

クラウドコスト最適化の取組み

イノベーション統括部 すみや てつお
住谷 哲夫

現在、国内外の企業においてパブリッククラウドが非常に多く活用されているが、主要なパブリッククラウドの課金体系は従量制であり、仮想サーバなどのリソース管理が疎かだと無駄にリソースを起動させてしまい、想定外のコストが発生する。そのため、初期の設計段階からリソース管理や、コストの最適化について検討しておく必要がある。本稿では、ドコモがこれまで培ってきたクラウドコスト最適化のノウハウ・取組みについて解説する。

1. まえがき

近年、パブリッククラウドを利用したサービスを提供する企業が増えている。パブリッククラウドは従来のようなデータセンタでのシステム構築とは異なり、利用者が管理コンソールから数クリックするだけで簡単に仮想サーバなどのリソースを立ち上げることができるため、システム構築のスピードを圧倒的に速くすることができ、企業の競争力強化に貢献している。一方で、簡単に利用できる分、多くの企業はいまだにパブリッククラウドのコストの管理や抑制に苦慮している。主要なパブリッククラウドの課金体系は基本的に従量制であり、仮想サーバなどのリソース管理が疎かだと無駄にリソースを保持してしまい、想定外のコストが発生する。そのた

め、初期の設計段階からリソース管理や、コストの最適化について検討しておく必要がある。

コストの最適化には、まずコストの可視化を行い、コスト管理の効果を見えるようにし、コスト削減施策と効果検証を繰り返していくことが重要となる。コストを可視化したリソースは継続して利用状況を定期的にチェックし、状況に応じて、構成変更、利用している仮想サーバの能力（AWS：Amazon Web Services^{*1}ではEC2（Elastic Compute Cloud）^{*2}インスタンス^{*3}タイプなど）の見直し、料金プラン活用などの検討を行うことが必要である（図1）。

本稿では、クラウドコスト最適化の取組みについて、そのポイントとなる考え方を述べ、具体的なノウハウを解説する。

©2021 NTT DOCOMO, INC.

本誌掲載記事の無断転載を禁じます。

本誌に掲載されている社名、製品およびソフトウェア、サービスなどの名称は、各社の商標または登録商標。

^{*1} AWS：Amazon Web Services社が提供するクラウドコンピューティングサービス。

^{*2} EC2：AWSが提供するIaaSの1つ。仮想マシン（^{*15}参照）を提供するサービス。

^{*3} EC2インスタンス：AWSで提供される仮想サーバ。

2. コスト最適化

コストの最適化は以下の優先度で検討をする（図2）。

STEP①：構築したアーキテクチャの見直し

コスト削減に最も有効なのは、アーキテクチャ全体を見直し極力マネージドサービス（自分で構築するのではなく、すでにクラウド事業者が提供しているサービス）の活用を考慮することである。運用中のシステムの変更は難しいため、初期設計時やシステム更改時にしっかりとコスト観点での設計を行う必要がある。

STEP②：不要なリソースの削除，利用リソースの最適化

運用期間が長くなっているアカウントでは、使われなくなったリソースが保持され続けてい

るケースや、本来であれば必要のない過剰なリソースを利用しているケースが散見される。無駄が無いよう定期的なリソース見直しを運用に取り入れることで、コスト削減につなげることができる。

STEP③：各種料金プランの活用

主要なパブリッククラウド事業者は、リソース利用のコミット（定められた期間の利用量をあらかじめ宣言）をすることで利用料を削減することができる料金プランを用意している。例えば、AWSではRI（Reserved Instance）、SP（Saving Plans）といった、1年もしくは3年の利用をコミットした上で料金を下げることができる料金プランがある。Google Cloud Platform^{*4}では確定利用割引、Microsoft Azure^{*5}でも

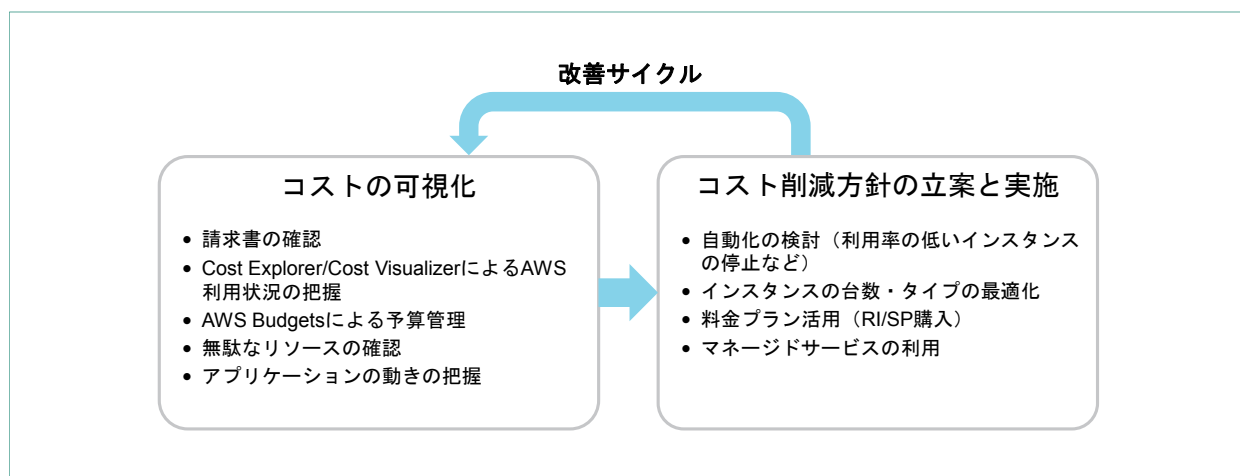


図1 コスト最適化の改善サイクル



図2 コスト最適化の検討プロセス

*4 Google Cloud Platform：Google社が提供するクラウドコンピューティングサービス。

*5 Microsoft Azure：Microsoft社が提供するクラウドコンピューティングサービス。

Reserved VM (Virtual Machine)^{*6} Instances という形で同様の料金プランがある。リソースを最適化した上でどうしても一定期間継続的に必要となる場合には、このような料金プランを活用することでコストを削減することができる。ただし、リソース利用のコミットをする場合は、コミット期間内のシステム構成の大幅な変更や、利用リソースの見直しの実施が難しくなるため、STEP①、②の検討を行った上で活用を考えるべきである。

3. コストの可視化

上記のポイントを考慮した上で、システムのどの部分にコストがかかっているのか、現状のコストを可視化して継続的にコスト構造を把握することが重要である。

主要なパブリッククラウド事業者はコストの可視化ツールを提供している。ここでは、AWSが提供するCost Explorerを解説する。また、Cost Explorerの提供開始前からドコモで内製開発し、全社的に利用しているCost Visualizerについても解説する。

3.1 Cost Explorerによる利用状況の把握

Cost ExplorerはAWS標準のツールで、請求金額をグラフで表示することができる(図3)。そのため、サービス別、メンバーアカウント^{*7}別など、さまざまな切り口でコストの細分化が可能である。また、デフォルトでRIやSPの利用率、カバー率など複数のレポートを出力することができる。

注意点として、Cost Explorerの利用に当たっては、必要最低限の権限のみ(例えばコストの閲覧権限のみ)を付与したIAM (Identity and Access Management) ユーザ^{*8}を作成し、アクセスするべきである。

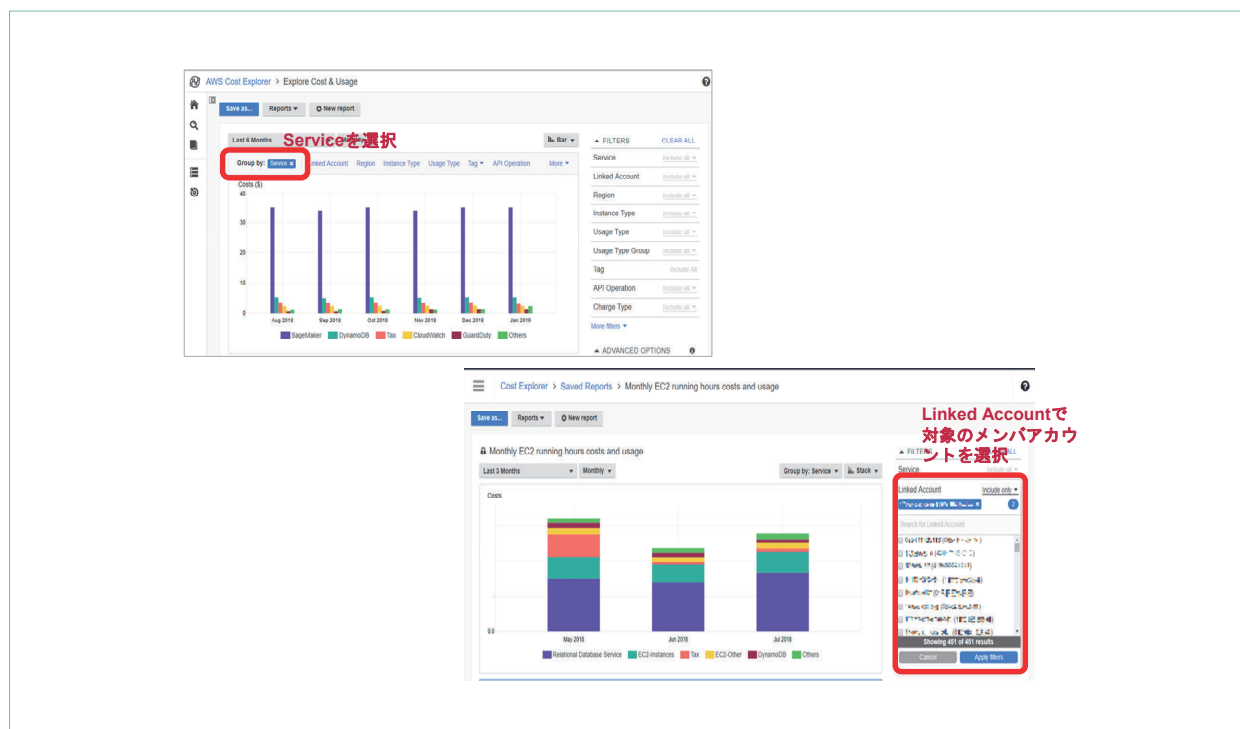


図3 Cost Explorer操作画面イメージ

^{*6} VM：仮想マシン（*15参照）。

^{*7} メンバアカウント：複数のAWSアカウントを統合する組織に所属する、管理アカウントでないアカウント。

^{*8} IAMユーザ：IAMサービスで作成した、AWS環境へのアクセス権をもつユーザ。

3.2 Cost Visualizerによる利用状況の把握

Cost Visualizerは、ドコモが開発・提供をしているコスト分析ツールである。ドコモが大規模にAWSを使い始めた2012年時点では前述のCost Explorerが提供されておらず、コスト管理上の必要性から本ツールが内製開発された。

Cost Visualizerは、AWSとは権限設定が分離しており、アカウント管理が別になっているため、AWSのサポートレベルや請求閲覧権限によらず利用することが可能で、Cost Explorerよりも細かい権限管理や、円グラフ表示、グルーピング表示など、Cost Explorerと異なる分析が可能である（図4）。

Cost Visualizerのシステム・アーキテクチャを図5に示す。AWSから提供されるコストと使用状況レポート（CUR：Cost and Usage Report）がAmazon S3（Simple Storage Service）^{*9}に自動的に格納され、そのデータをETL（Extract, Transform, Load）サービス（データベースにデータをロードして利用可能な状態にするサービス）であるAWS Glue^{*10}で抽出、変換を行い、Cost Visualizerが動作する仮想サーバ内のデータベースにロードする。マネージドサービスであるデータベースサービス（AWSではRDS：Relational Database Service）を使わずに取って仮想サーバ内にデータベースを起動させるのは、

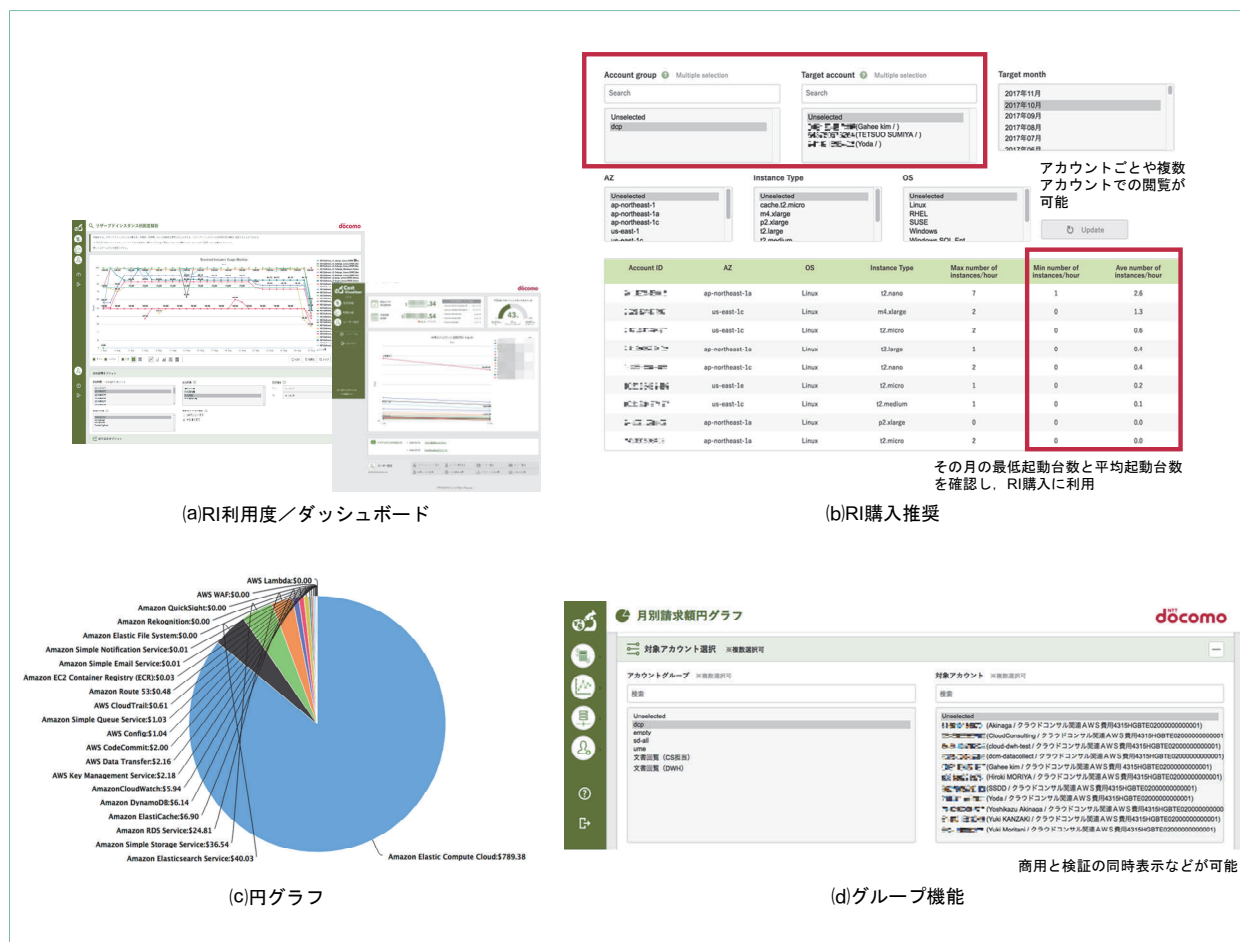


図4 Cost Visualizer操作画面イメージ

*9 Amazon S3：AWSが提供するストレージサービス。

*10 AWS Glue：AWSが提供するPaaSの1つ。データの分類、加工のための処理を実行できるサービス。

3.3 予算管理サービス

主要なパブリッククラウド事業者は予算管理のためのサービスをもち、コストまたは使用量が設定値を超過したり、超過する可能性を検知したりするとアラートをメールなどで通知している。AWS Budgetsは、コストだけではなくAWSリソースの利用量やRIの使用率なども設定することができ、コスト可視化ツールと合わせて利用することで普段のコスト意識を高めることができる。

4. コスト削減方針の立案と実施

4.1 マネージドサービスの利用

主要なパブリッククラウド事業者は、仮想サーバのようなコンピューティングリソースだけではなく、処理の特性に合わせたマネージドサービスを提供している。こういったマネージドサービスは、リソー

スを活用して処理を行った時間分の課金であるケースが多く、プロビジョニング^{*16}しているだけでコストがかかる仮想サーバでシステムを構築するよりもコストを削減することができる。

例えば、AWSであれば以下のような対応をすることで大幅なコスト削減を見込むことができる。

- ・長時間稼働させるサービスでリクエストが少ないものについては、EC2からLambdaへ置き換える。
- ・バッチ処理^{*17}を流したい場合はEC2ではなく、AWS Batch^{*18}の利用を検討する。AWS Batchを使用すると、バッチジョブのボリュームと必要リソースに応じて、コンピューティングリソースが動的にスケーリング^{*19}されるためコストを削減できる。
- ・図7のようなCognito, API Gateway, Lambda, DynamoDBなどによるサーバレスアーキテク

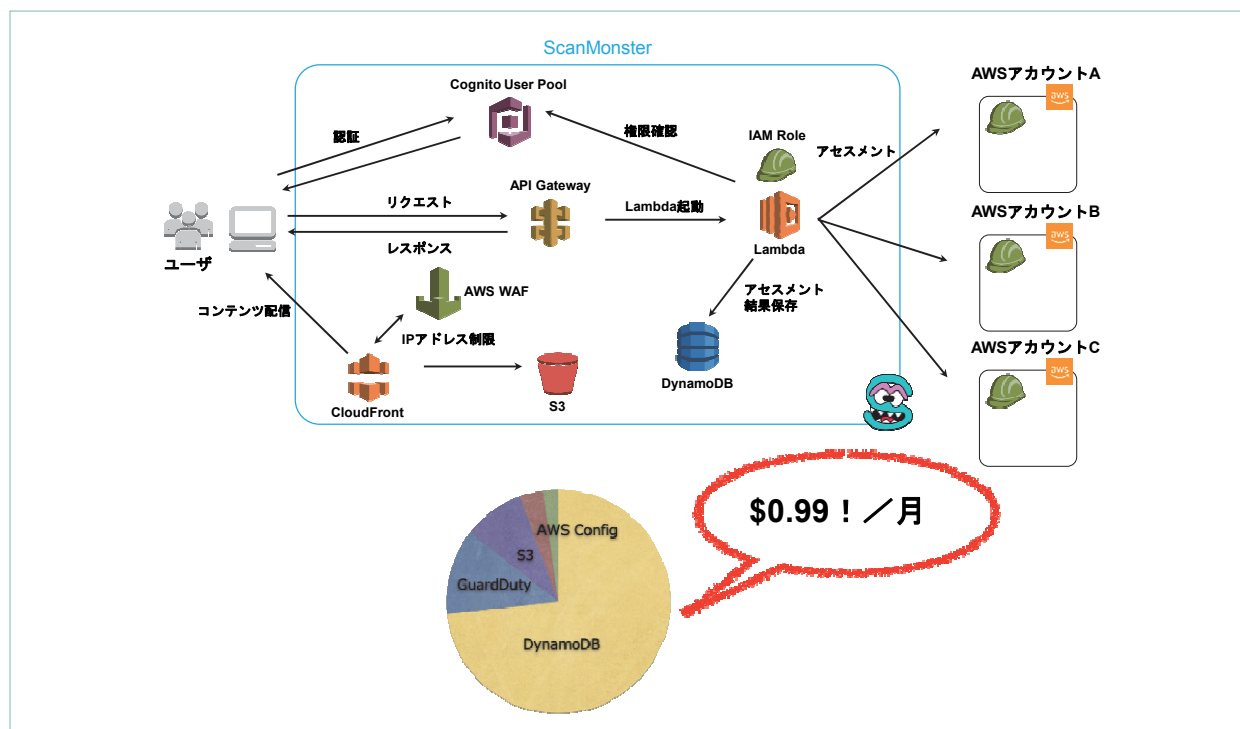


図7 サーバレスアーキテクチャの一例 (ScanMonster)

^{*16} プロビジョニング：アプリケーションを実行するために必要となるサーバやネットワークなどのリソースの確保やそれらを動作させるための各種設定作業。

^{*17} バッチ処理：一定量、一定期間のデータを集め一括処理をする処理方法。

^{*18} AWS Batch：AWSが提供するPaaSの1つ。大規模なバッチ処理を簡単かつ効率的に実行できるサービス。

^{*19} スケーリング：ハードウェアやVMの負荷状況に応じて処理能力が不足、あるいは余剰になった際に、VMを増減することにより処理能力を最適化すること。

チャに変更する。

4.2 無駄なリソースの確認

クラウドのリソースは一度プロビジョニングすると実際に利用をしなくてもコストが発生してしまうため、定期的に無駄なリソースを保持していないかチェックする必要がある。例えば、起動中のEC2インスタンスに関連付けられていないAmazon EBS (Elastic Block Store) ボリューム^{*20}、タグも付けられていない古いEBSスナップショット^{*21}などが

そういったリソースに該当する。

これらの無駄なリソースはクラウドサービスのコンソールから確認することができるが、多くのリソースを扱う場合、確認が難しくなってしまう。こういった無駄なリソースの確認の手段として、AWS Trusted Advisorなどを活用することが効率的である。Trusted Advisorでは、以下の項目を確認することができる(表1)。

実際にコスト削減に繋がった例として、あるプロジェクトでTrusted Advisorを利用してリソース状

表1 AWS Trusted Advisorで確認できるコスト最適化ポイント

| 確認項目 | 概要 |
|--------------------------------|---|
| 使用率の低いEC2インスタンス | CPU使用率、ネットワークI/Oの利用量から利用状況を判断 |
| アイドル状態のロードバランサー | ロードバランサーへのリクエスト数や、関連付けられているEC2インスタンスの数から利用状況を判断 |
| アイドル状態のRDS DBインスタンス | RDS DBインスタンスへの接続頻度によって利用状況を判断 |
| 利用頻度の低いAmazon EBSボリューム | EBSボリュームがEC2インスタンスにアタッチされていない、もしくは書き込み頻度によって利用状況を判断 |
| Route 53レイテンシリソースレコードセット | 非効率的に設定されたレイテンシーレコードセットを確認 |
| 使用率の低いRedshiftクラスタ | Redshiftへのクラスタ接続頻度、CPU使用率から利用状況を判断 |
| 関連付けられていないElastic IP Address | 実行中のEC2インスタンスに関連付けられていないElastic IP Addressを確認 |
| RI有効期限切れ | 前後30日の間に有効期限が切れたRIを確認 |
| Amazon EC2 RIの最適化 | EC2 RIの推奨購入数を表示 |
| Amazon RedShiftリザーブドノードの最適化 | Red Shift RIの推奨購入数を表示 |
| Amazon RDS RIの最適化 | RDS RIの推奨購入数を表示 |
| SPの推奨事項 | SPの推奨購入数を表示 |
| Amazon ElastiCacheリザーブドノードの最適化 | ElastiCache RIの推奨購入数を表示 |
| Amazon Elasticsearch RIの最適化 | Elasticsearch RIの推奨購入数を表示 |

Amazon ElastiCache：AWSが提供する、完全マネージドのインメモリデータストアサービス。

Amazon Elasticsearch：AWSが提供する、オープンソースである検索エンジンElasticsearchのマネージド・サービス。

Elastic IP Address：AWSが提供する、固定IPアドレスのサービス。

Redshiftクラスタ：AWSが提供するデータウェアハウスサービスのクラスタ。

Route 53レイテンシリソースレコードセット：AWSが提供するドメインネームサービスであるRoute 53に登録できる。エンドユーザからのレイテンシを最小にできるドメインとEC2インスタンスなどの組合せ。

ロードバランサー：サーバにかかる負荷を平等に割り振るための装置。AWSはロードバランサーをサービスとして提供している。

^{*20} Amazon EBSボリューム：AWSが提供する高性能で可用性に優れたブロックストレージのサービス。ブロックストレージとは、記録領域をボリュームという単位に分割し、ボリュームの内部をさらに固定長のブロックという単位に分割して管理するストレージを指す。

^{*21} EBSスナップショット：Amazon EBSボリュームのバックアップデータ。

況を確認したところ42個のブロックストレージのうち、22個がアタッチされておらず、これらを削除することでコンピューティングコストを約1割削減することができたケースがあった。また、他のプロジェクトではタグが付いていない1,000を超えるスナップショットがあり、これらを削除することでコスト削減に繋がった。

4.3 アプリケーションの動きの把握

アプリケーションをデプロイ^{*22}する際、当初の見込みよりも実際はトラフィックが無く、過剰なプロビジョニングをしているケースがある。オンプレミス^{*23}ではリソースを縮退することは難しいが、クラウドであれば適切なリソースに縮退や拡張をすることができる。リソースの使用状況については、AWSであればCloudwatch^{*24}、Google Cloud PlatformはCloud Monitoring^{*25}、Microsoft AzureはAzure Monitor^{*26}といったサービスで確認が可能である。クラウド事業者提供のサービス以外にも、New Relic社やDatadog社などが提供する監視サー

ビスもあり、そういったサービスを活用することで適切なリソースへの変更が可能である。AWSではCompute Optimizerという機能があり、アイドル状態のインスタンスや使用率の低いインスタンスを特定しコストを削減できるレコメンドを行うことができる（図8）。

また、インスタンスタイプの価格は、最新のものが安くなるので、最新のインスタンスタイプへの変更を常に意識する必要がある。

4.4 自動化の検討

常時起動の必要がない検証環境については、深夜・休日を停止することでかなりコストを抑えることができる。例えば、平日深夜5時間停止に加えて土日に停止をさせることで、コストを約6割以下に削減することができる。ここまで停止することができれば、1年程度のコミット利用による価格低減よりも安くなる可能性が高い。多くのリソースを扱う場合は手動で毎日停止することは難しいため、自動化することで漏れ無く停止することができる。また、



図8 AWS Compute Optimizerの操作画面イメージ

- *22 デプロイ：アプリケーションをそれらの実行環境に配置して展開すること。
- *23 オンプレミス：企業がシステムを構成するハードウェアを自社で保有し、自社で保守運用すること。
- *24 Cloudwatch：AWSが提供する、AWSリソースやAWSで実行されるアプリケーションなどの監視サービス。

- *25 Cloud Monitoring：Google Cloud Platformが提供する、Google Cloud PlatformのリソースやGoogle Cloud Platformで実行されるアプリケーションなどの監視サービス。
- *26 Azure Monitor：Microsoft Azureが提供する、AzureのリソースやAzureで実行されるアプリケーションなどの監視サービス。

バックアップの定期実行を設定し、古いバージョンについては自動的に削除する世代管理をすることでコストを低減することができる。

4.5 料金モデルの検討

これまでに記載したコスト削減の取組みを行った上で、どうしても必要になるリソースに対して、料金プランを活用することでさらにコストを削減することができる。AWSでは、コンピューティングリソースに対して、RIやSPという仕組みがある。RIは、OSの種別、リージョン^{*27}単位かAZ（Availability Zone）^{*28}単位、インスタンスファミリー^{*29}などを指定して利用期間をコミットすることで料金を安くできる仕組みである。これに対して、SPは前述の条件（OSの種別、リージョン単位かAZ単位、インスタンスファミリーなど）の指定を緩めて純粋に利用料金のコミットをするため、柔軟に購入ができる分、割引率は低くなる。しかし、新しいインスタンスファミリーが随時発表されることを考えるとSP

をベースにコミットを行うことで柔軟な運用が可能になる。

5. あとがき

本稿では、クラウドコスト最適化の取組みについて、そのポイントとなる考え方を述べ、具体的なノウハウを解説した。クラウドが従量制であること、そのためリソースや利用状況の管理を適切に行うことがコスト最適化の観点で重要となり、管理のためには可視化と効果検証を繰り返す必要があると述べた。コスト最適化には初期のシステム構成からコスト対効果を意識した設計が必要であり、運用フェーズでも継続して利用状況を定期的にチェックし、状況に応じて構成変更、インスタンスタイプの見直し、料金プラン活用などの検討を行うことが必要である。今後は、RIおよびSPの適用率に応じてSPを代表アカウントで購入するなど、ドコモ社全体としてのクラウドコスト削減の検討を行う。

^{*27} リージョン：クラウドサービスを提供するためのデータセンターが配置されている地域。

^{*28} AZ：物理的、ソフトウェア的に自律しているデータセンターの集合単位。

^{*29} インスタンスファミリー：インスタンスの種別を用途別に分類したもので「汎用」「コンピューティング最適化」「メモリ最適

化」などに分かれる。