でも、代替構成で全体の性能キャパシティを維持します。

スポットとリザーブドのインスタンスの違いについて、次にまとめます。

- スポットインスタンスは、入札形式で最も安く利用できますが、インスタンスが強制終了されるリスクがあるため、一時的に利用したい用途に適します。
- リザーブドインスタンスは、長期的な利用で割引が受けられ、システムの安定・常時稼働が求められる用途に適します。

試験対策

スポットインスタンスの主な適用用途と、強制終了される場合の条件(スポット価格が入札価格を上回る)は重要です。

参考

実際にはインスタンスの強制終了前に、2分間の警告期間が設けられています。この時間内に、インスタンスの状態保存など終了タスクを設定できます。

2 リソースの適切なサイジング

AWSでは、多様なシステム要件やユースケースをサポートするために、大小さまざまなリソースのタイプを提供しています。

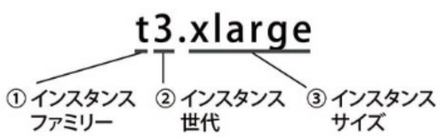
ここでは、Amazon EC2に代表されるインスタンスとAmazon S3に代表されるストレージのサイジングについて説明します。

●適切なインスタンスタイプの選定

Amazon EC2 や Amazon RDS、Amazon Redshift、Amazon OpenSearch Service(旧Amazon Elasticsearch Service)などのサービスでは、**インスタンスタイプ**にさまざまなサイジングの選択肢が提供されています。

たとえば、EC2ではインスタンスタイプは次のように構成されています。

【インスタンスタイプの構成】



① インスタンスファミリー

インスタンスファミリーとは、EC2などのコンピューティングリソースの利用用途や特性に合わせた、さまざまなサイジングの種類のことです。

主なインスタンスファミリーは次の表に示すとおりです。

【インスタンスファミリーの種類の例】

インスタンスファミリー	主な種類	特　徴
汎用	T2、T3、M4、M5	一般的なシステム用途に適用可能なバランス型
コンピューティング最適化	C4、C5	コスト効果の高いCPU処理に特化
メモリ最適化	X1e、X1、R5	大容量のメモリに特化
高速コンピューティング	P2、P3、G3、F1	GPUやFPGAなど高速処理に特化
ストレージ最適化	H1、I3、D2	ストレージ容量やI/Oスループット性能に特化

② インスタンス世代

M4やM5など、種類の表記に付与されている数字は**インスタンス世代**と呼ばれています。数字が大きいほど新しい世代を意味しており、一般的には最新のアーキテクチャが使われているため、高性能かつ安価なケースが多くなります。

参考

同等のサイジングであるm4.largeとm5.large(vCPUが2、メモリが8GB)のオンデマンドインスタンス料金を比較した場合、前者が1時間あたり0.129ドル、後者が1時間あたり0.124ドルと、最新世代のm5のほうが安価(かつ高性能)です。(2022年12月現在)

③ インスタンスサイズ

各インスタンスファミリーには、largeやxlarge、2xlargeなどのインスタンスのサイズを示す単語が付与されています。

たとえば、c5.4xlargeはコンピューティング最適化のC5インスタンスファミリーで、vCPU数が16個搭載されたサイジングを意味します。

参考

用途に合わせた適切なインスタンスファミリーの選択とインスタンス世代・サイズの変更により、コスト最適化を実現できます。

参考

一般的にvCPU数は、largeが2、xlargeが4、2xlargeが8、4xlargeが16といったように増加していきます。

●適切なストレージ選定

Amazon EC2だけでなく、Amazon S3やAmazon S3 Glacierに代表されるオブジェクト ストレージサービスや、EC2インスタンスにアタッチして使用するブロック ストレージサービスのAmazon EBSにおいても、コスト最適化を意識した適切なストレージ選定が必要です。

● S3 のストレージクラス

S3では、ストレージの利用頻度などの用途に合わせた最適なストレージクラスを選択することで、コスト最適化が可能です。

S3では、用途別に次のようなストレージクラスが提供されています。

① 標準(Standard)

S3では、デフォルトが標準(Standard)クラスになります。標準クラスは、複数箇所にデータを複製することで、99.999999999%(イレブンナイン)の耐久性を実現しています。

② 標準ー低頻度アクセス(Standard-Infrequent Access)

標準クラスと同等の耐久性を備え、かつデータの格納コストが標準クラスと比較して安価です。ただし、データの読み出し容量に対しても課金されるため、データへのアクセス頻度が低い場合に適しています。

③ Glacier

主にS3上でのファイルアーカイブに利用されるストレージで、**最も安価**です。耐久性は前述した2クラスと同等です。ただし、データの取り出しに通常は3〜5時間を要するため、データを一度保存したあと年に1〜2回しかアクセスしないようなバックアップ目的での利用に適しています。

Glacierよりもさらに安価な**Glacier Deep Archive**というクラスも提供されています。このクラスはデータの取り出しに12時間以上を要するため、データ取り出しの緊急度が高くない場合に利用します。

一方で、データを即時に取り出したい場合は、**Glacier Instant Retrieval**というサービスが利用できます。

④ 1ゾーンー低頻度アクセス(One Zone-Infrequent Access)

1つのAZのみにデータを保存します。複数のAZにデータを複製する標準クラスと比較すると、コストを約20%削減できます。

データへのアクセス頻度が低く、高い耐久性を必要とせず、必要に応じてすぐにデータを取り出したい場合に適しています。

⑤ Intelligent-Tiering

S3上に格納したオブジェクトへのアクセス頻度に応じて、コスト最適なストレージ層を自動的に使い分けるタイプです。

具体的には、コスト構造の異なる低頻度と高頻度の2階層のストレージ層があり、オブジェクト別にアクセス頻度に応じてS3が自動的にオブジェクトの階層を移動させることで、コストを自動的に最適化してくれます。30日間アクセスされないオブジェクトは低頻度層に移動し、アクセスされると自動的に高頻度層に移動します。

試験対策

Amazon S3の5種類のストレージクラスの可用性やアクセス頻度、コスト観点の違いを押さえておきましょう。S3では、可用性やアクセス頻度に応じてストレージクラスを使い分けることでコスト最適化が可能です。また、長期保存向けなど、アクセス頻度が低く取り出しに時間を要してもよい場合は、Glacierが最も安価な選択になります。

● EBS のボリュームタイプ

Amazon EBSは、EC2インスタンスに接続して利用できるブロックレベルのストレージです。OSから見えるストレージボリュームで、アプリケーションやユーザーデータなどを格納します。

Amazon S3と同様に、EBSにおいても用途に応じて以下の5種類のタイプ(①、②はSSDタイプ、③〜⑤はHDDタイプ)が提供されています。

① 汎用SSD(General Purpose SSD：gp2およびgp3)

EBSのデフォルトのボリュームタイプです。費用対性能比が高く、I/O性能も最大16,000IOPS程度まで利用できます。OSのブート領域や中規模のデータベース用途など、汎用的に利用できるのが特徴です。

② プロビジョンドIOPS SSD(PIOPS SSD：io1およびio2)
最もパフォーマンスの高いタイプのEBSで、最大64,000IOPSまで指定できます。汎用SSDよりも高いI/O性能が求められるケースでの利用、たとえば、大規模データベースなどでの利用が想定されます。なお、このタイプのみストレージ容量だけでなく、指定したIOPS数に対しても課金されるため注意が必要です。

③ スループット最適化HDD(st1)
Amazon EMRによるログ分析など、主にファイルへのシーケンシャルアクセスが多い場合に高いスループットを提供するタイプです。ビッグデータ処理などに向いており、価格もSSDの2タイプと比較して安価です。

④ コールドHDD(sc1)
高いスループットが不要な場合には、より低価格なコールドHDDのタイプも使用できます。ログファイルの保管など、高スループットを求められないバックアップ領域などが主な用途です。

⑤ マグネティック
旧世代のEBSのタイプです。汎用SSDが登場する前はデフォルトのタイプでしたが、現在も利用可能です。

試験対策

プロビジョンドIOPS SSDでは、実際に利用しているストレージ容量に対する課金に加え、IOPS数にも課金されるため、コスト面で注意が必要です。

Q 演習問題

1 **ある企業では、Amazon EC2上でアプリケーションを運用しています。このアプリケーションは24時間365日かつ長期間稼働していて、常に一定のアクセス負荷があります。コストを最適化するために、EC2のどの購入オプションを選択すべきでしょうか。**

A　リザーブドインスタンス

B　オンデマンドインスタンス

C　スポットインスタンス

D　スケジュールされたリザーブドインスタンス

2 **ある企業では、複数のAmazon EC2上でアプリケーションを運用しています。このアプリケーションはログ分析の分散バッチ処理を行っており、EC2が障害などで停止した場合でも単独で再実行できるように設計されています。コストを最適化するために、EC2のどの購入オプションを選択すべきでしょうか。**

A　リザーブドインスタンス

B　オンデマンドインスタンス

C　スポットインスタンス

D　スケジュールされたリザーブドインスタンス

3 **ある企業では、Amazon EC2のインスタンスサイズがc4.8xlargeで、I/O処理が多い分析アプリケーションが稼働しています。CPU使用率が低い場合、コストを最適化しつつ分析アプリケーションのパフォーマンスを向上させるには、どうしたらよいでしょうか。**

A　より小さいサイズでI/O性能が高いc5インスタンスに変更する

B　より大きいサイズのc4インスタンスに変更する

C　メモリ最適化されたx1eインスタンスに変更する

D　Amazon S3のストレージクラスをGlacierに変更する

4 **ある企業は、500TBのログファイルを保持しています。不定期にログファイルにアクセスする必要があり、通常は30分以内にログデータを取得したいと考えています。次のAmazon S3ストレージクラスのうち、最もコスト効果が高いクラスはどれですか。**

A　低冗長化ストレージ

B　スタンダード

C　標準－低頻度アクセス

D　Glacier（標準データ取り出し）

5 **IOPSに対する課金があるAmazon EBSボリュームタイプは、次のうちどれですか。**

A　汎用SSD

B　スループット最適化HDD

C　コールドHDD

D　プロビジョンドIOPS SSD

6 **Amazon EC2上のデータベースサーバーで読み込み、書き込みが頻繁に発生している場合、性能を一定に保つことができ、かつ最もコスト効果の高いAmazon EBSボリュームタイプは、次のうちどれですか。**

A　コールドHDD

B　プロビジョンドIOPS SSD

C　スループット最適化HDD

D.　汎用SSD

A 解答

1 A

リザーブドインスタンスは、あらかじめ長期間（1年または3年）の予約購入を行うことで、オンデマンドインスタンスに比べて大幅な割引価格（最大で75％オフ）で利用できます。

2 C

スポットインスタンスは最も割引率が高い（最大で90％オフ）購入オプションで、中断されても特に支障なく再実行が可能なアプリケーションに向いています。

3 A

CPUスペックは下げることができます。このケースでは、I/O性能が向上しているインスタンスタイプ（c5）の利用が適切です。

4 C

S3の標準－低頻度アクセスのストレージクラスは、標準（Standard）と比較して安価なデータ格納コストで、低頻度なアクセスの場合にコスト効果が高くなります。

5 D

プロビジョンドIOPS SSDでは、実際に利用しているストレージ容量に対する課金に加え、指定したIOPS数に対しても課金されます。

6 B

プロビジョンドIOPS SSDは最もI/Oパフォーマンスの高いEBSボリュームタイプで、データベースサーバーなどに利用されます。

5-3 需要と供給のマッチングによるコスト最適化

パブリッククラウドのメリットの1つとして、需要に応じてリソースを柔軟かつ迅速に調達・供給できる点があげられます。この特徴を活かした需要と供給のマッチングをAWSサービスで実現することで、無駄なコストの削減が可能になります。
本節では、AWSにおいてリソースの需要と供給をうまくマッチングさせるための手法について説明します。

1 需要の変化に応じたITリソースの供給

AWSをはじめとするパブリッククラウドでは、使いたいときにリソースを簡単に調達できます。たとえば、Amazon EC2であれば、AWSマネジメントコンソール上でわずか数分で新しいインスタンスを購入・起動でき、時間課金で利用できます。

しかし、あまり使われていない状況でサービスを利用し続けていると、無駄な料金がかさみます。したがって、需要に即した適切なAWSサービスを利用することも、コスト最適化の観点では重要です。

次項から、需要と供給のマッチングを実現するAWSのサービスをいくつか紹介します。

2 ELBとAuto Scalingによる柔軟なリソース供給

Elasticity(伸縮性)という言葉があります。クラウドでは、全世界に設置されたデータセンターに膨大なリソースが配備されており、ユーザーは必要に応じてリソースを取得・拡張・縮小できます。この特性のことを伸縮性と呼んでいます。

AWSをはじめとするクラウドサービスでは、この伸縮性の特性を活かして、

トラフィックなどの需要の増減に応じてリソースを柔軟に増減させることができます。

AWSでは、AWS Auto Scalingというサービスを通して、この伸縮性をうまく実現しています。

●ELB＋Auto Scalingの組み合わせ方法

AWS Auto Scalingは通常、さまざまなトラフィックを処理するAmazon EC2インスタンスの負荷を分散するためにElastic Load Balancing(ELB)とともに利用され、負荷状況に合わせてEC2インスタンスの増減を制御します。

トラフィック量などの負荷状況に合わせた柔軟な制御を行うため、Amazon CloudWatchの監視メトリクスもあわせて使用します。

ELBとAuto Scalingを組み合わせた拡張方式の詳細については、「3-3 コンピューティングにおける高可用性の実現」を参照してください。

3 SQSやKinesisによるバッファリング

もう1つの手法として、バッファリングで処理を非同期化することで、需要と供給を管理する方法があります。

Auto Scalingの利用は、「発生する需要(負荷)の増減に応じて柔軟にリソースを自動増減すること」が主な目的です。

しかし、必ずしも同期的に処理する必要がない場合は、バッファリングなどの仕組みを利用して需要(負荷)を一時的に蓄積しておき、処理する側の必要に応じた間隔で非同期的に処理を進めるという方法もあります。

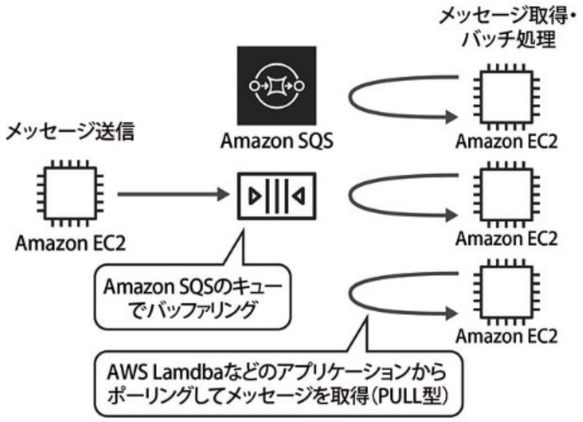
AWSでは、バッファリングをAmazon Simple Queue Service(SQS)やAmazon Kinesisで実現できます。

●Amazon Simple Queue Service(SQS)

Amazon SQSはメッセージキュー(MQ)のマネージドサービスです。

アプリケーション間でメッセージをキューイングすることで、疎結合のアーキテクチャを実現します。たとえば、複数のEC2インスタンスへ非同期に分散処理するケースなどでSQSを活用できます。

【SQSの処理イメージ】



SQSで重要な機能は、主に次の4つです。

●標準キューと FIFO キュー

SQSでは2つのタイプのキューを作成することができます。

標準キューでは、キューイングされたメッセージを取り出して処理・配信を行う際の順番が保証されません。このため、処理タイミングによっては、同一のメッセージが2回処理・配信される可能性があります。この特性によって、たとえば、動画のエンコーディング処理など順序性は関係なく並列で行うバッチ処理での利用が考えられます。

これに対して**FIFO(先入れ先出し)キュー**では、メッセージをキューイングした順番で処理・配信することが保証されますが、標準キューより処理性能が若干劣ります。業務上、順序性が重要となるワークフローなどへの適用が想定されます。

●ロングポーリング(Long Polling)

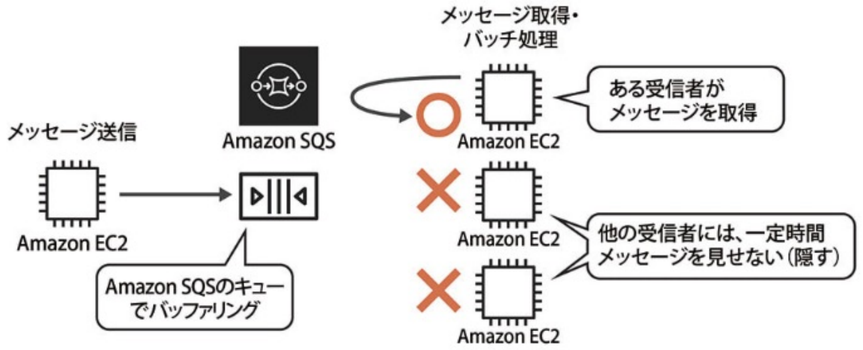
通常、SQSキューに対して受信者がメッセージ取得要求(Receive Messageリクエスト)を送ると、キューが空の場合でもEmptyメッセージが返送されます。この方式を**ショートポーリング**と呼びます。SQSはリクエスト単位で課金されるため、空振りが多い場合には、この方式はコストの観点から見ると非効率的です。

ロングポーリングでは、キューが空の場合にメッセージを取得できるまで待つ時間(1〜20秒)を設定することで、メッセージ取得要求の数を減らすことができます。

●可視性タイムアウト(Visibility Timeout)

ある受信者がメッセージを取得した場合に、ほかの受信者にはそのメッセージをある一定時間(デフォルトでは30秒間)見せないようにすることで、処理の重複を防止したり、リクエスト数を減らすことができます。

【可視性タイムアウトのイメージ】



●スポットインスタンスとの組み合わせ

可視性タイムアウトとスポットインスタンスを組み合わせることで、さらにコスト効果が高まります。

前述のとおり、スポットインスタンスは最も安価にEC2インスタンスを利用できますが、不意に強制終了される場合もあります。

メッセージ受信のアプリケーションにスポットインスタンスを使用することで、仮にスポットインスタンスが強制終了されたとしても、可視性タイムアウト(デフォルトは30秒)を超過すると自動的にほかの受信者がメッセージを取得できるようになるため、処理の再実行が可能になります。

SQSの機能では、標準キューはメッセージの順番が保証されず、同一メッセージが重複して配信される可能性があること、FIFOキューはメッセージを順番に配信できるが性能が若干劣ること、可視性タイムアウトを設定すると処理の重複を防止できることが、重要なポイントです。

試験対策

Amazon SQSによるバッファリング・非同期処理の特徴、2つのキュー(標準、FIFO)の特徴、可視性タイムアウト設定による処理の重複防止は重要ですので、押さえておきましょう。

●Amazon Kinesis

Amazon Kinesisは、主にストリーミングデータの収集、処理、リアルタイム分析に利用されます。

サーバーレスで処理ボリュームに応じて拡張するため、AWS Auto ScalingとAmazon EC2を利用して自前で構築するよりも、簡単かつ低コストでストリームデータのバッファリング処理が可能になります。

なお、SQSとの主な違いは、Kinesisでは複数のコンシューマー(メッセージを取得する側)が同時に同じメッセージを取得して処理できる点にあります。

Kinesis は、主にAmazon Kinesis Data Streams、Amazon Kinesis Data Firehose、Amazon Kinesis Data Analyticsの3サービスで構成されます。

● Amazon Kinesis Data Streams

Kinesis Data Streamsは、ストリーミングデータをほぼリアルタイムで収集することができ、収集されたデータは、Amazon EMRやAWS Lambdaなどのサービス上に構築された独自アプリケーションと連携処理することができます。

また、流れてくる大容量のデータを効率的に処理するために、シャードと呼ばれる単位でデータを分割して並列処理することができます。

● Amazon Kinesis Data Firehose

Kinesis Data Firehoseは、独自にアプリケーションを構築することなく、ストリーミングデータをAWSの各サービス（Amazon S3やAmazon Redshift、Amazon OpenSearch Serviceなど)に簡単に配信・保存できるサービスです。

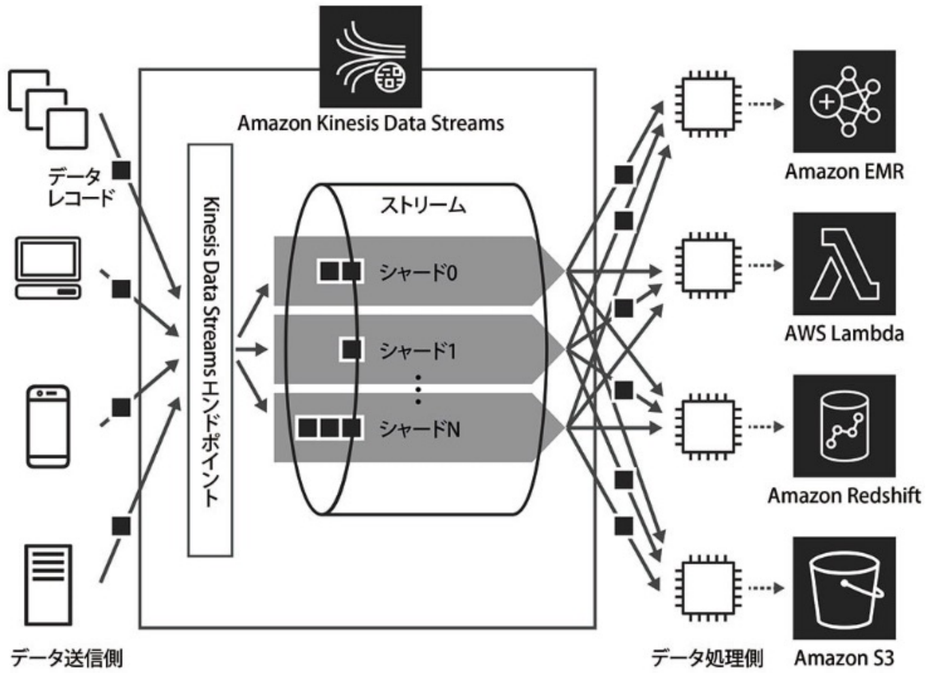
たとえば、ストリーミングされるデータを分析用データとしてS3やRedshiftに蓄積するケースなどで利用されます。

● Amazon Kinesis Data Analytics

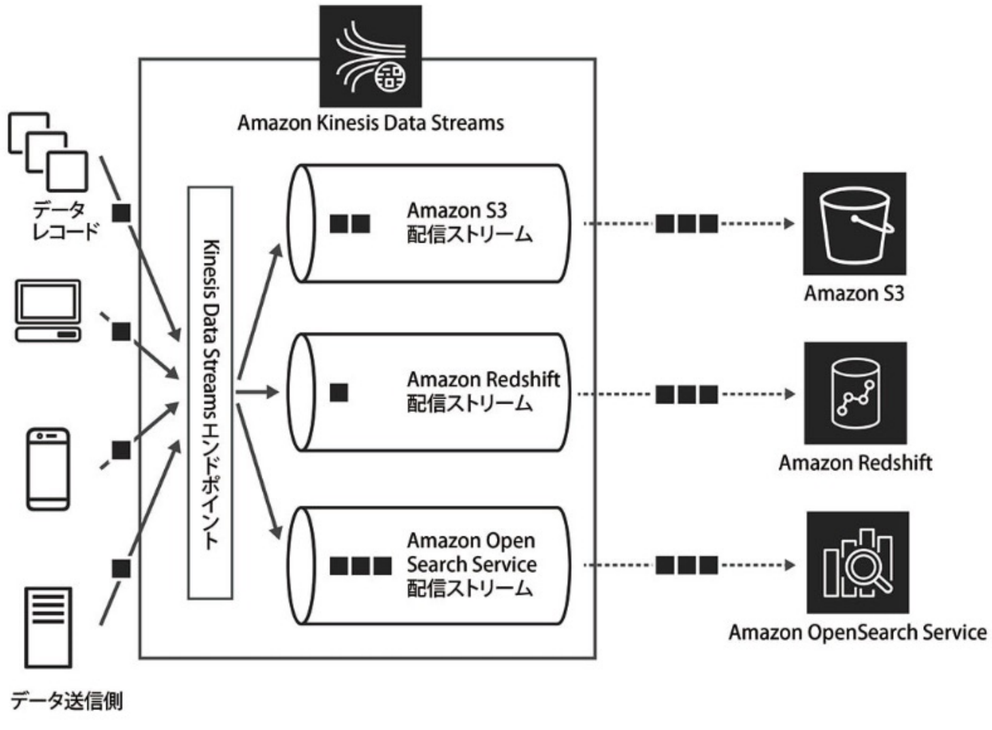
Kinesis Data Analyticsは、ストリーミングデータに対してSQLクエリを実行し、リアルタイム分析を行うサービスです。

SQLクエリを利用できるため、たとえば、1分ごとのストリーミングデータの合計値や平均値などを簡単に計算できます。

【Kinesis Data Streamsのイメージ】



【Kinesis Data Firehoseのイメージ】



試験対策

Kinesisの3つのサービスの違いを押さえておきましょう。それぞれ、ストリーミングデータの収集、保存、分析と役割の違いを理解すると覚えやすくなります。

Q 演習問題

1 Amazon SQSの用途として適切な項目は、次のうちどれですか。

A　アプリケーションから同期処理でデータを格納するデータベースインフラストラクチャ

B　アップロードされたメディアファイルを複数のEC2インスタンスで非同期処理する場合のジョブ制御

C　Webアプリケーションのログイン セッションの保存

D　リアルタイムでのストリーミングデータ処理

2 Amazon SQSの機能で、処理の重複を防ぐために利用可能な機能は次のうちどれですか。

A　ロングポーリング

B　可視性タイムアウト

C　メッセージキュー(MQ)

D　バッファリング

3 ある企業は、ストリーミングデータの処理をAWSで行いたいと考えています。複数の独自アプリケーションから同時に同じストリーミングデータを読み込み、処理する必要がある場合に利用するサービスとして、適切なサービスは次のうちどれですか。

A　Amazon SQS

B　Amazon Kinesis Data Analytics

C　Amazon Kinesis Data Streams

D　Amazon Kinesis Data Firehose

A 解答

1 B

SQSはメッセージキュー(MQ)により、非同期なアプリケーション処理のインフラストラクチャを提供します。

2 B

SQSの可視性タイムアウト(Visibility Timeout)により、ある受信者がメッセージを取得した場合に、ほかの受信者にはそのメッセージをある一定時間(デフォルトは30秒間)見せないようにすることで、処理の重複を防ぐことができます。

3 C

Amazon Kinesis Data Streamsは、ストリーミングデータをほぼリアルタイムで保存することができ、登録されたデータはAmazon EMRやAWS Lambdaなどのサービス上に構築された独自アプリケーションで処理することができます。また、流れてくる大容量のデータを効率的に処理するために、「シャード」と呼ばれる単位でデータを分割して並列処理を行うことができます。

5-4 コストの管理

ここまでの節で、AWSにおけるコスト最適化に必要なサービスや手法を説明しました。
本節では、継続的にコスト最適化を実施するうえで必要なコスト管理方法や可視化、レポーティングについて説明します。

1 AWSにおけるコスト管理

AWSでは、アカウントごとにサービスを利用した分のコストが発生します。

通常、コストの支払い管理は、アカウントの管理者(たとえば、IT部門など)が行います。しかし、どの程度利用しているかといった利用率の詳細把握や、決められた予算を超過していないかなどの予算実績管理、利用部門へのコスト配賦など、コスト管理で行うべきことは多岐にわたります。

たとえば、次の表に示すタスクに対して、それぞれに対応するAWSサービスを駆使することで、コストを把握・分析・最適化することができます。

【主なコスト管理のタスク】

タスク	タスク概要	主なAWSサービス
コストの請求・支払い	複数のAWSアカウントをまとめて支払い管理する	Consolidated Billing(一括請求)
コスト状況の詳細把握	コストとリソース使用量の詳細をレポート・分析する	AWS Cost and Usage Report(コストと使用状況レポート)
コストの可視化・傾向分析	コストの可視化や深掘り分析、将来予測を行う	AWS Cost Explorer
過剰利用の監視	予算策定や予算超過のアラートを管理する	AWS Budgets(予算設定)
コスト最適化の検討	コスト最適化の余地があるリソースを分析・改善する	AWS Trusted Advisor

前表で説明した代表的なコスト・請求に関わるサービスには、AWS請求コンソールとダッシュボード(Billing and Cost Management Dashboard)からアクセスできます。

次に、各AWSサービスの概要を説明します。

2 Consolidated Billingによる支払い管理

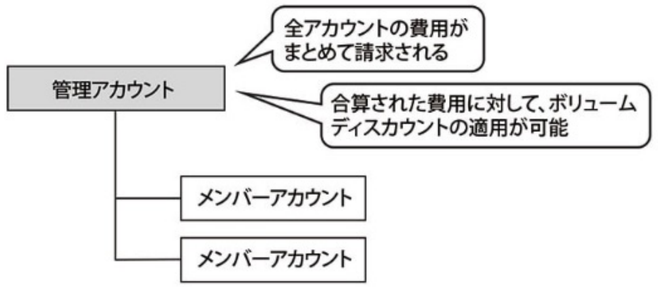
Consolidated Billing(一括請求)は、複数のAWSアカウントの請求を1つに統合し、まとめて支払いができる機能です。これは、複数のAWSアカウントを一元管理する**AWS Organizations**と呼ばれるサービスの一機能として提供されています。

この機能を利用することでユーザーは、次に示すような利点が得られます。

- 複数のAWSアカウントのコストが1つの請求書に統合されることによる支払い業務の効率化
- 各AWSアカウント(たとえば、業務部門やIT部門、開発環境や本番環境など)の使用状況を統合的に把握可能
- 複数のAWSアカウントの使用量を統合することで、ボリュームディスカウントによるコスト削減が可能

次の図に示すように、請求先の**管理アカウント**(Management Account)に対し、**メンバーアカウント**(Linked Account)の請求が統合できます。

【Consolidated Billing(一括請求)のイメージ】



試験対策

通常、請求先の管理アカウントは支払い専用のAWSアカウントとして作成します。メンバーアカウントは、開発環境やテスト環境などの環境ごとやAWSを利用する部門ごとなど、利用用途に応じて作成されるのが一般的です。

3　AWS Cost and Usage Reportによる詳細レポート・分析

AWS Cost and Usage Report(コストと使用状況レポート)では、AWSのサービスごとのリソース使用状況とコストを時間単位や日次単位で収集し、CSV形式でAmazon S3に保存またはAmazon Redshiftに格納したり、**Amazon QuickSight**で分析・可視化できます。

Consolidated Billingを利用している場合は、管理アカウントのレポートにメンバーアカウントの情報が表示されます。たとえば、このレポートで取得可能な情報には次のような情報があります。

- bill/PayerAccountId：支払いアカウントのAWSアカウントID
  (Consolidated Billingの場合は管理アカウントのID)
- product/ProductName：利用しているAWSサービスの名前
- product/region：AWSサービスのリージョン
- lineItem/UsageAmount：指定した期間に発生したリソース使用量

なお、このレポートでは、オンデマンドインスタンスだけでなくリザーブドインスタンスの利用状況や料金などの情報も取得できるため、定期的に監視・分析することで、リザーブドインスタンスの見直しなど、コスト最適化のための計画が検討できます。

参考

以前はDetailed Billing Report(請求明細レポート)が利用されていましたが、現在は推奨されていません。

4　Cost Explorerによるコストの可視化と傾向分析

AWS Cost Explorerは、前項の表で説明したAWS Cost and Usage Reportのデータをグラフィカルに可視化したり検索できるサービスです。

たとえば、Amazon EC2のリソース使用量やコストを時系列にグラフ化し、詳細に分析したい場合は、さまざまな条件(サービス別やタグ指定など)や期間を指定してフィルタリング・検索することができます。また、カスタムレポートの作成も可能で、AWSのマネジメントコンソール上でコストをインタラクティブに分析できます。

具体的には、「可視化」「分析」「予測」の3つの機能が提供されています。

●可視化

可視化機能では、最長で過去12カ月分のコストデータをグラフ表示できます。あらかじめ次のような可視化グラフが用意されており、簡単に切り替えて表示できます。

- Monthly costs by service：サービスごとの月次コスト推移
- Monthly costs by linked account：AWSアカウントごとの月次コスト推移
- Daily costs：日次コスト推移
- Monthly EC2 running costs and usage：月次のEC2使用時間・コスト推移
- RI Utilization：リザーブドインスタンスのコスト推移やオンデマンドインスタンスと比較した場合のコスト削減額など

●分析

分析機能では、サービスやAWSアカウント、リージョン、アベイラビリティーゾーン(AZ：Availability Zone)、インスタンスタイプ、タグなどの条件でフィルタリングしてコスト表示できます。カスタムレポートの作成も可能です。

●予測

予測機能は、可視化や分析だけでなく、過去の使用量のトレンドから今後3カ月間のコストを予測することができます。

試験対策

AWS Cost Explorerの特徴と機能を押さえておきましょう。
Cost Explorerを使用すると、サービスごとの詳細なコスト分析やレポート作成ができます。なお、簡易分析であれば、Billing and Cost Management Dashboard上で、過去2カ月のインスタンスタイプ別のEC2コストなどをグラフ化して参照できます。

実際の現場では、AWSのリソース使用量に応じて部門ごとにコストを配賦したいケースがあります。Cost Explorerで部門を示すタグを定義しておくと、タグでフィルタリングすることで部門ごとのコストを集計できます。

5　予算設定による利用超過の監視

AWSでは、Billing and Cost Management DashboardのBudgetsから予算を作成することができます。

予算額やリソース使用量をあらかじめ設定し、現在どのくらい消費しているかを確認したり、超過時の通知方法を設定できます。具体的には、次の項目で設定できます。

●予算額の設定

コストやリソース使用量などの監視対象を選び、月次や四半期などのタイミングを指定して監視する予算額や使用量を設定します。

●通知の設定

設定した予算額を超過しそうな場合、または超過した場合にAmazon CloudWatchアラームやAmazon SNSトピック、メールなどによる通知が可能です。

超過の条件は予算額に対する比率(%)や実際の消費量(金額)より大きい、小さい、等しいなど、さまざまな設定が可能です。

6　Trusted Advisorによるコスト最適化

ここまで、コストの詳細把握・可視化と分析・予測・監視などに利用できるAWSサービスを説明しました。

コスト管理に関連する最後のタスクはコスト最適化です。AWSでは、コスト最適化のヒントを推奨するサービスとして、AWS Trusted Advisorが提供されています。

Trusted Advisorは、AWSのベストプラクティスに基づいて、コスト最適化、セキュリティ、耐障害性、パフォーマンス、サービスの制限の5項目でユーザーのAWS利用状況をチェックし、改善すべき事項を推奨するサービスです。

Trusted Advisorのコスト最適化機能では、たとえば、以下の観点からコストを診断できます。

●使用率の低いEC2インスタンス

少なくとも過去14日間、常に稼働しているAmazon EC2インスタンスがあり、1日のCPU使用率が10%以下かつネットワークI/Oが5MB以下である日が4日以上ある場合に、アラートを発報します。

使用率が低いEC2インスタンスに対して、インスタンス数やインスタンスサイズを調整することでコスト削減が可能です。

●EC2リザーブドインスタンスの最適化

前月の1時間ごとのEC2使用量に基づいて、リザーブドインスタンスの最適数を推奨します。

推奨されたリザーブドインスタンス数を購入することで、少なくともEC2使用料金の10%が節約できます。

●関連付けられていないElastic IPアドレス(EIP)

稼働中のEC2インスタンスに関連付けられていないAWS Elastic IP(EIP)をチェックします。

EIPに課金が発生するため、EC2インスタンスへ割り当てられていないEIPや、割り当てられていてもEC2インスタンスが停止中の場合には、使用していないEIPを解放することでコストの削減が可能です。

これ以外にもELBやAmazon EBS、Amazon RDSなどのサービスについても、使用状況などからコスト最適化の診断を行うことができます。

> 参考
> AWS Trusted Advisorのコスト最適化機能を利用するには、ビジネスまたはエンタープライズレベルのAWSサポートプランを選択している必要があります。

Q 演習問題

1 **ある企業では、AWSサービス別のコストを可視化して分析したいと考えています。最も適切なサービスは、次のうちどれですか。**

A　Consolidated Billing

B　Amazon Athena

C　AWS Cost Explorer

D　AWS Budgets

A 解答

1 C

AWS Cost Explorerにより、AWSで発生するコストの詳細な可視化や分析が可能です。

Column　クラウド利用で非機能要件定義は不要になるのか？

ここまで本書で学習を進めてきた方は、すでにAWSを利用するメリットを十分に実感できていると思います。

クラウドを利用するメリットの1つとして『サーバーのスペックを容易に短時間で変更できるようになった』ことがあげられます。

オンプレミス環境でシステム基盤を準備するには、事前に「システムが今後どの程度利用されるか」や「ピーク時の最大負荷」などを十分検討したうえで、システムのスペックを検討する必要がありました。

というのも、オンプレミス環境のシステムを本番環境で運用し始めてから処理性能が不足した場合、その対応には大きな手間とコストがかかることがあるためです。それがクラウドを利用すると、画面上の操作だけでサーバーやネットワークが準備でき、容易に構築、変更、削除できるわけですから、そのメリットはかなり大きいでしょう。

では、クラウドを利用するようになると、非機能要件定義が不要になるのでしょうか？　結論からいいますと、非機能要件定義はなくならず、システム構築前の重要な作業として残り続けます。

クラウド環境ではサーバーやデータベースなどのインスタンスタイプが簡単に変更できるため、事前の設計は楽になりました。ただし、そのシステムに要求される可用性やセキュリティなど、要件によって利用するサービスやインスタンスなどが変わったり、利用料にも違いがあります。

そのため、システムの設計・構築前に要件定義（非機能要件）を実施した方がよいことに変わりはありません。『システム構築する際の要件定義』について詳しく知りたい場合は、情報処理推進機構（IPA）のサイトなどを参照してみてください。

索引

記号・数字

.NET 63
$LATESTバージョン 279
1ゾーン 81
1ゾーン一低頻度アクセス 74, 81, 313
6つの柱 13
99.999999999% 312

A

ACID 91
Active Directory 167
AD Connector 167
AdministratorAccess 160
AES-256 74
ALIASレコード 32
Amazon API Gateway 60
Amazon AppFlow 124
Amazon AppSync 68
Amazon Athena 112
Amazon Aurora 92, 97, 100
Amazon Aurora Global Database 102
Amazon CloudFront 30, 223, 266
Amazon CloudWatch 138
Amazon CloudWatch Logs 197
Amazon Cognito 167
Amazon Comprehend 126
Amazon DocumentDB 115
Amazon DynamoDB 103, 256, 289
Amazon DynamoDB Accelerator (DAX) 289
Amazon DynamoDB Auto Scaling 290
Amazon DynamoDBグローバルテーブル 290
Amazon EBS-Backed 46
Amazon Elastic Block Store (EBS) 77, 243, 283
Amazon Elastic Compute Cloud (EC2) 43, 221
Amazon Elastic Container Registry (ECR) 57
Amazon Elastic Container Service (ECS) 53
Amazon Elastic Container Service Anywhere (ECS Anywhere) 55
Amazon Elastic File System (EFS) 80, 243
Amazon Elastic Kubernetes Service (EKS) 55
Amazon Elastic Kubernetes Service Anywhere (EKS Anywhere) 56
Amazon Elastic Kubernetes Service Distro (EKS Distro) 56
Amazon Elastic Transcoder 68
Amazon ElastiCache 110, 232, 258, 293
Amazon Elasticsearch Service 125
Amazon EMR 125
Amazon EventBridge 124
Amazon Forecast 126
Amazon Fraud Detector 126
Amazon FSx 81, 244
Amazon GuardDuty 197
Amazon Inspector 198
Amazon Keyspaces 115
Amazon Kendra 126
Amazon Kinesis 123, 322
Amazon Kinesis Data Analytics 322
Amazon Kinesis Data Firehose 322
Amazon Kinesis Data Streams 322
Amazon Lex 126
Amazon Lightsail 66
Amazon Linux 46
Amazon Macie 199
Amazon Managed Streaming for Apache Kafka (MSK) 124
Amazon Machine Image (AMI) 43, 45, 250
Amazon MQ 120
Amazon Neptune 111
Amazon OpenSearch Service 125
Amazon Outposts 86
Amazon Pinpoint 67
Amazon Polly 126
Amazon Quantum Ledger Database (QLDB) 116
Amazon QuickSight 113, 125, 328
Amazon Redshift 107, 291
Amazon Redshift concurrency scaling 292

Amazon Rekognition 126
Amazon Relational Database Service (RDS) 91, 254, 288
Amazon RDS for Oracle 189
Amazon Route 53 ..... 32, 214, 271
Amazon SageMaker 126
Amazon SageMaker Savings Plans 308
Amazon S3 Glacier 75, 244
Amazon S3 Transfer Acceleration 76
Amazon Simple Email Service (SES) 120
Amazon Simple Notification Service (SNS) 120
Amazon Simple Queue Service (SQS) 119, 209, 319
Amazon Simple Storage Service (S3) 71, 244
Amazon Textract 126
Amazon Timestream 115
Amazon Transcribe 126
Amazon Translate 126
Amazon Virtual Private Cloud (VPC) 18, 212
Amplify CLI 61
Amplify Developer Tools 61
Amplify Framework 61
Amplify Studio 62
Apache ActiveMQ 119
Apache Cassandra 115
Apache Hive 125
Apache HTTP Server 63
Apache Spark 125
APIエンドポイント 60
APIエンドポイントタイプ 61
APIコール 27
Application Auto Scaling 50
Application Load Balancer (ALB) 28
Atomicity (原子性) 91
Aurora Serverless 102
Auroraマルチマスタークラスター 101
Auroraレプリカ 100
Auto Scaling 50, 221, 278
Auto Scalingグループ 51
AWS Agentless Discovery Connector 64
AWS Amplify 61
AWS Application Discovery Agent 64
AWS Application Discovery Service 65
AWS Application Migration Service (AWS MGN) 65
AWS Artifact 199
AWS Artifact Agreements 199
AWS Artifact Report 199
AWS AuditManager 199
AWS Auto Scaling 50, 227, 319
AWS Backup 86, 243
AWS Batch 66
AWS Certificate Manager (ACM) 29, 30, 185
AWS CLI 164
AWS Client VPN 25
AWS CloudFormation 131
AWS CloudHSM 188
AWS CloudTrail 196
AWS CodeBuild 135
AWS CodeCommit 135
AWS CodeDeploy 135
AWS CodePipeline 135
AWS Command Line Interface (CLI) 40
AWS Compute Optimizer 68
AWS Config 144, 198
AWS Control Tower 132
AWS Cost and Usage Report 328
AWS Cost Explorer 329
AWS Database Migration Service (AWS DMS) 64, 113
AWS Data Exchange 125
AWS Data Pipeline 122
AWS DataSync 35, 114, 250
AWS Device Farm 67
AWS Direct Connect 25, 213, 215
AWS Directory Service 35, 167
AWS Elastic Beanstalk 62
AWS Fargate 56
AWS Firewall Manager 181
AWS Global Accelerator 34
AWS Glue 122
AWS Identity and Access Management (IAM) 38, 152
AWS Key Management Service (KMS) 187
AWS Lake Formation 125

AWS Lambda....57, 224, 278
AWS License Manager....133
AWS Managed Grafana....144
AWS Managed Service for Prometheus....144
AWS Marketplace....46
AWS Migration Hub....64
AWS Network Firewall....35
AWS Nitro System....49
AWS OpsWorks....135
AWS Organizations....168, 327
AWS Outposts....18
AWS PrivateLink....27
AWS Proton....135
AWS Resource Access Manager (RAM)....34
AWS SDK....41, 164
AWS Schema Conversion Tool (AWS SCT)....113
AWS Security Hub....200
AWS Security Token Service (STS)....166
AWS Serverless Application Model (SAM)....66
AWS Serverless Application Repository....66
AWS Server Migration Service (AWS SMS)....64
AWS Service Catalog....133
AWS Shield....180
AWS Site-to-Site VPN....25, 215
AWS Snowball....85, 251
AWS Snowcone....85
AWS Snow Family....84
AWS Snowmobile....85
AWS Step Functions....123
AWS Storage Gateway (SGW)....83
AWS Systems Manager....49, 144
AWS Transfer Family....85
AWS Transfer for FTP....86
AWS Transfer for FTPS....86
AWS Transfer for SFTP....86
AWS Transit Gateway....27
AWS Trusted Advisor....144, 198, 331
AWS Virtual Private Network (VPN)....25
AWS Web Application Firewall (WAF)....180
AWS Well-Architectedフレームワーク....12, 152, 157, 301
AWS X-Ray....134
AWSアカウント....38, 155
AWSグローバルインフラストラクチャ....151
AWS請求コンソール....327
AWS脆弱性／侵入テストリクエストフォーム....200
AWSマネージドサービス....17
AWSマネジメントコンソール....38, 39
AWS管理ポリシー....160
AWS責任共有モデル....150
AWSリソースのイベント....141
Aレコード....32, 235

B

BI (Business Intelligence)....107
bill/PayerAccountid....328
Billing and Cost Management Dashboard....327
Budgets....330

C

CDN....17, 30, 266
CIS (Center for Internet Security)....199
Classic Load Balancer (CLB)....28
Client Side Encryption (CSE)....186
CIDR....19
CloudTrail....143
CloudWatch Events....141
CloudWatch Logs....142
CloudWatchアラーム....140
CNAMEレコード....32
Compute Saving Plans....308
Connection Draining....29
Consistency (一貫性)....91
Consolidated Billing....327
Content Delivery Network....17, 30, 266
Convertibleタイプ....307
CPU使用率....140
CSE....191
cURLコマンド....49

D

DDoS (Distributed Denial of Service)....180
Design for Failure....207
Direct Connectゲートウェイ....25
DNS (Domain Name System)....271
DNSサービス....271
DNSフェイルオーバー....235
Docker....53, 63
DoS (Denial of Service)....180
Durability (耐久性)....91

E

EBS-Backedインスタンス....245
EBS最適化インスタンス....276
EC2 Auto Scaling....50
EC2 Instance Savings Plans....308
EC2インスタンス....43
Edge Compute Optimized....85
Edge Storage Optimized....85
EKSコンソール....56
Elastic IP (EIP)....34
Elastic Load Balancing (ELB)....28, 172, 224
Elastic Network Interface (ENI)....34, 279
Elasticity (伸縮性)....318
EventBridge....142
ExecutionRole....59

F

FIFO (先入れ先出し) キュー....320
FSxファイルゲートウェイ....84
FSx for Lustre....82
FSx for Windows File Server....82

G

Gateway Load Balancer (GLB)....28
GDPR....199
Glacier....312
Glacier Deep Archive....75, 312
Glacier Flexible Retrieval....75
Glacier Instant Retrieval....75, 312
Go....63
gp2....78, 284, 313
gp3....78, 284, 313
Grafana....144
GraphQL....68

H

Hadoop....275
HDD (Hard Disk Drive)....283
HSM (Hardware Security Module)....188

I

IAM....38
IAMグループ....38, 157
IAMポリシー....39, 157
IAMユーザー....38, 156
IAMユーザー名....163
IAMロール....164
IDフェデレーション....166
Infrastructure as Code (IaC)....131
Instance Store-Backed....46
Instance Store-Backedインスタンス....245
Intelligent-Tiering....75, 313
Internal....30
Internet Gateway (IGW)....21
Internet-facing....30
io1....78, 284, 314
io2....78, 284, 314
IOPS (I/O Operations Per Second)....283
IoT....289
IPスプーフィング....152
iSCSI....83
Isolation (独立性)....91

J

Java....63
JSON....73, 115, 158

K

Kinesis Data Analytics....123
Kinesis Data Firehose....123
Kinesis Data Streams....123
Kinesis Video Streams....124
Kubernates....55

L

Lambda@Edge....59, 270
Lambda関数....57, 270
LDAP....166
lineItem/UsageAmount....328
Linked Account....327

M

Management Account 327
MariaDB 92, 97
Memcached 232, 293
MFA 155
Microsoft Active Directory (AD) 35, 167
Microsoft Internet Information Services (IIS) 63
Microsoft SQL Server 92, 97
Microsoft Windows Server 46
Migration Evaluator Collector 64
Migration Hubインポート 64
MongoDB 115
Multi-Factor Authentication 155
MySQL 92, 97

N

NAT 22
NATインスタンス 23
NATゲートウェイ 22, 172, 213
Network Address Translation 22
Network Load Balancer (NLB) 28
NFS 80, 247
Nginx 63
Node.js 63
NoSQL 103
NoSQLデータベース 256

O

OpenID Connect 166
Oracle Database 92, 97
OSイメージ 43

P

Passenger 63
Payment Card Industry (PCI) 199
PCI DSS 199
PHP 63
PostgreSQL 92, 97
PowerUserAccess 160
Presto 125
product/region 328
Prometheus 144
Provisioned Concurrency 279
Pub/Sub 121
Puma 63
Python 63

R

RabbitMQ 119
Receive Messageリクエスト 320
Red Hat Enterprise Linux 46
Redis 232, 293
Redshift Spectrum 109
RDP (Remote Desktop Protocol) 178
Ruby 63

S

S3ファイルゲートウェイ 84
SAML 166
Savings Plans 308
sc1 ..... 78, 284, 314
Security Assertion Markup Language (SAML) 166
Server Side Encryption (SSE) 186
Service Organization Control (SOC) 199
Session Manager 49
Simple AD 167
SLA 207
SMB 82, 247
Sorryページ 77
SPOF (Single Point of Failure) 207
SQLインジェクション 180
SSD (Solid State Drive) 283
SSE-C 191
SSE-KMS 191
SSE-S3 191
SSH (Secure Shell) 178
SSL接続 99
SSL暗号化通信 31
SSL証明書 31
SSL復号 29
SSL/TLS 184
SSL/TLS証明書 269
st1 ..... 78, 284, 314

Standardタイプ 307

T

Temporary Security Token 166
TTL (Time to LIVE) 機能 106

U・V

Ubuntu 46
Virtual Private Cloud (VPC) 171
Virtual Private Gateway (VGW) 26
Visibility Timeout 321
VM Import/Export 63
VMware Cloud on AWS 67
VPC 171
VPCエンドポイント 27
VPCピアリング 26
VPCフローログ 144, 196
vSphere 53

W

Web IDフェデレーション 167
Webサイトホスティング 77

X・Y・Z

XML 115
Zone Apexレコード 32, 235

ア行

アイデンティティ管理 152
アウトバウンド 23, 173
アクセスキーID 40, 163
アクセス管理 152
アクセス制御 73
アドテクノロジー 289
アベイラビリティーゾーン 16, 71
アベイラビリティーゾーンサービス 18
暗号化 73, 98
暗示的な拒否 160
位置情報ルーティング 33
一部前払い 306
イベント 141
イベントソース 141
イベント駆動型 58
イレブンナイン 312
インスタンス構成 307
インスタンスサイズ 311
インスタンスストア 83, 243
インスタンス世代 311
インスタンスタイプ 44, 310
インスタンスファミリー 44, 310
インスタンスメタデータ 48
インターネットゲートウェイ 21
インターフェイス型 27
インバウンド 23, 173
インフラストラクチャ構成管理 130
インメモリキャッシュ 289
インメモリデータベース 258
インラインポリシー 161
運用上の優秀性 13
エージェントベース検出 65
エージェントレス検出 65
エッジ 111
エッジ最適化APIエンドポイント 61
エッジロケーション 17, 266
エフェメラルディスク 83, 243
エンドポイント 29
オブジェクトストレージ 71
オブジェクトロック機能 72
オリジン 30
オリジンアクセスアイデンティティ (OAI) . 270
オンデマンドインスタンス 306
オンデマンドバックアップ 257

カ行

鍵の管理 187
可視化 329
可視性タイムアウト 321
加重ルーティング 33
カスタマー管理ポリシー 161
カスタマーマスターキー (CMK) 185
カスタムエラーページ 31
カスタムメトリクス 139
仮想プライベートゲートウェイ 26
稼働率 206
可用性 206, 289
環境変数 58

管理アカウント 327
管理イベント 144
キーバリュー型 104, 293
キーペア 45
基本モニタリング 139
キャッシュ型ボリュームゲートウェイ 84
キャパシティモード 106
行指向データベース 291
強力な整合性 105
クールダウン 222
クラスター 54, 107
クラスターコンピューティング 274
クラスタープレイスメントグループ 274
クラスターボリューム 100
グラフデータベース 111
グローバルサービス 18
クロスアカウントデータ共有 108
クロスサイトスクリプティング 180
クロスゾーン負荷分散 29
クロスリージョンスナップショット 108
クロスリージョンリードレプリカ 98, 254
クロスリージョンレプリケーション 72
ゲートウェイ型 27
結果整合性モデル 105
権限の優先度 160
公開鍵 45
公開鍵認証方式 45
高可用性 28, 80
高可用ネットワーク 215
構成管理 129
構造化データ 103
購読者（サブスクライバ） 120
購入オプション 305
コールドHDD 78, 284, 314
個人識別情報（PII） 199
コスト管理 326
コスト最適化 14, 74, 82, 300, 318
コンテナ 53
コンテナオーケストレーション 53
コンピュータノード 107
コンピューティング最適化 311

サ行

サーバーレスコンピューティングエンジン 56
サーバーレスアーキテクチャ 225, 263
サーバーレスアプリケーション 66
サービスコントロールポリシー（SCP） 168
災害復旧（DR） 16
最小権限の原則 152, 157
サブスクライバ 120
サブネット 19
シークレットアクセスキー 40, 163
持続可能性 14
自動スケーリング 29, 80
シャード 322
詳細モニタリング 139
冗長化 206
ショートポーリング 320
従量課金制 277
署名付きURL 31, 77
需要と供給のマッチング 318
シングルサインオン 166
シンプルスケーリング 51
シンプルルーティング 33
信頼性 13
スケーリングプラン 51
スケールアウト 221
スケールイン 222
スケジュール 141
スケジュールされたリザーブドインスタンス 307
スタック 132
スタンバイデータベース 92
スティッキーセッション 29
ステートフル 23, 173
ステートレス 24, 174
ステップスケーリング 51
ストリーミング処理 123
ストリーミングデータ 292
ストリーミング配信 31
ストレージクラス 74, 80, 312
スナップショット 79, 107, 248
スプレッドプレイスメントグループ 275
スポットインスタンス 305, 309, 321
スポットフリート 309

スポットブロック 309
スループット最適化HDD 78, 284, 314
静的コンテンツ 268
セキュリティ 13
セキュリティ監視 153
セキュリティグループ 23, 44, 172
全額前払い 306
疎結合 209

タ行

ターゲット 141
ターゲットトラッキングスケーリング 51
台帳データベース 116
多要素認証 155, 163
単一障害点 207
地域制限 31
地理的近接性ルーティング 33
地理的ブロッキング 31
通信の暗号化 184
通知の設定 330
低冗長化ストレージ 75
データ暗号化 190
データイベント 144
データウェアハウス 291
データ転送サービス 114
データの保護 153
データベースインスタンス 93, 100
データベースサービスの比較 294
データベースの認証情報 189
テープゲートウェイ 84
デフォルト暗号化オプション 186
デプロイ 129
テンプレート 131
投資効果 302
動的コンテンツ 268
動的コンテンツのWebサイトホスティング 246
ドキュメントデータベース 115
トピック 120

ナ行

内部ELB 30
認証情報 163, 189
ネットワークアクセスコントロールリスト（ACL） 24
ネットワークインターフェイス 34
ネットワークACL 174
ネットワークセキュリティ 152
ネットワークレイテンシー 265
年間故障率 284
年間停止時間 206
ノード 107, 111

ハ行

バージョニング機能 72
バースト性 292
パーティション 275
パーティションプレイスメントグループ 274
パイプライン処理 122
バケット 71
バケットポリシー 73, 162
パスワード 163
バックアップ 93
バックエンドインスタンス 28
パフォーマンス効率 13
パブリックサブネット 20, 212
パブリックホストゾーン 32
パブリッシュ 121
半構造データ 103
汎用SSD 78, 284, 313
非構造データ 103
秘密鍵 45
秘密保持契約（NDA） 199
ビルド 130
標準（Standard） 74, 80, 312
標準キュー 320
標準ー低頻度アクセス 74, 81, 312
標準メトリクス 139
ファイル暗号化 185
ファイルシステム 80
フィールドレベル暗号化 269
フェイルオーバー 206
フェイルオーバールーティング 33
複数値回答ルーティング 33
踏み台サーバー 178
プライベートAPIエンドポイント 61

プライベートサブネット 20, 212
プライベートホストゾーン 32
プライマリ・スタンバイ構成 228
プライマリテーブル 290
プライマリデータベース 92
プライマリ／レプリカ構成 258
プレイスメントグループ 274
ブロックストレージ 77, 243
ブロックパブリックアクセス 73
プロパティ 111
プロビジョニング 129
プロビジョンドIOPS SSD 78, 314
分散型サービス拒否 152
分散ワークロード処理 275
分析 329
ヘルスチェック 29, 227
ポイントインタイムバックアップ 107
ポイントインタイムリカバリ 230, 257
保管型ボリュームゲートウェイ 84
ボリュームタイプ 283, 313

マ行

マイクロサービスアーキテクチャ 134
前払いなし 306
マグネティック 78, 284, 314
マネージドサービス 90
マルチAZ 95, 104
マルチAZ構成 16, 225
マルチパートアップロード 76
マルチリージョン 17
マルチリージョン構成 236
明示的な許可 160
明示的な拒否 160
メッセージキューイング 119
メッセージキュー（MQ） 319
メッセージの発行（パブリッシュ） 121
メトリクス 139
メンバーアカウント 327
モバイルゲーム 289
モノリシックアーキテクチャ 134

ヤ行

有効期限 106
ユーザーデータ 48
予算額の設定 330
予算設定 330
予測 329

ラ・ワ行

ライフサイクルフック 223
ライフサイクルポリシー 76
リージョン 16
リージョンAPIエンドポイント 61
リージョンサービス 18
リージョン別エッジキャッシュ 266
リーダーノード 107
リードレプリカ 97, 288
リザーブドインスタンス 306
リストア 93
リレーショナルデータベース 91, 288
ルーティングポリシー 32
ルートテーブル 21
ルートユーザー 38, 155
レイテンシーベースルーティング 33, 271
列指向データベース 291
レプリカテーブル 290
ロードバランシング 224
ロングポーリング 320
ワンタイムパスワード 163

［著者］

鳥谷部 昭寛 ／とりやべ あきひろ

クラウド等の先端技術を用いたデジタル戦略立案などのコンサルティング業務に従事。AWSソリューションアーキテクト–プロフェッショナルをはじめ、Microsoft MCSE、Google Cloud Architect Professionalなどのクラウド関連資格を複数保持。趣味はゴルフと筋トレ。

宮口 光平 ／みやぐち こうへい

AWSをメインとしたクラウドシステムの業務に携わり、近年はPMOとして活動している。AWSソリューションアーキテクト–プロフェッショナルをはじめ、AWSおよびGoogle Cloudのクラウド認定資格を複数保持。趣味はスイーツを食べること、体験型イベントへの参加、旅行。

半田 大樹 ／はんだ だいき

AWSを利用するクラウドシステムの設計構築に従事。主にデータ分析基盤の関連業務に携わる機会が多い。AWS、Google Cloud、Microsoft Azureの認定資格を複数保持し、AWSは2022年12月時点で公開されている全資格を取得している。趣味は今年飼い始めた猫を可愛がること。

STAFF

編集	株式会社ソキウス・ジャパン
制作	相馬喜代子
表紙デザイン	馬見塚意匠室 阿部 修（G-Co, Inc.）
デスク	千葉加奈子
編集長	玉巻秀雄