JAPAN | 2024

aws SUMMIT

AWS-05

AWS コスト管理の最前線

石王 愛
アマゾン ウェブ サービス ジャパン合同会社
技術支援本部
シニアテクニカルアカウントマネージャー

自己紹介

石王　愛（いしおう　あい）

シニアテクニカルアカウントマネージャー

- エンタープライズのお客様のAWS 活用支援をサポート
- コスト最適化推進
- 好きなAWSのサービス：

  AWS Cost and Usage Reports

このセッションでお話しすること

- **話すこと**
  - AWSのコスト増加の原因と担当者を調べ、コストを最適化する方法
- **想定する参加者**
  - これからコスト管理・最適化をはじめたいと思っている CCoE、FinOps チーム、アプリケーションチーム、インフラチームの方
- **話さないこと**
  - 各サービスにおける具体的なコスト最適化方法

リリースしたら満足してませんか？

開発時

開発を始めるときは・・・
コスト試算は慎重に
やります！

運用時

運用に入ったら・・・
あとは知りません

アプリケーション担当

こんなことになっていませんか？

利用料金が上昇してます
原因調査・コスト最適化
お願いします

どこをコスト最適化
できるか確認する体力が
ありません

うちのプロジェクトの
コストどうやって見るん
ですか？

経理

アプリケーション担当

Agenda

1. AWS コスト管理のベストプラクティス

2. ユースケースで考えるコスト最適化

3. まとめ

AWS コスト管理のベストプラクティス

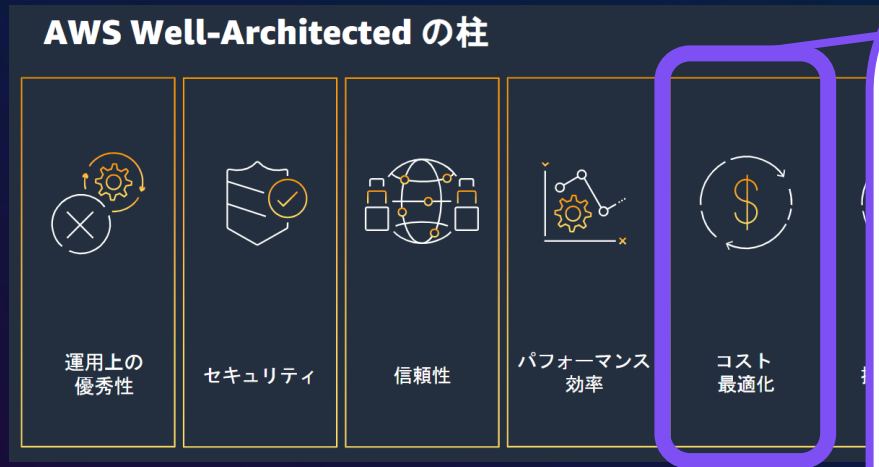
クラウドの料金の特性とは

必要な時に、必要なだけ、IT リソースを提供

光熱費のお支払いと同じ感覚で**利用した分だけお支払い**

Well-Architected Framework
コスト最適化の柱

5 つの設計原則に従うことでコスト最適化を達成することが可能

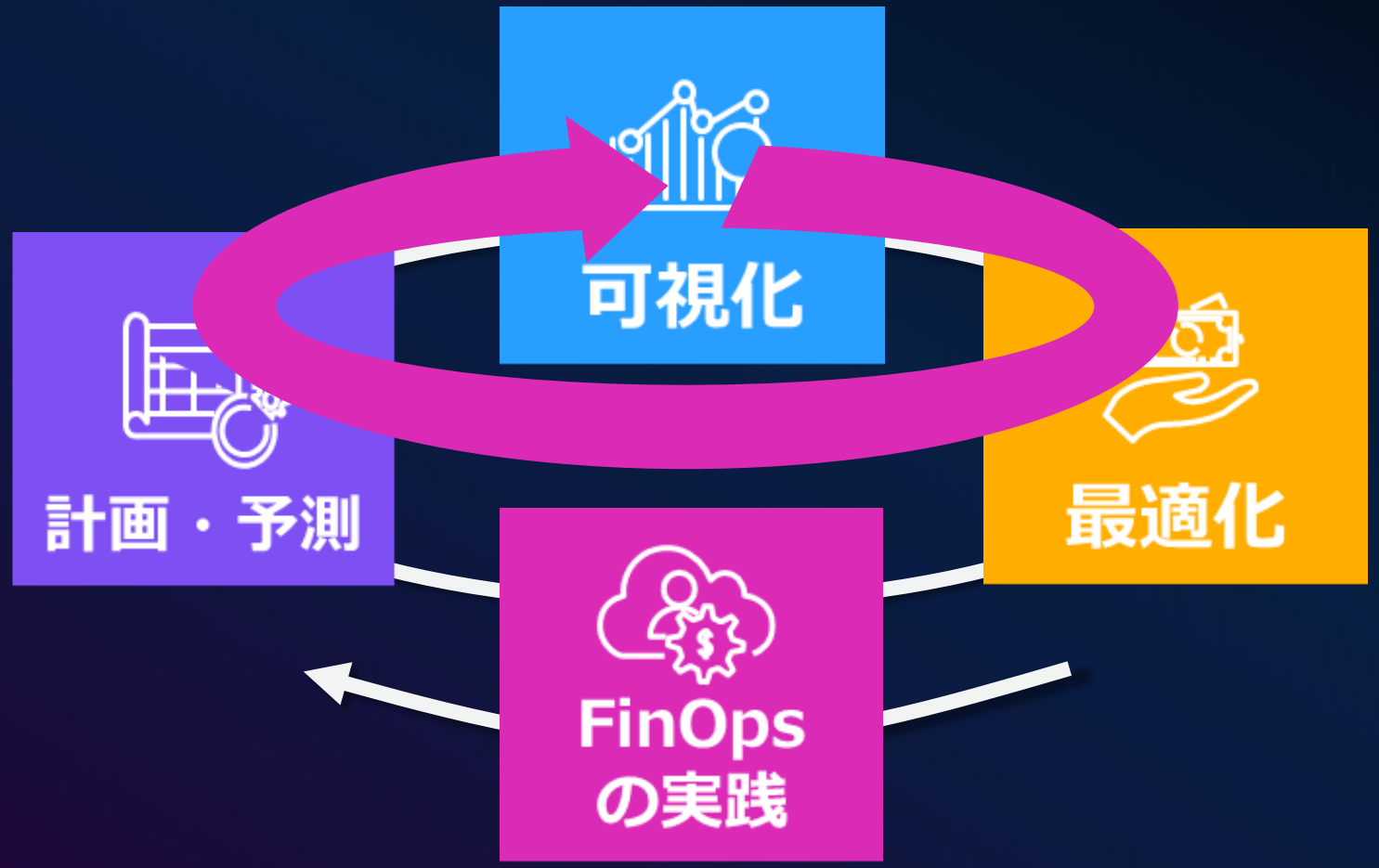


5 つの設計原則

1. クラウド財務管理（CFM）を実装する
2. 不要な「使用」を削減する
3. ビジネス成果の測定値とデリバリコストを使ってコスト効率性を評価する
4. 差別化につながらない高負荷の作業に費用をかけるのをやめる
5. 費用を分析し帰属関係を明らかにする

Cloud Financial Management (CFM) の全体像

CFM には、「可視化」、「最適化」、「計画・予測」、「FinOps の実践」の 4 つの柱があります。



ユースケースで考える
コスト最適化

6月のとある日・・・

5月分の請求書を見たところ
利用料金が上昇してました
原因調査とコスト最適化
お願いします！

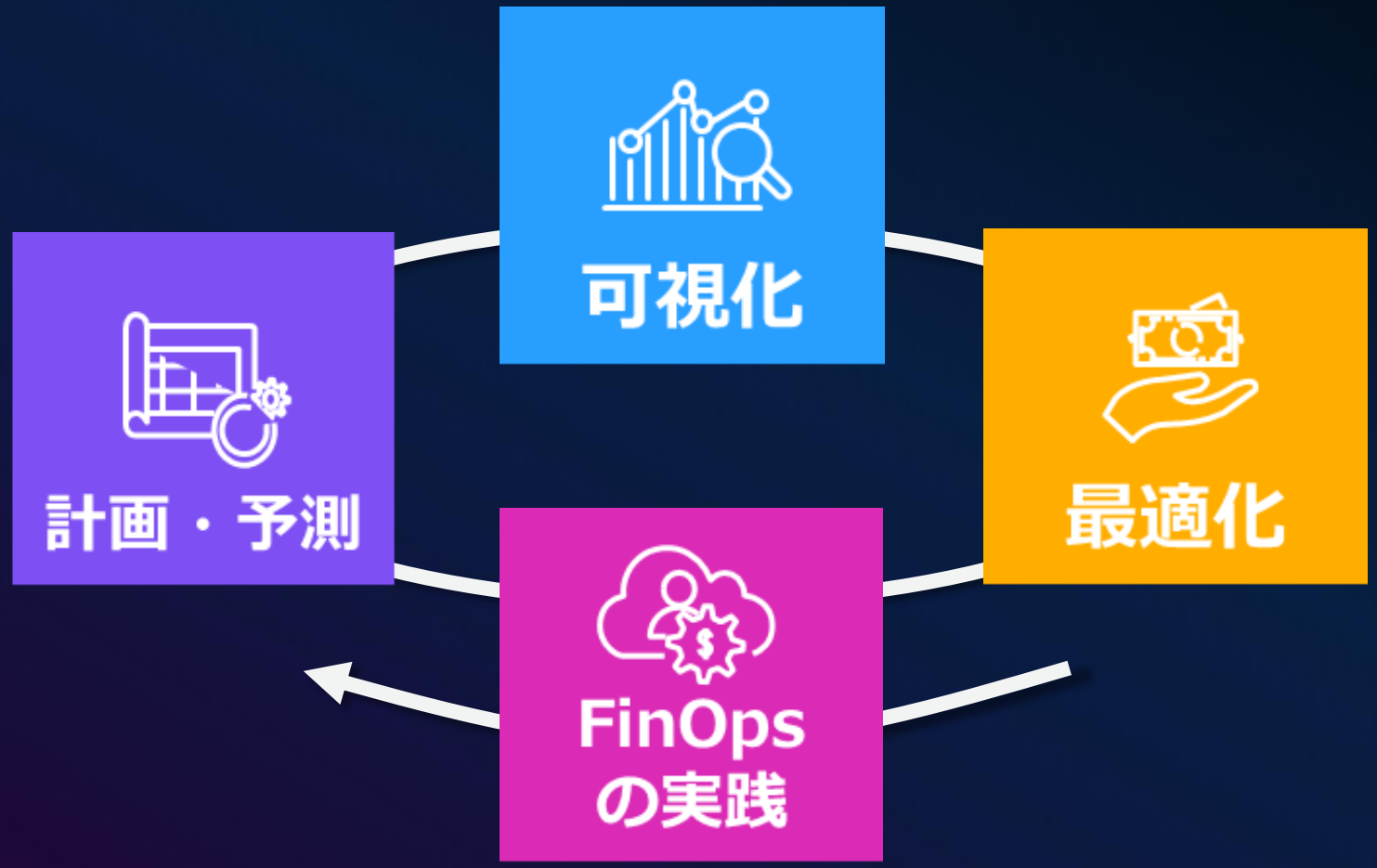
経理
A さん

クラウド管理者
B さん

コスト最適化何からはじめる？

ミッション 1
可視化

CFM の「可視化」から始めましょう！

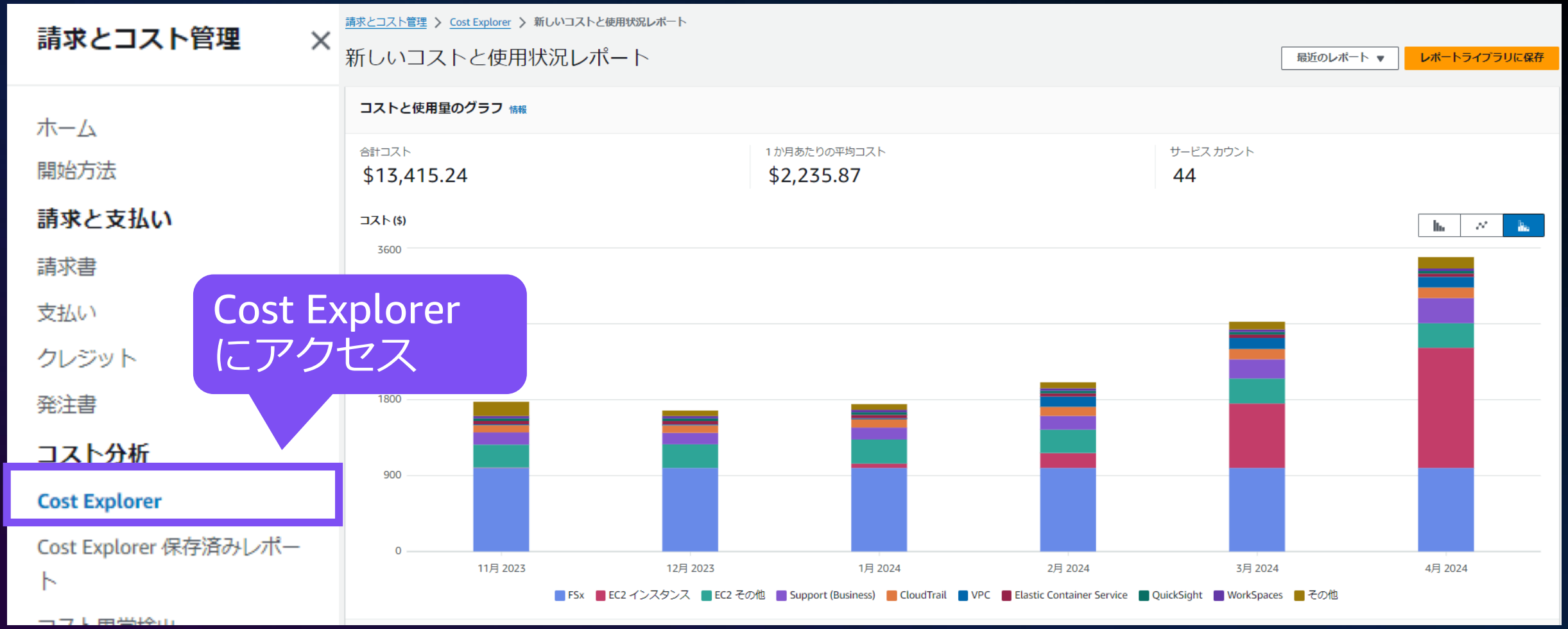


[ミッション１　コストの可視化]

パパっと確認

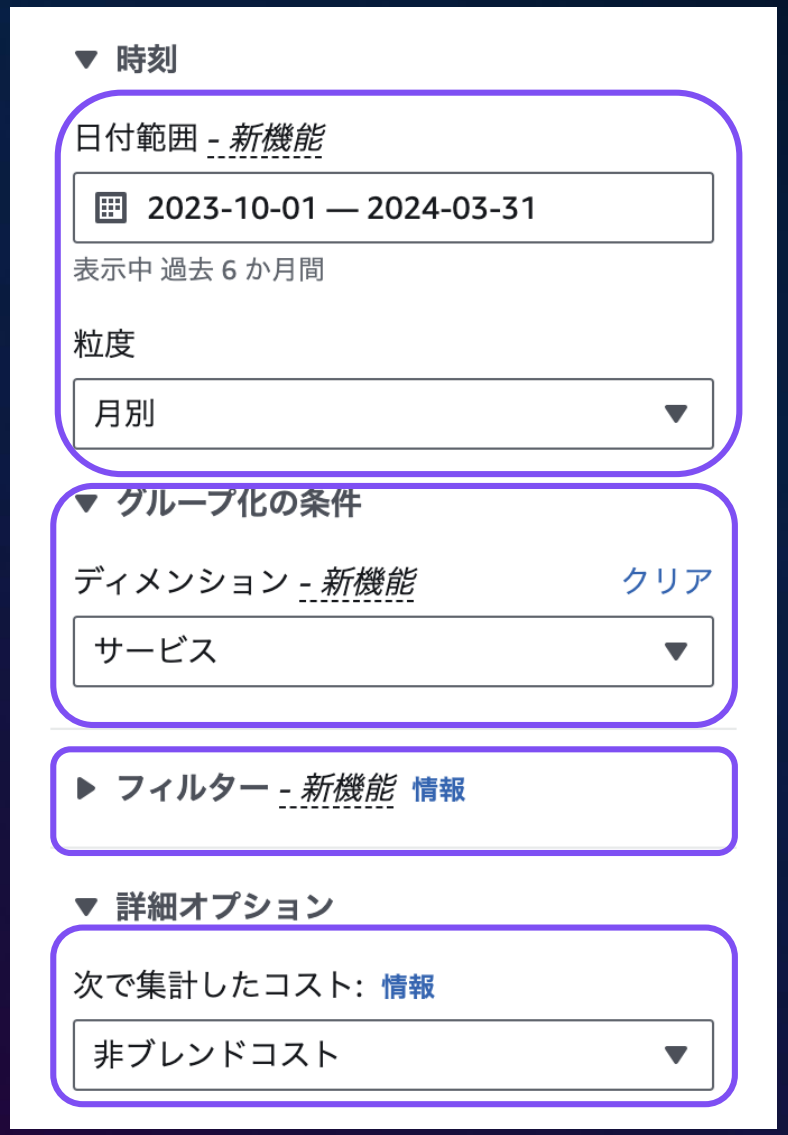
パパっとコストを確認したいときは・・・
AWS Cost Explorer (CE)

ミッション 1
可視化



CE のレポートパラメータ

ミッション 1
可視化



期間・時間粒度

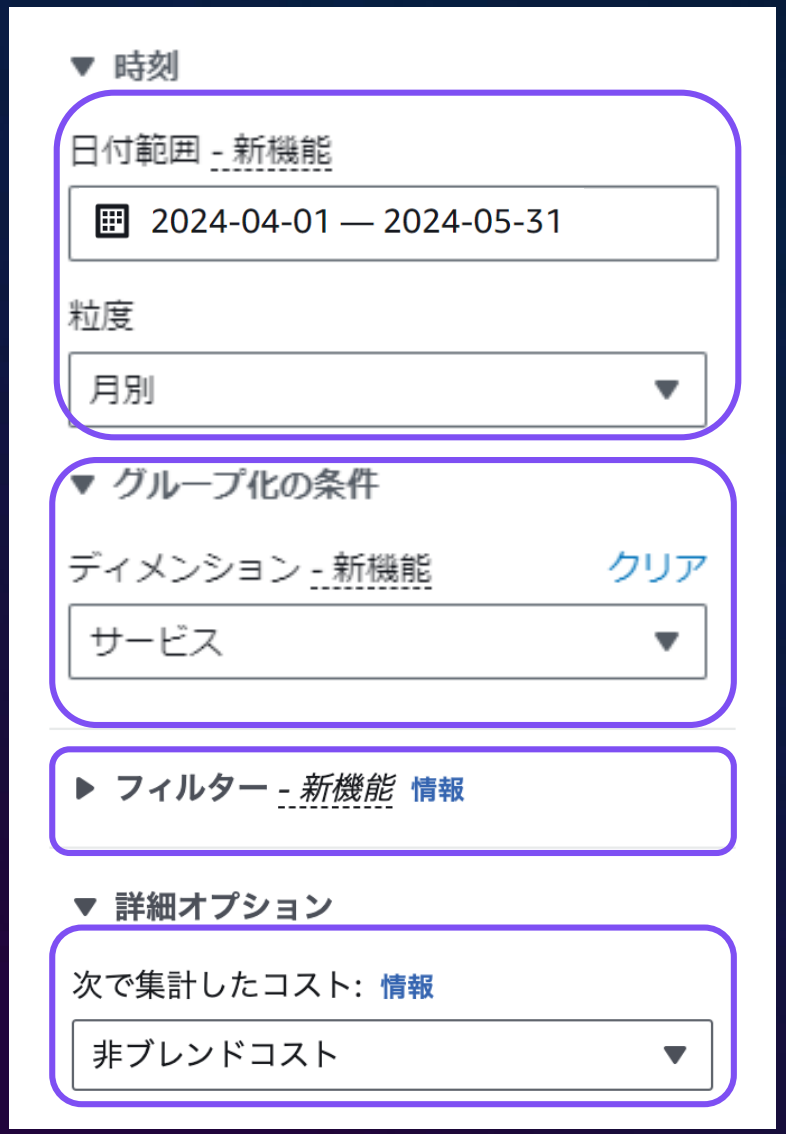
グルーピング

フィルタリング

集計コスト
（コストの表示方法）

5月にどのサービスが増えたか確認しましょう

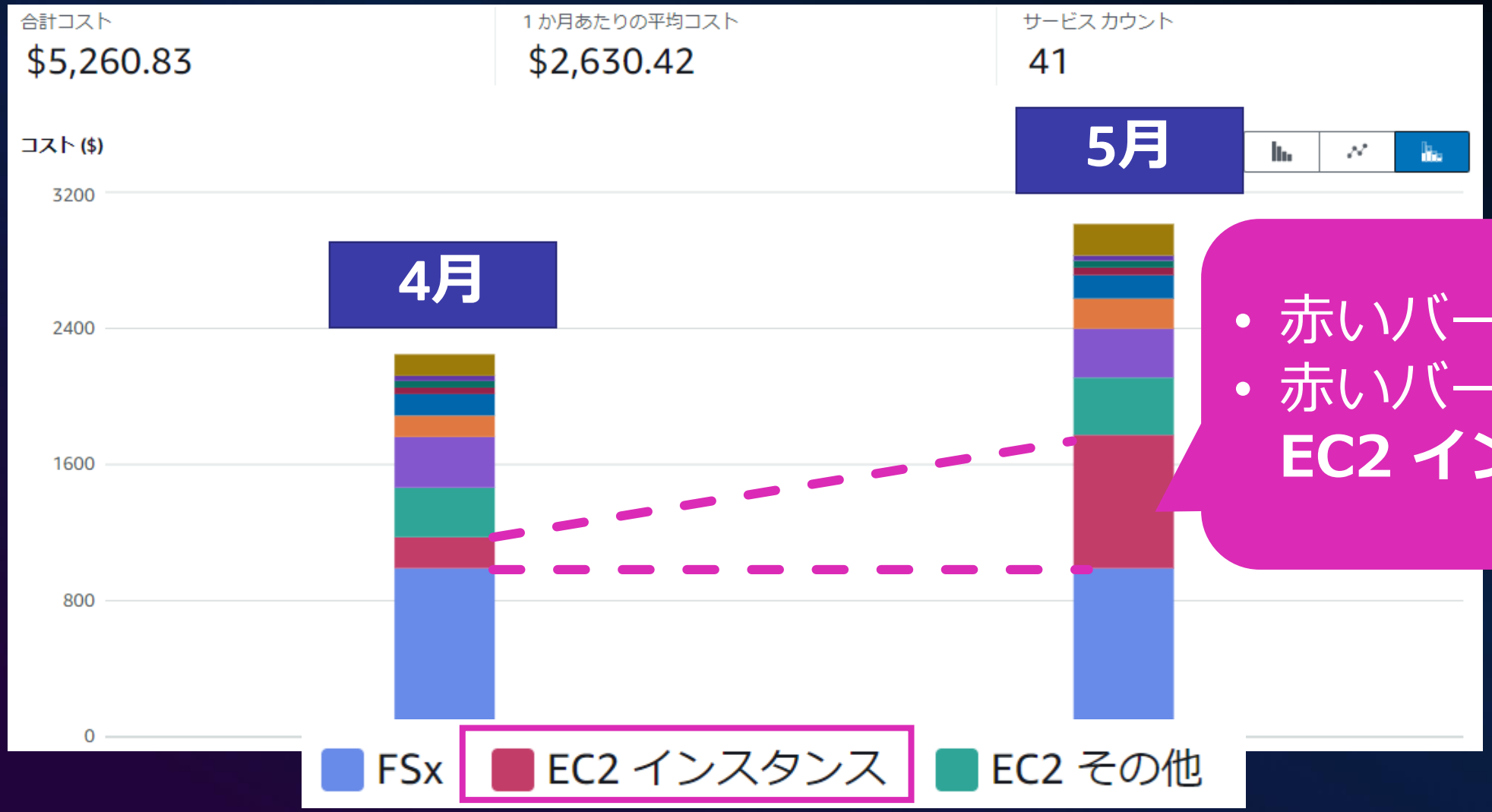
ミッション 1
可視化



期間・時間粒度	4月と 5月を比較したいので 日付：2024/4/1～2024/5/31 粒度：月
グルーピング	どのサービスが伸びているか 見たいので「サービス」を選択
フィルタリング	なし
集計コスト	請求書と同じ金額が見たいので 「非ブレンドコスト」を選択

5月にどのサービスが増えたか確認しましょう

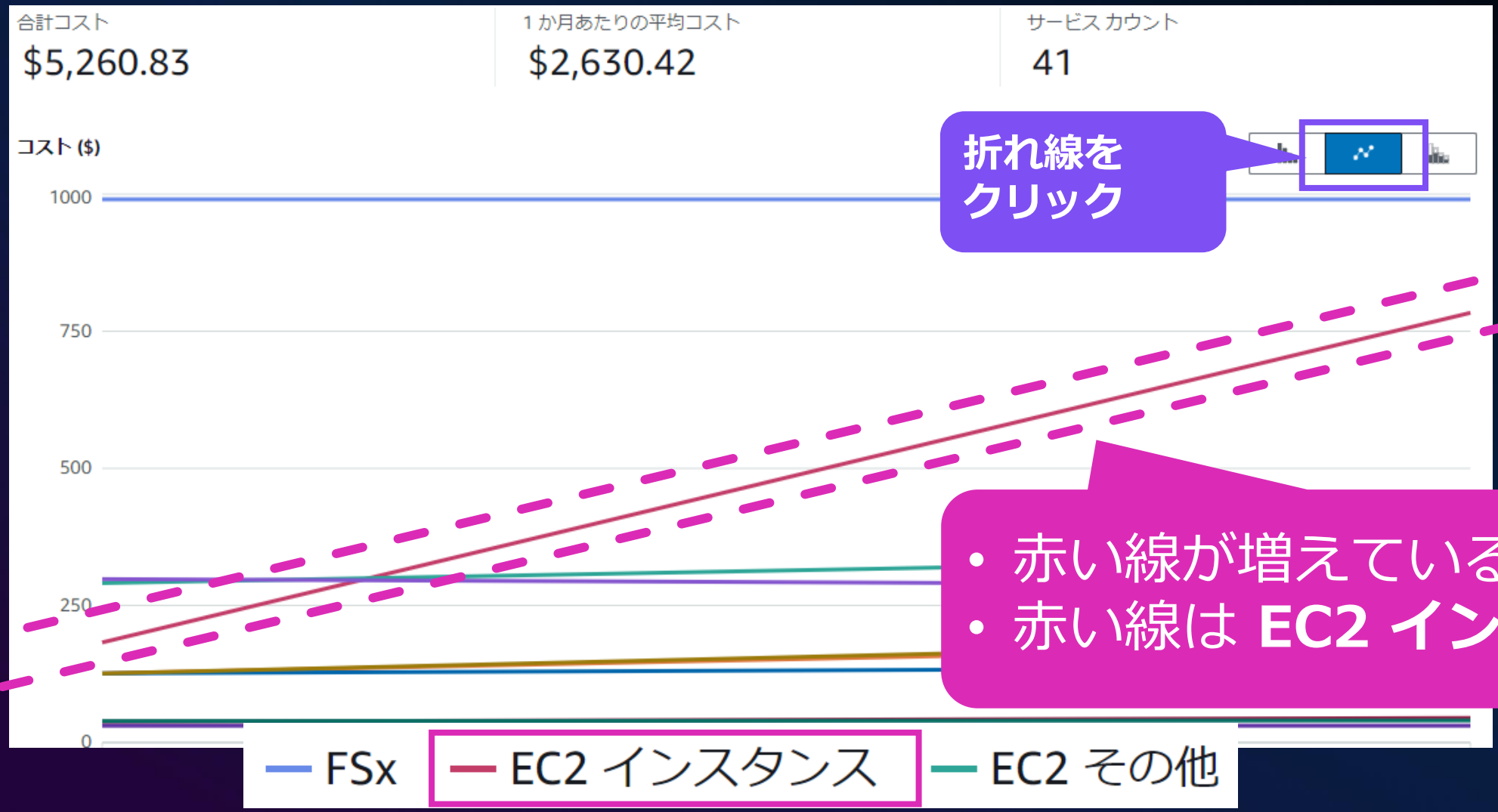
ミッション 1
可視化



- 赤いバーが増えている
- 赤いバーは
  EC2 インスタンス

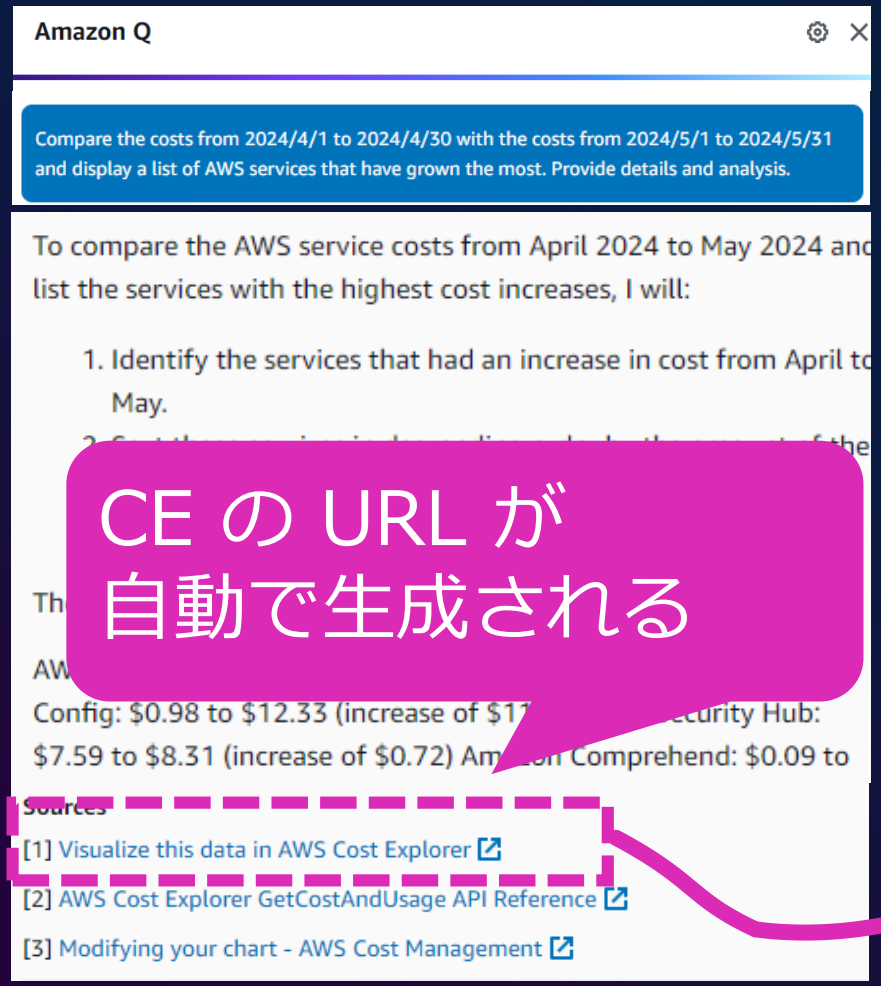
折れ線なら変化がわかりやすい

ミッション 1
可視化



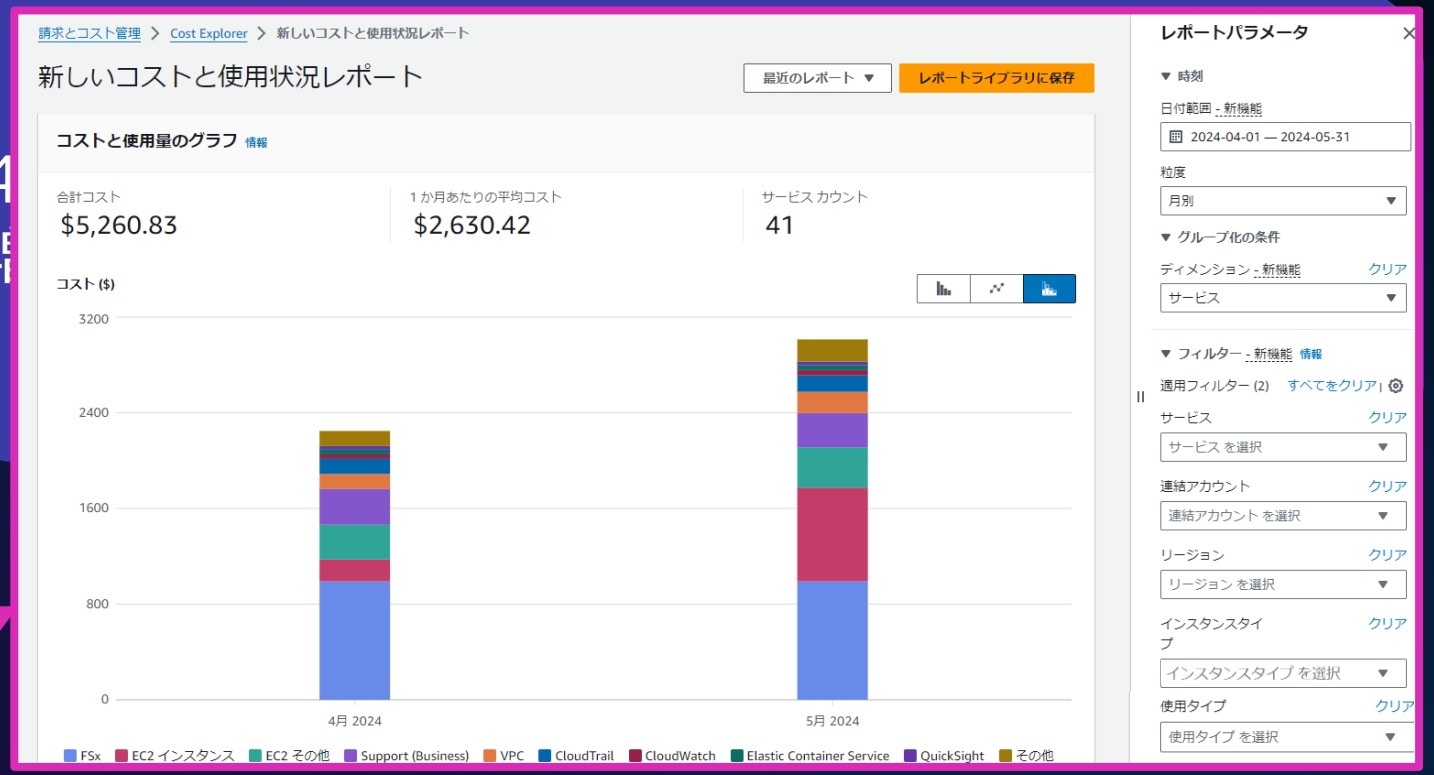
生成 AI に聞くことも
Amazon Q Developer can now help you understand your costs (preview)

ミッション 1
可視化



CE の URL が
自動で生成される

4月と 5月を比較して最もコストが高い
AWS サービスとその分析をしてください



5月に EC2 のどのインスタンスタイプが増えたか確認しましょう

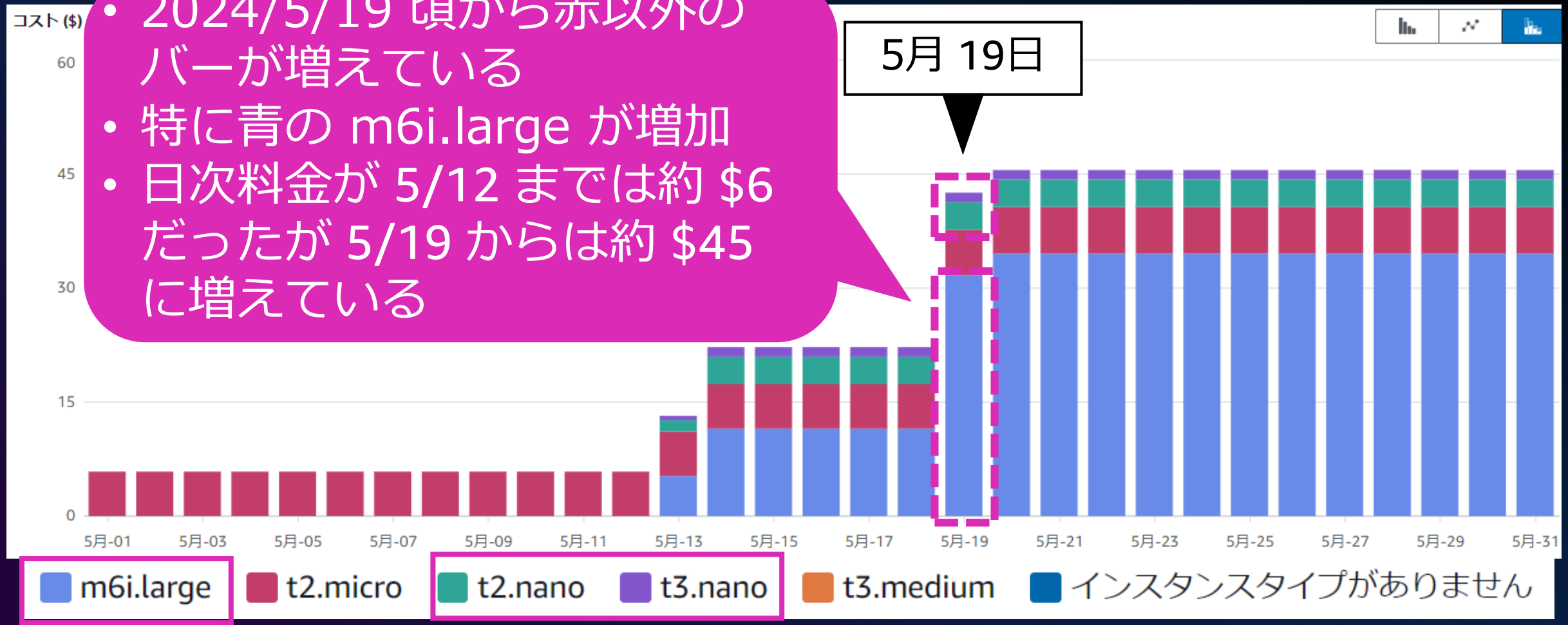
ミッション 1
可視化

期間・時間粒度	5月の詳細を見たいので 日付：2024/5/1～2024/5/31 粒度：日
グルーピング	どのインスタンスタイプが伸びているか見たいので 「インスタンスタイプ」を選択
フィルタリング	EC2 に絞ってみたいので 「 サービス：EC2 インスタンス」を選択
集計コスト	請求書と同じ金額が見たいので「非ブレンドコスト」を選択

5月に EC2 のどのインスタンスタイプが増えたか確認しましょう

ミッション 1
可視化

- 2024/5/19 頃から赤以外のバーが増えている
- 特に青の m6i.large が増加
- 日次料金が 5/12 までは約 $6 だったが 5/19 からは約 $45 に増えている



調査の結果を報告したところ・・・

ミッション 1
可視化

5月 19日から m6i.large、 t2.nano 、
t3.nanoの料金が一日当たり
約 $40 増えていることが
料金上昇の原因です！

クラウド管理者
B さん

経理
A さん

おお、素晴らしい！
その EC2 を管理しているのは
誰ですか？
コスト最適化できないか確認
してもらいましょう

タグを使ってリソースの管理者を識別する

ミッション１
可視化

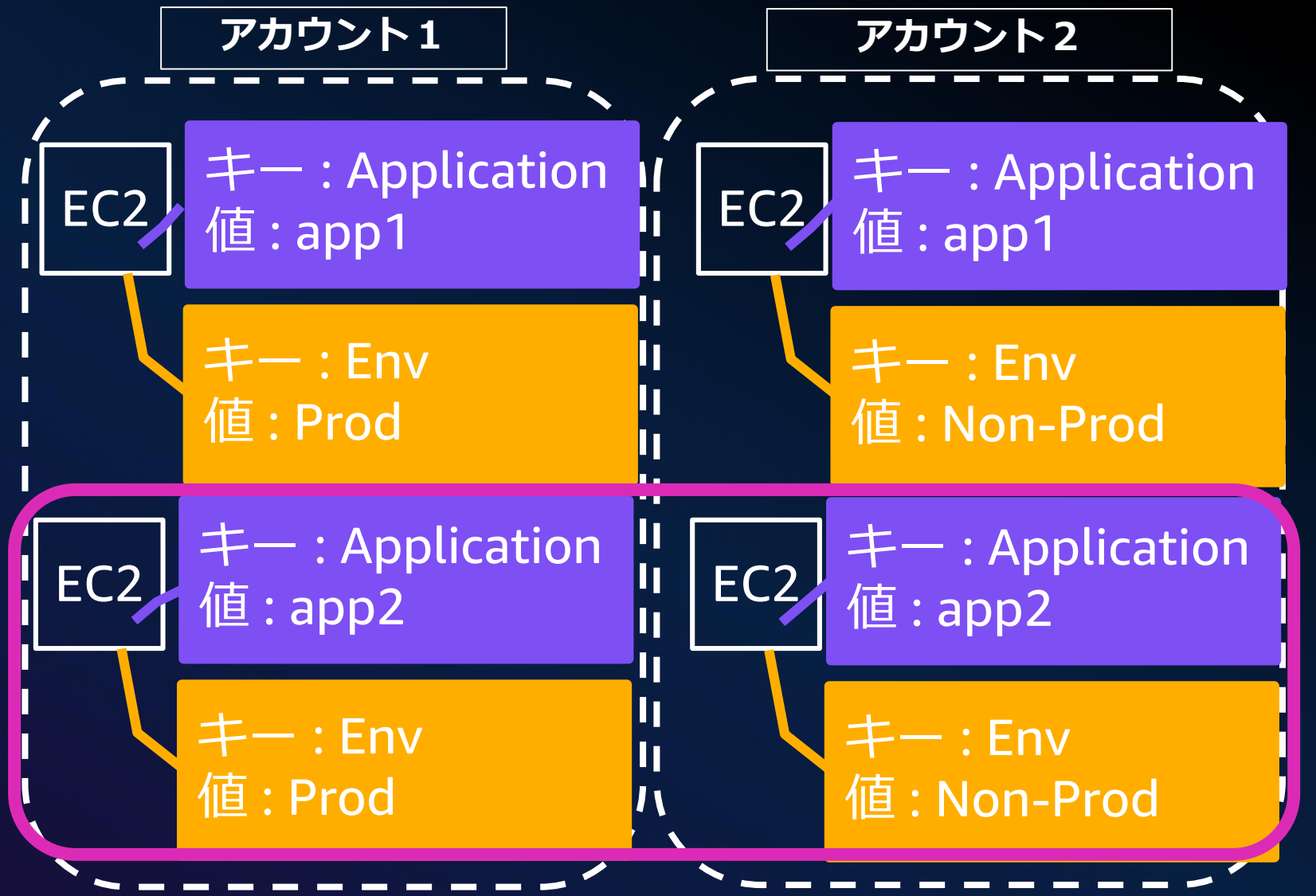
タグはAWSリソースに
つける「名前」や
「ラベル」のようなもの

タグを使ってリソースに
様々な情報を付加できる

- 組み合わせは**キー**と**値**
- タグの単位例：

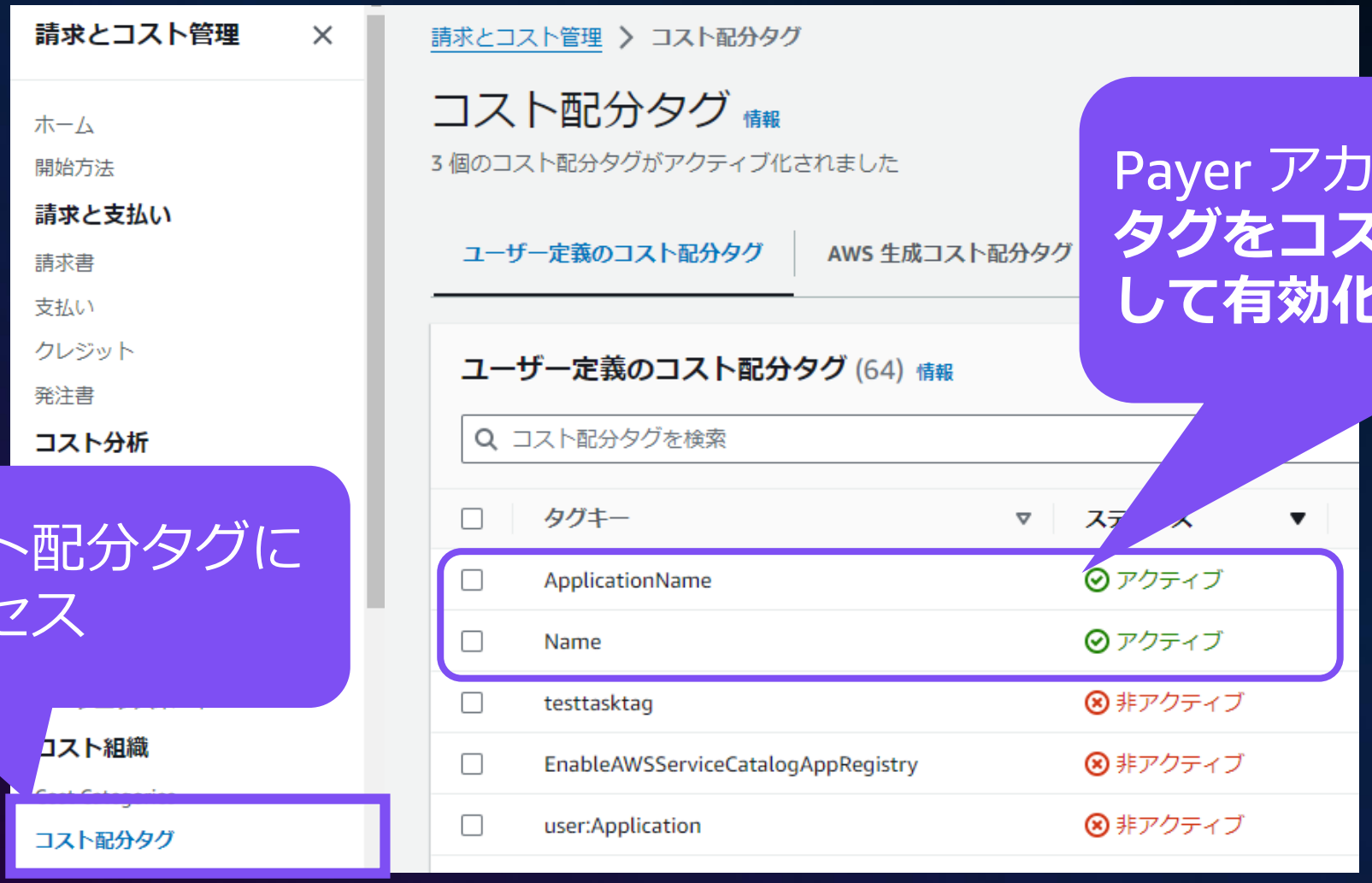
  プロジェクト・環境・

  アプリケーション・部門　等



タグをコスト関連のサービスで見るには

ミッション 1
可視化



Payer アカウントで
タグをコスト配分タグとして有効化することが必要

コスト配分タグにアクセス

ミッション 1
可視化

コスト配分タグのバックフィル機能を利用し、過去まで遡ってコスト関連のサービスでタグを見ましょう(最大 12か月間)

2023年 5月　タグ付与
タグキー： ApplicationName
タグ値： app2

2024年 3月　タグ付与
タグキー： ApplicationName
タグ値： app2

ApplicationName
コスト配分タグ
有効化

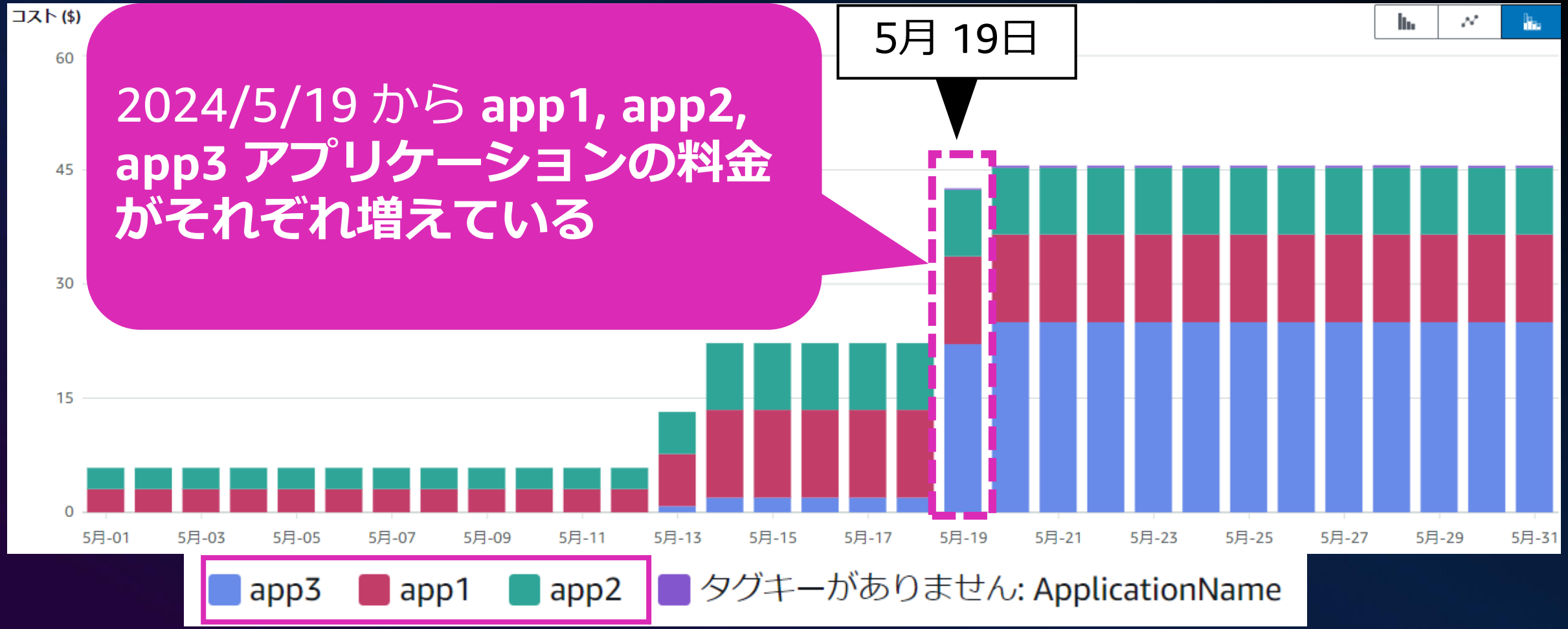
5月に誰の EC2 が増えたか確認しましょう

ミッション 1
可視化

期間・時間粒度	5月の詳細を見たいので 日付：2024/5/1～2024/5/31 粒度：日
グルーピング	どのチームの EC2 が伸びているか見たいので**「タグ」** **タグキー：ApplicationName を選択**
フィルタリング	EC2 に絞ってみたいので 「 EC2 インスタンス」を選択
集計コスト	請求書と同じ金額が見たいので 「非ブレンドコスト」を選択

5月に誰の EC2 が増えたか確認しましょう

ミッション 1
可視化



でもタグの付与って漏れちゃうことありませんか？

ミッション 1
可視化

予防的

適切なタグが付与されるようガードレールを設定

- Service control policies
- IAM Policies

能動的

開発時にタグを付与する

- AWS CloudFormation
- AWS Service Catalog

受動的

タグが付いていないリソースの検知/検索・タグを付与する

- AWS Resource Explorer
- AWS Tag Editor
- AWS Config
- AWS Security Hub

タグが付いていないリソースを発見できる様々なサービス

ミッション 1
可視化

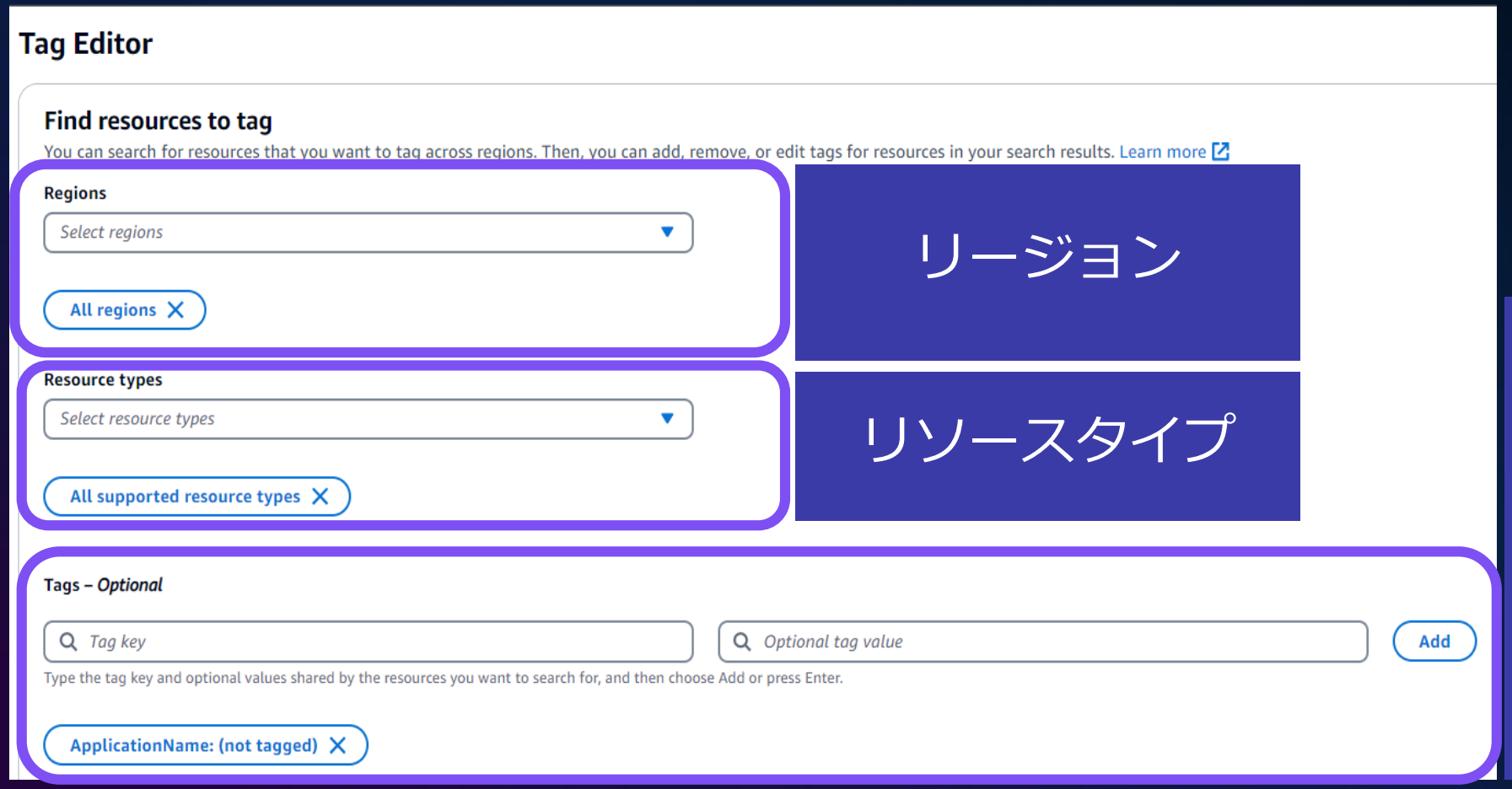
種別	サービス	クロスアカウント検索	料金
検索	AWS Resource Explorer	○	無料
	AWS Tag Editor	×	無料
検知・通知	AWS Config Rule (required-tags)	○	有料
	AWS Security Hub	○	有料

AWS Resource Explorer で 指定したタグが付いていないリソースを検索

ミッション 1
可視化

各アカウントの AWS Tag Editor で タグが付いていないリソースを検索

ミッション 1
可視化



リージョン

リソースタイプ

検索したいタグキー

※複数指定可
※タグキー & タグの値の組合せでも検索可能

各アカウントの AWS Tag Editor でタグを付与！

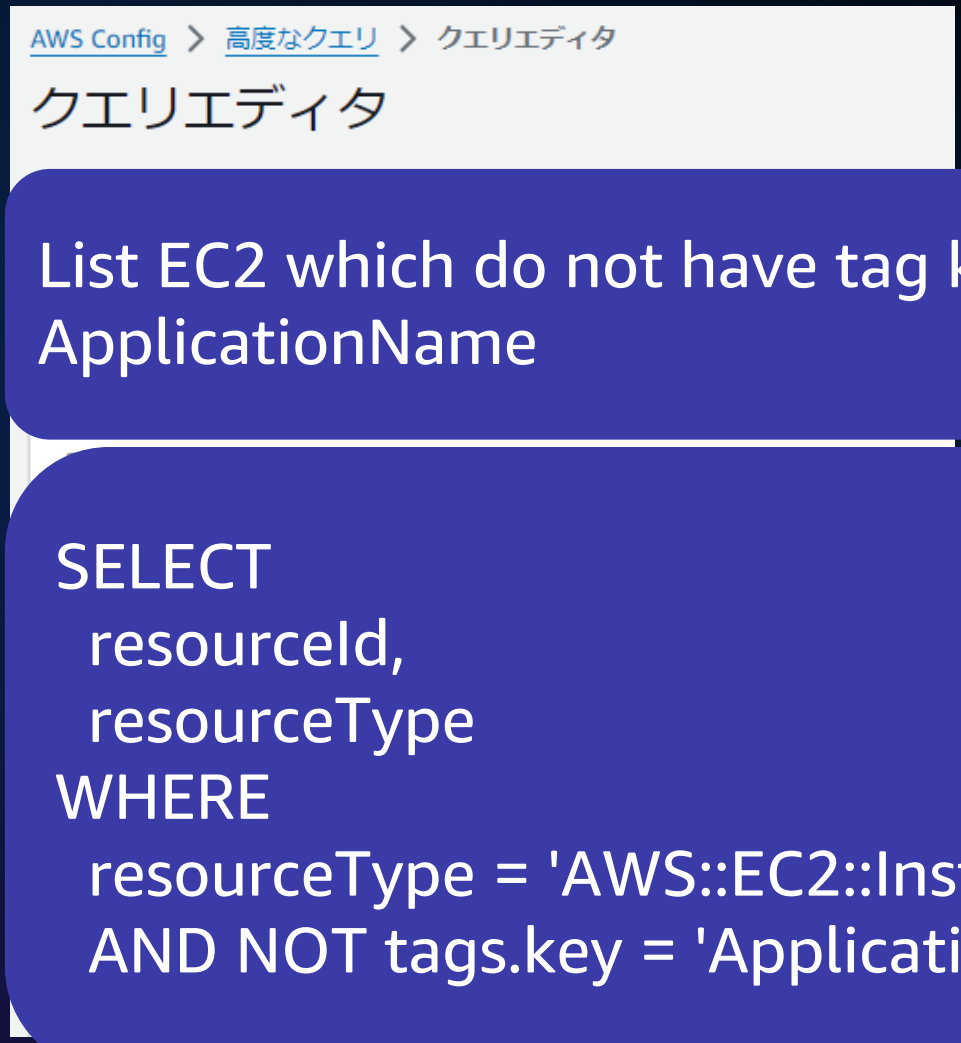
ミッション 1
可視化

ミッション 1
可視化

補足 : AWS Config 高度なクエリ 自然言語クエリプロセッサ (preview)

AWS Config 高度なクエリでもタグがついていないリソースを Organizations 内を横断で検索できます。

自然言語クエリプロセッサを利用すれば簡単な英語で尋ねるだけで、ほしい情報を検索する SQL を生成できます。



補足 :タグの付与にはこんな裏技も AWS Service Catalog AppRegistry

ミッション 1
可視化

あとから CloudFormation(CFN) スタックのリソースにまとめてタグ付けするときに便利

- 自動で **AWSApplication** タグがリソースに付与される
- 自動でコスト配分タグ **AWSApplication** が有効化される

ここまでのまとめ

ミッション 1
可視化

- **AWS Cost Explorer** でパパっとコストを確認しよう！

  - まずは AWS Cost Explorer を使って、いつ、どのインスタンスタイプが増えたのかを絞り込みましょう。
  - 次にタグを使って、どのチームのリソースが増えたのかを絞り込む。コスト配分タグの有効化も忘れずに。
  - タグが無いものは AWS Resource Explorer で検索し AWS Tag editor を使って付与しましょう。

よし、担当者がわかった！

ミッション 1
可視化

増えた原因は
app1, app2, app3 アプリケーション
のリソースだとわかりました！

クラウド管理者
B さん

経理
A さん

ありがとうございます！
では 各アプリケーションの
EC2 一覧を抽出して担当者に
内容を確認してもらいましょう！

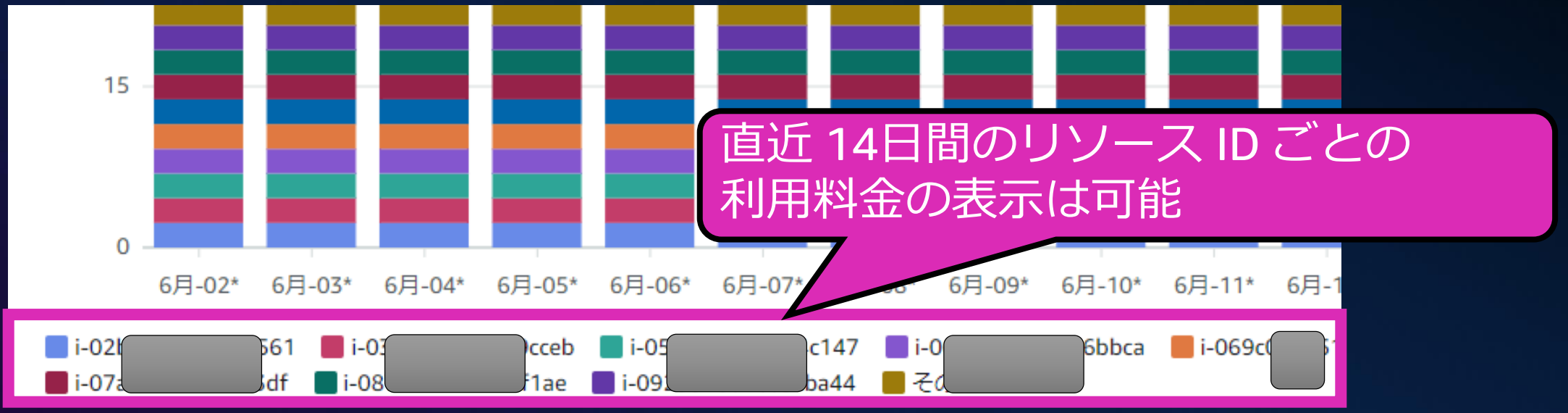
[ミッション１　コストの可視化]

細かく確認

ミッション 1
可視化

アプリケーションチームごとの EC2 一覧を作成する

AWS Cost Explorer では**直近 14日間のリソース ID** を確認できる(*)。
しかし、グルーピング条件は一つしか指定できないため、
アプリケーションチームごとのリソース ID 一覧は作成できない。



* **請求とコスト管理> コスト管理の設定> 履歴データおよび詳細データの設定**
で「日単位の詳細度のリソースレベルのデータ」有効化が必要（無料）

ミッション 1
可視化

AWS Cost and Usage Reports (CUR) ならアプリケーションチームごとの EC2 一覧を作成できる

line_item_usage	line_item_usage_	line_item_usage_type	line_item_product_code	product_i	line_item_	line_item_resource_id	line_item_	line_item_	resource_tags
123456789123	2024/5/1 0:00	BoxUsage:t2.micro	AmazonEC2	t2.micro	Usage	i-03[redacted]958	0.0116		{user_application_name=app1, user_aws_application=arn:aws:resource-groups:us-east-1:123456789123:group/dep1/00x1hqj2xxkbye9z1rcj1jsco8}
12				t2.micro	Usage	i-03[redacted]958	0.0116		{user_application_name=app1, user_aws_application=arn:aws:resource-groups:us-east-1:123456789123:group/dep1/00x1hqj2xxkbye9z1rcj1jsco8}
12				t2.mic		[redacted]958			{user_application_name=app1, user_aws_application=arn:aws:resource-groups:us-east-1:123456789123:group/dep1/00x1hqj2xxkbye9z1rcj1jsco8}
123456789123	2024/5/1 3:00	BoxUsage:t2.micro	AmazonEC2	t2.micro	Usage	[redacted]958	0.0116		{user... use... 1:12...
123456789123	2024/5/1 4:00	BoxUsage:t2.micro	AmazonEC2	t2.micro	Usage	i-03[redacted]958	0.0116		{use... use... 1:12...
123456789123	2024/5/1 5:00	BoxUsage:t2.micro	AmazonEC2	t2.micro	Usage	i-03[redacted]958	0.0116		{use... user_aws_applic...esource-groups:us-east-1:123456789123:group...0x1hqj2xxkbye9z1rcj1jsco8}
123456789123	2024/5/1 6:00	BoxUsage:t2.micro	AmazonEC2	t2.micro	Usage	i-03[redacted]958	0.0116		{user_application_name=...p1, user_aws_application=arn:aws:resource-groups:us-east-1:123456789123:group/dep1/00x1hqj2xxkbye9z1rcj1jsco8}

1 時間単位の
コストデータが見れる

リソース ID も
わかる

各リソース ID のタグ
情報もわかる

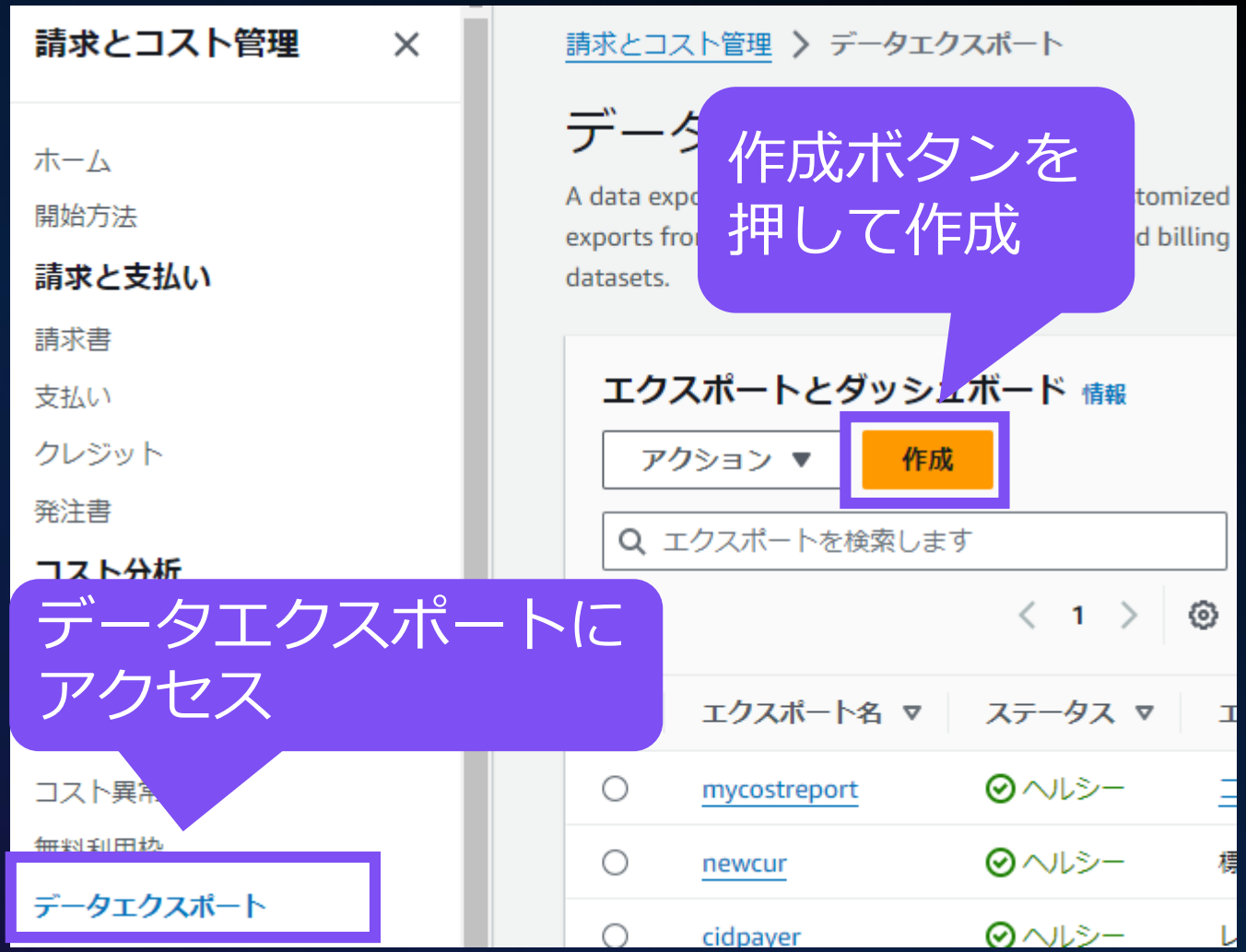
AWS Cost and Usage Reports (CUR)

ミッション 1
可視化

AWS の利用状況と
利用料金情報を最も細かく
包括的に提供する
コストデータ

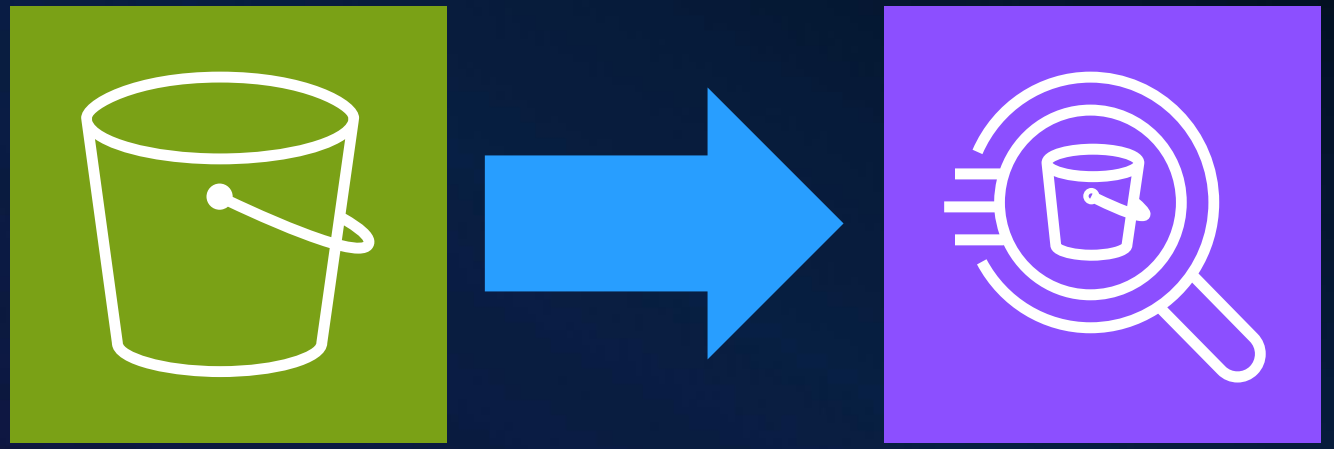
- 格納先 S3 を指定して作成
- CUR を作成するときに時間・日・月単位などの時間粒度やリソース ID を含めるかといったことが設定時に指定可能

作成方法は[こちら]



ミッション 1
可視化

CUR は Amazon Athena で SQL を使って分析するのがおすすめ！



CUR を作成すると
Amazon S3 に
格納される

Amazon Athena で
SQL を
使った分析も可能

CUR 作成からAmazon Athena で分析するための設定はこちら

ミッション 1
可視化

CUR を使えば・・・
アプリケーション毎の EC2 一覧の抽出も簡単！

BillingPeriod	tag_name	tag_value	AccountName	InstanceType	ResourceId	Hours	Cost
May-24	user_application_name	app1	Prod	t2.micro	i-0222222222222221	744	8.6304
May-24	user_application_name	app1	Prod	t2.micro	i-0333333333333339	743	8.6188
May-24	user_application_name	app1	Non-Prod	t2.nano	i-066666666666666a	442	2.5636
May-24	user_application_name	app1	Non-Prod	t2.nano	i-0999999999999953	442	2.5636
May-24	user_application_name	app2	Prod	t2.nano	i-0111111111111fc	442	2.5636
May-24	user_application_name	app2	Prod	t2.nano	i-0555555555555553	442	2.5636
May-24	user_application_name	app2	Non-Prod	t2.nano	i-0888888888888881	442	2.5636
May-24	user_application_name	app3	Prod	t2.nano	i-0444444444444447	442	2.5636

ミッション 1
可視化

CUR を使えば・・・
EC2 ではなくコンテナだったとしても大丈夫！

追加の設定で Amazon ECS, Amazon EKS の各 Task や Pod 単位の利用状況に基づきコストを配分したデータを取得することができます。

- コスト管理の設定で「コスト配分データを分割」を有効化
- CUR を作成するときに「コスト配分データの分割(SCAD)」を有効化

lineItem/ ResourceID	lineItem/ lineItemType	lineItem/ UsageType	lineItem/ UnblendedCost	lineItem/ NetUnblendedCost	savingsPlan/ SavingsPlanEffectiveCost	savingsPlan/ NetSavingsPlanEffectiveCost	splitLineItem/ ParentResourceId	splitLineItem/ SplitUsage	splitLineItem/ SplitCost	splitLineItem/ NetSplitCost	splitLineItem/ UnusedCost	splitLineItem/ NetUnusedCost
i-12345	SavingsPlan CoveredUsage	BoxUsage: m5.xlarge	1.5	1.4	1	0.8						
EC2-Pod1	Usage	EKS-EC2-vCPU-Hours						1	0.14	0.11	0	0
EC2-Pod1	Usage	EKS-EC2-GB-Hours						4	0.08	0.06	0.01	0.01
EC2-Pod2	Usage	EKS-EC2-vCPU-Hours						1.9	0.27	0.21	0	0
EC2-Pod2	Usage	EKS-EC2-GB-Hours						6	0.12	0.09	0.02	0.01
EC2-Pod3	Usage	EKS-EC2-vCPU-Hours							0.14	0.11	0	0
EC2-Pod3	Usage	EKS-EC2-GB-Hours						2	0.04	0.03	0.01	0
EC2-Pod4	Usage	EKS-EC2-vCPU-Hours					i-12345	1	0.14	0.11	0	0
EC2-Pod4	Usage	EKS-EC2-GB-Hours					i-12345	2	0.04	0.03	0.01	0

CPU ,メモリ使用率に基づいて算出した Pod ごとのコスト

補足：改善された CUR 2.0 を使いましょう！

ミッション 1
可視化

昨年秋に CUR2.0 が発表され、今までの CUR は「レガシー CUR」に。
今後は**改善された CUR2.0** をご利用頂くことをお勧めします！

レガシーCUR
のご利用者

Reserved Instance とか
Savings Plans 購入したら
CUR の列が増えた

コスト配分タグを
追加したら
CUR の列が増えた

「支払アカウント ID 」
「使用アカウント ID 」しか
ないから名称は自分で紐づけ

CUR2.0
のご利用者

CUR の列が変わらない
含める列を自分で選べる　か
ら不要な列は除外できる

コスト配分タグを追加しても
resource_tags 列の値に
まとめて表示される

「支払アカウント名 」
「使用アカウント名」列がある

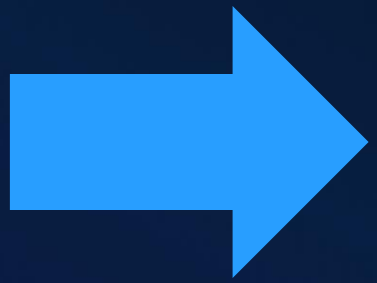
お勧め：簡単にコストダッシュボードを作成
AWS Cost and Usage Dashboard (CUD)

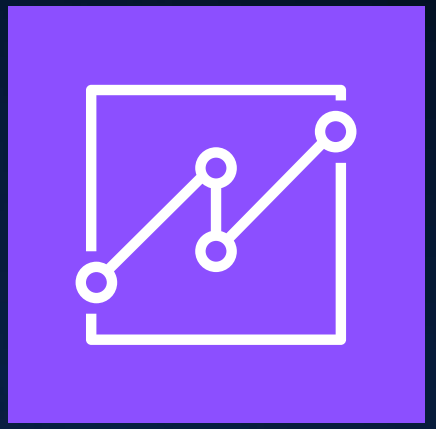
ミッション 1
可視化

ミッション 1
可視化

数クリックだけで自動で CUR の作成から Amazon QuickSight の設定まで行います

CUR を作成して
Amazon S3 に
格納

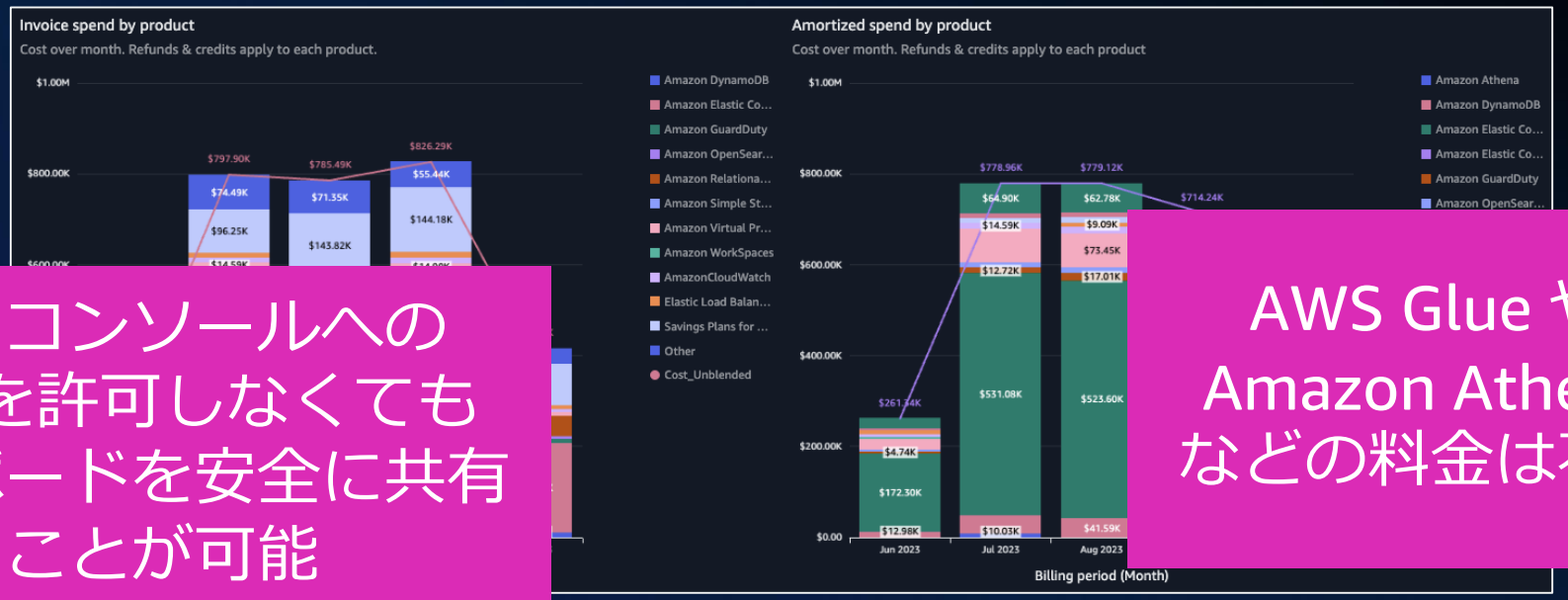




Amazon QuickSight
を使って 可視化

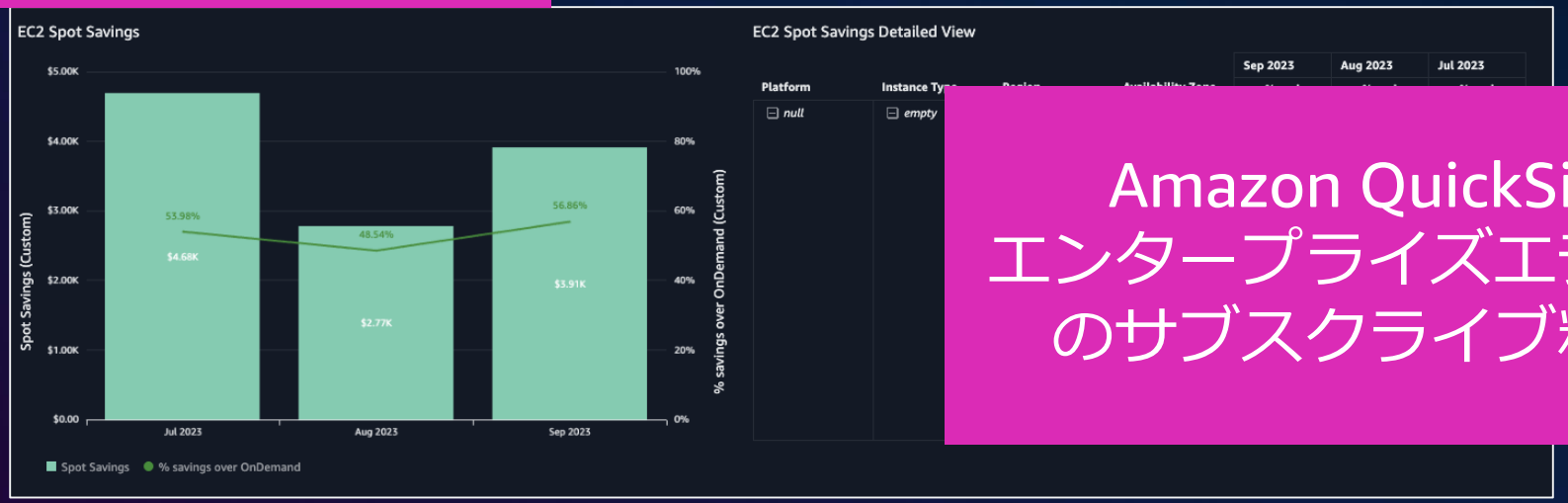
AWS Cost and Usage Dashboard (CUD)

ミッション 1
可視化



閲覧者にコンソールへの
アクセスを許可しなくても
ダッシュボードを安全に共有
することが可能

AWS Glue や
Amazon Athena
などの料金は不要



Amazon QuickSight の
エンタープライズエディション
のサブスクライブ料金のみ

ここまでのまとめ

ミッション 1
可視化

- AWS Cost and Usage Reports (CUR) で
  アプリケーションチーム毎の EC2 一覧を作成しよう！
  - CUR を使えばリソース単位や時間単位のコストを細かく
    分析できる。

- 簡単に AWS Cost and Usage Dashboard (CUD) で
  コストダッシュボードを作成しよう！

アプリケーションチームに連絡！

EC2 一覧を送付します
コスト最適化の余地がないか
確認してください！

クラウド管理者
B さん

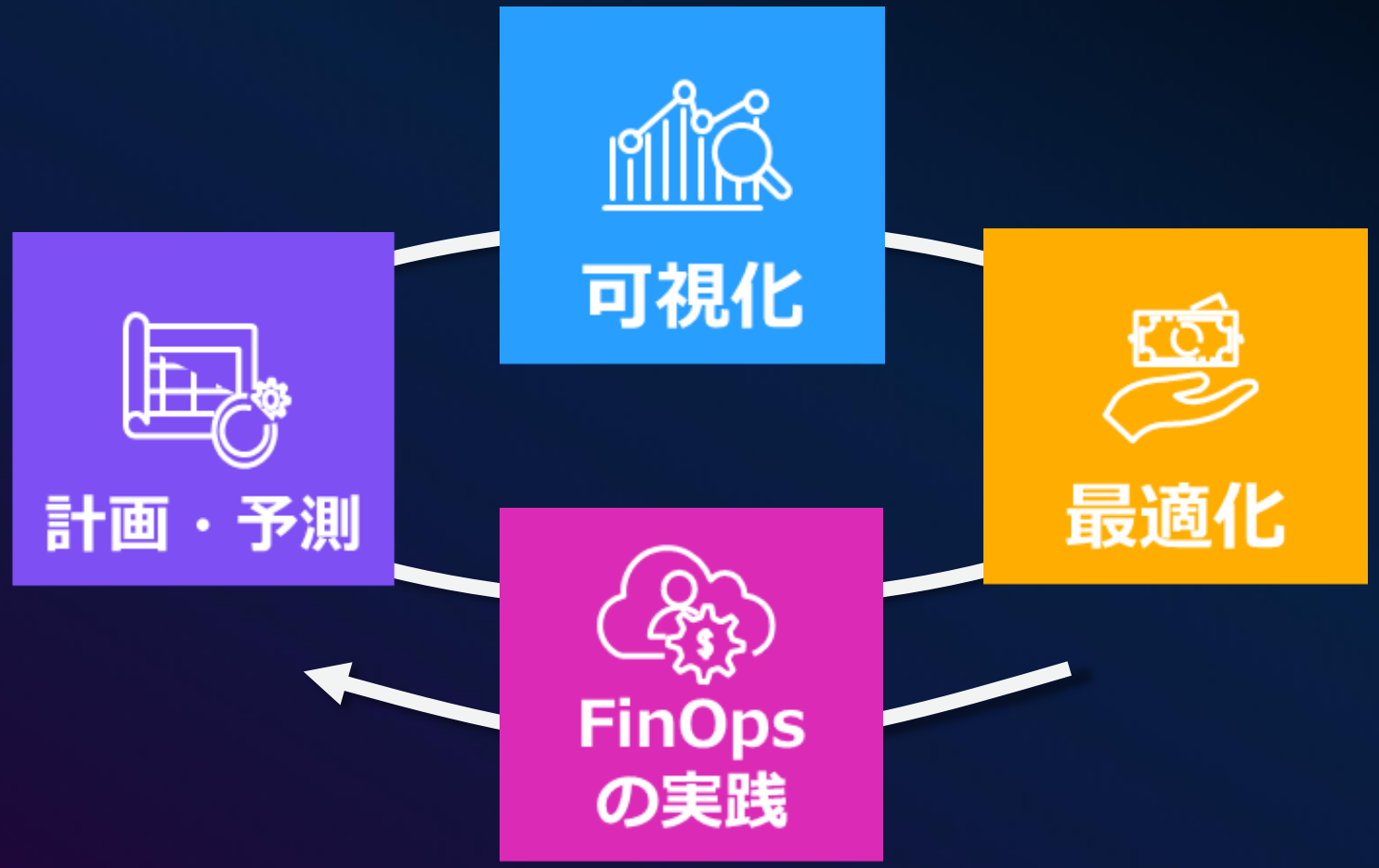
了解！

app1, app2, app3
アプリケーション
担当者

コスト最適化何からはじめる？

ミッション 2
最適化

CFM の「最適化」に取り組みましょう！



[ミッション 2　コスト最適化]

クイックウィン最適化

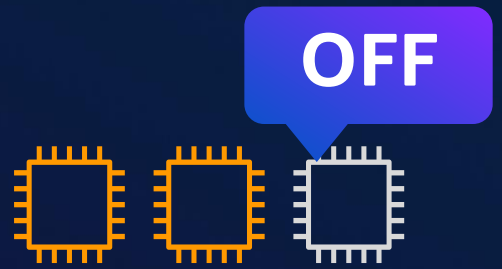
コスト最適化は何からはじめたらいいの？

ミッション 2
最適化

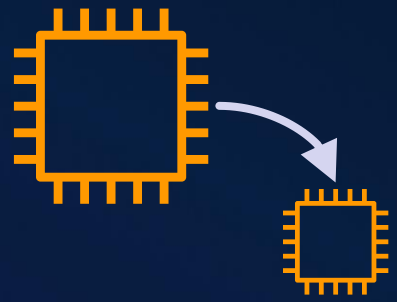
クイックウィン最適化

① 需要に応じた
スケジューリング

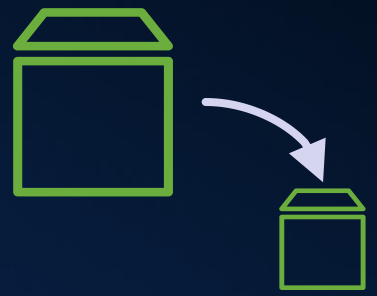
② 未使用
リソース停止



③ 適切な
インスタンスタイプ
の見直し



④ 適切な
ストレージの
見直し



⑤ Savings Plans の購入

インスタンス

EBS

①需要に応じたスケジューリング

ミッション 2
最適化

まずは開発環境などの
常時稼働不要なサーバを
夜間・土日停止しましょう

時間	月	火	水	木	金	土	日
8:00-20:00	○	○	○	○	○	×	×
20:00-8:00	×	×	×	×	×	×	×

サービス	特徴
Amazon EventBridge スケジューラと AWS Lambda	• 一番シンプルな実装方法
AWS Systems Manager Resource Scheduler	• 使いやすい UI • マルチアカウント制御可能
AWS Instance Scheduler	• AWS ソリューション • マルチアカウントの制御が可能 • 柔軟なスケジュール設定

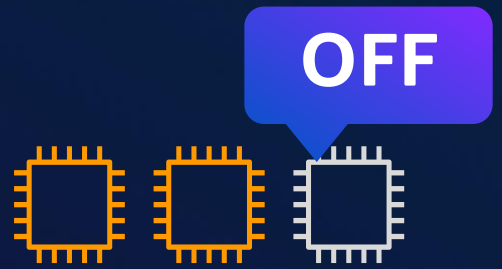
コスト最適化は何からはじめたらいいの？

ミッション 2
最適化

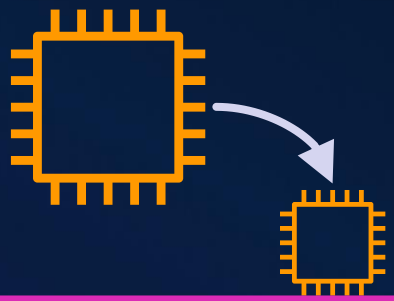
クイックウィン最適化

① 需要に応じた
スケジューリング

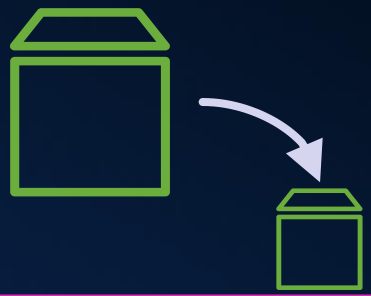
② 未使用
リソース停止



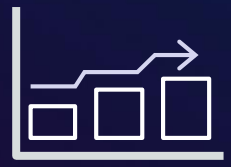
③ 適切な
インスタンスタイプ
の見直し



④ 適切な
ストレージの
見直し



⑤ Savings Plans の購入



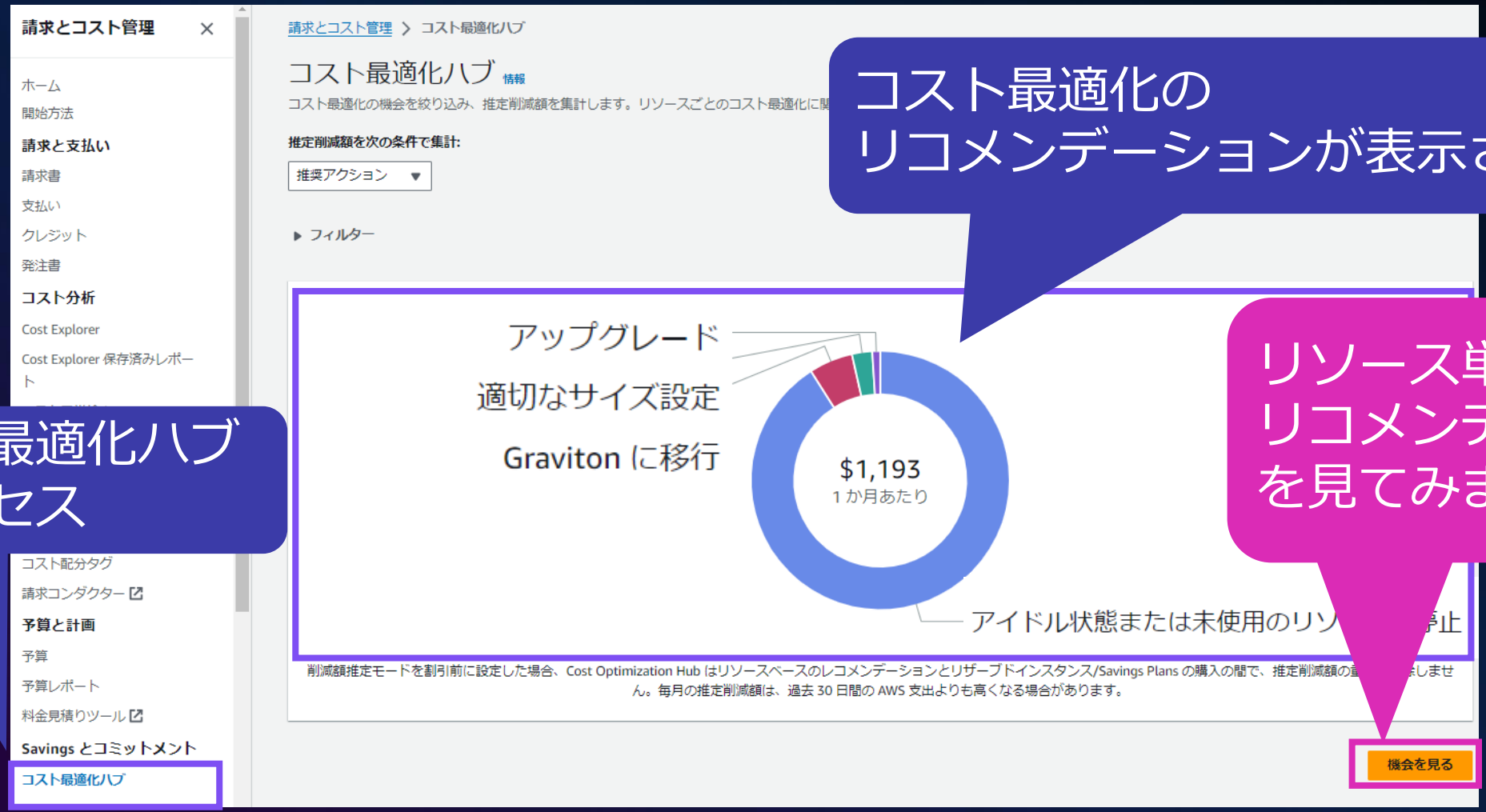
全てのリコメンデーションを
一度に確認することが可能

インスタンス

EBS

一度にリコメンデーションを確認できる！
AWS Cost Optimization Hub

ミッション 2
最適化



コスト最適化ハブにアクセス

コスト最適化のリコメンデーションが表示される

リソース単位のリコメンデーションを見てみましょう！

リコメンデーションの詳細

ミッション 2
最適化

まずリコメンデーションの対象を、見たい EC2 に絞ります

リコメンデーションの詳細

ミッション 2
最適化

各 EC2 にどのアクションをとるか<u>この画面だけで</u>決めることができる！

一つのインスタンスについて複数のコスト最適化
リコメンデーションと推定削減額が表示される

推定削減額があるリソース

テキスト、プロパティまたは値で分布をフィルタリング

リソースタイプ = EC2 インスタンス | タグ = ApplicationName | フィルター

1

	毎月の推定削減額	リソースタイプ	リソース ID	推奨アクション	現在のリソースの概要	推奨リソースの概要	推定削減率	推定月額コスト	実装作業
○	$13.87	EC2 インスタンス	[redacted]ece04f1ae	Graviton に移行	m6i.large	m6g.large	20%	$70.08	非常に高い
○	$0.18	EC2 インスタンス	[redacted]ece04f1ae	アップグレード	m6i.large	m7i-flex.large	1%	$70.08	中
○	$70.08	EC2 インスタンス	[redacted]ece04f1ae	Stop	i-08[redacted]f1ae	-	100%	$70.08	低

リソースの再起動は必要ですか	ロールバックは可能ですか	アカウント ID	リージョン	タグ	日数単位のルックバック期間
はい	はい	19[redacted]316	米国東部 (バージニア北部)	ApplicationName:app2	32
はい	はい	19[redacted]316	米国東部 (バージニア北部)	ApplicationName:app2	32
はい	はい	19[redacted]316	米国東部 (バージニア北部)	ApplicationName:app2	32

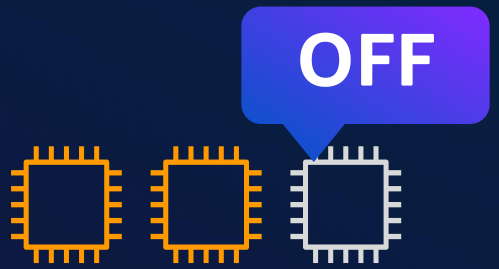
コスト最適化は何からはじめたらいいの？

ミッション 2
最適化

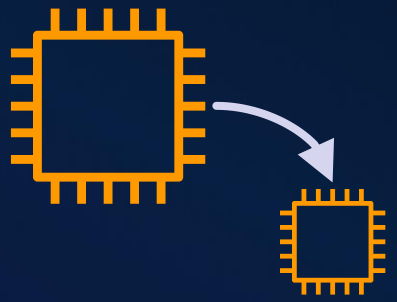
クイックウィン最適化

① 需要に応じた
スケジューリング

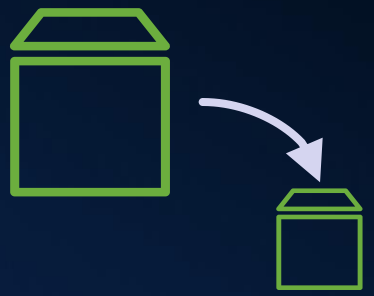
② 未使用
リソース停止



③ 適切な
インスタンスタイプ
の見直し



④ 適切な
ストレージの
見直し



⑤ Savings Plansの購入

適切な稼働、キャパシティへの
変更が完了したら
コミットメントの購入検討を

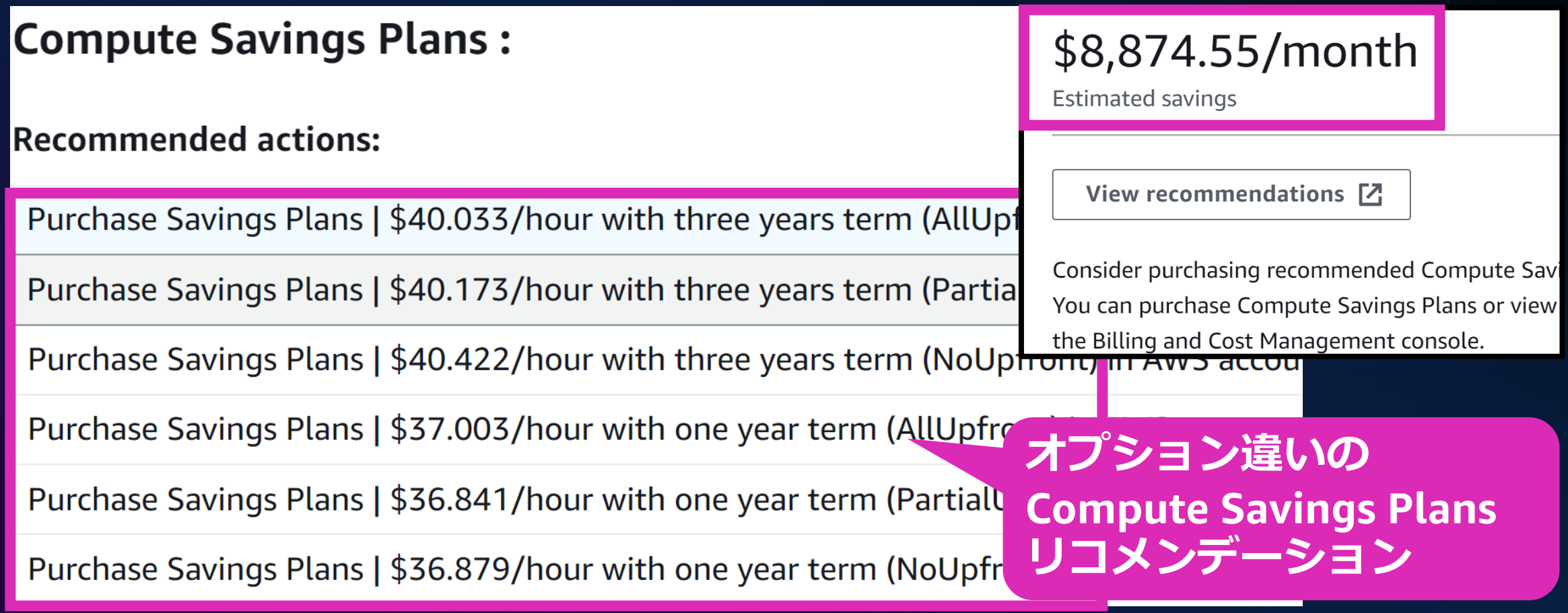
インスタンス

EBS

ミッション 2
最適化

①~③ のアクションが完了したら
⑤ Savings Plansの購入を検討しましょう

推定削減月額



オプション違いの
Compute Savings Plans
リコメンデーション

ミッション 2
最適化

補足 : Amazon CloudWatch natural language query generation

Amazon CloudWatch Logs / Amazon CloudWatch メトリクスについて、簡単な英語で尋ねるだけでほしい情報を検索するSQL を生成できます。

この例では Amazon CloudWatch Logs Insights で請求額が最も高い Lambda 呼び出しの上位 100 件を検索するSQL を生成しています。

ここまでのまとめ

ミッション 2
最適化

- コスト最適化は**クイックウィン最適化**から
  - まずは開発インスタンスなどの夜間休日停止から始めましょう
  - 次に Cost Optimization Hub で低稼働率のインスタンスの停止やインスタンスタイプの変更のそれぞれの最適化効果を比較し、最も適切な施策を実行しましょう
  - それが終わったら Savings Plans などのコミットメントの購入による最適化効果を Cost Optimization Hub で確認し、購入を検討しましょう

その後・・・

ミッション 2
最適化

PoC インスタンスの削除で約 $100
検証環境の夜間休日停止で約 $200
削減できました！
本番稼働したら
Compute Savings Plans も購入予定です！

app1, app2, app3
アプリケーション
担当者

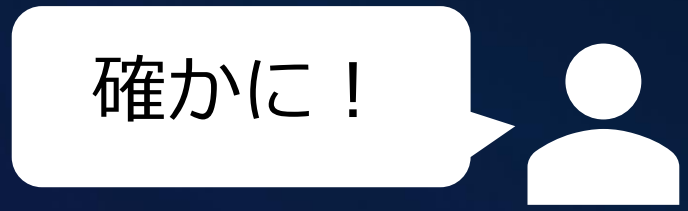
[ミッション 3　計画・予測]

更なる改善

振り返って・・・

今回のような利用料金の上昇は
請求書のタイミングじゃなく、
上がったときに気づけたら
もっと早くコスト最適化
できましたね

経理
A さん

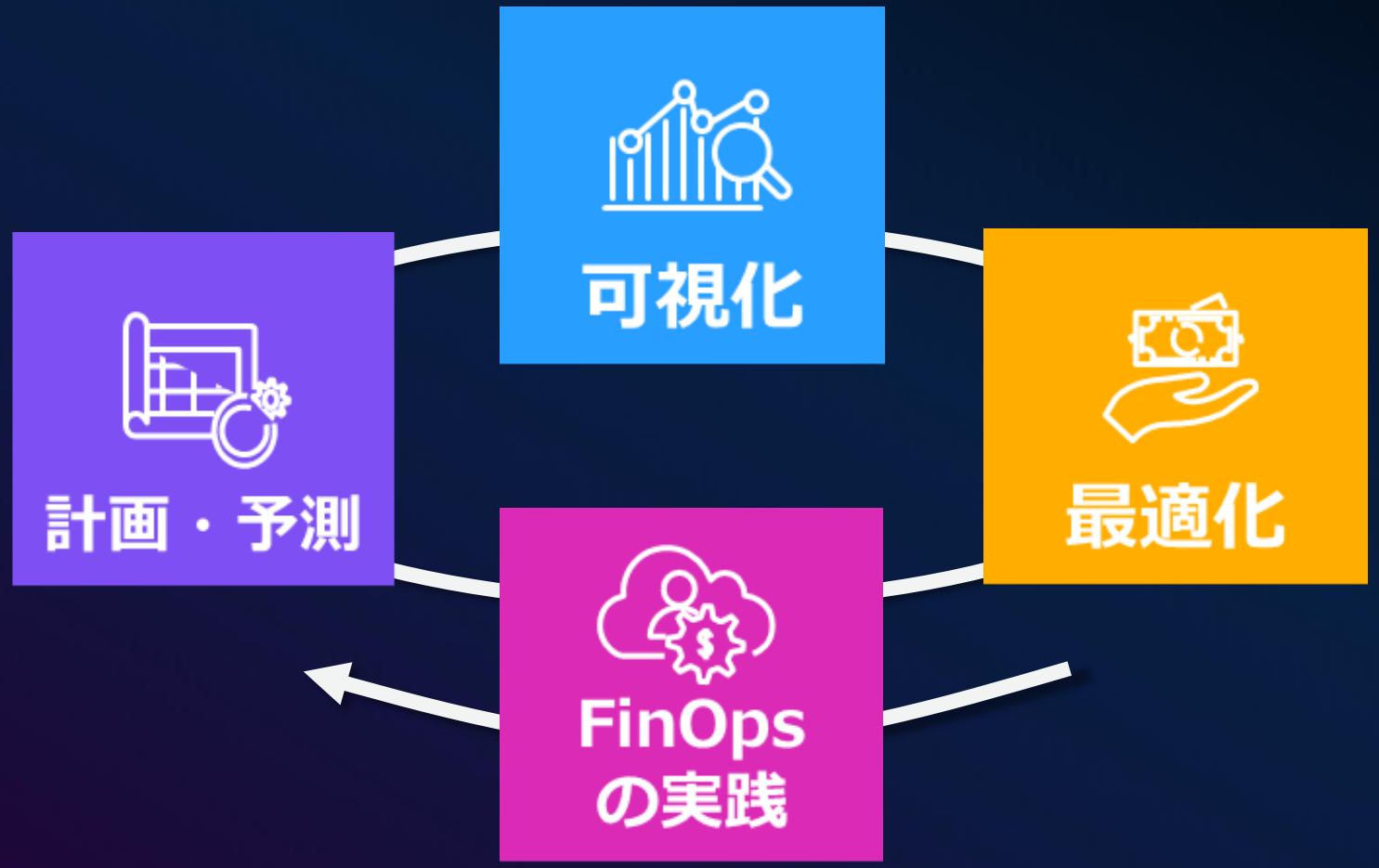


クラウド管理者
B さん

コスト最適化何からはじめる？

ミッション 3
計画・予測

CFM の「計画・予測」に取り組みましょう！



予算と予算アラートを設定しましょう

ミッション 3
計画・予測

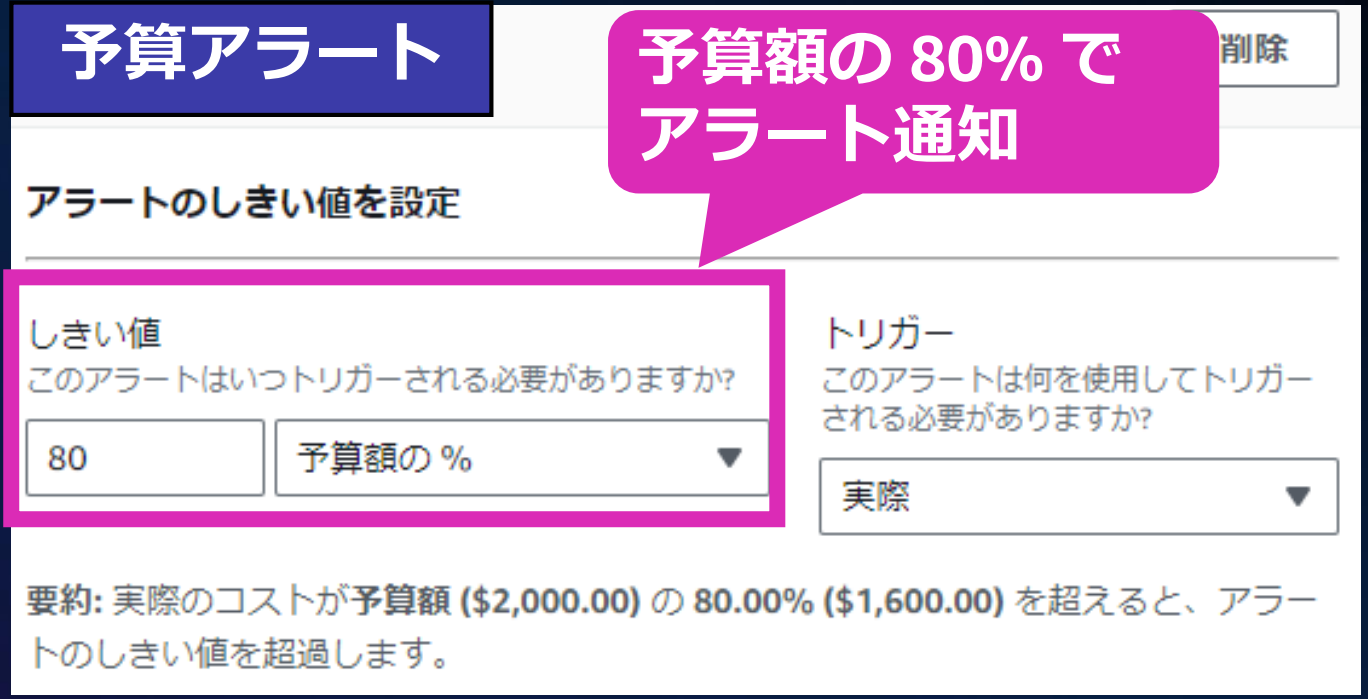
AWS Budgets

予算を超過する前にアラートを通知したり、アラートが発生したらインスタンスを自動停止する等の自動オペレーションを実行できます。

予算

ひと月 $2,500 の予算を設定

予算アラート

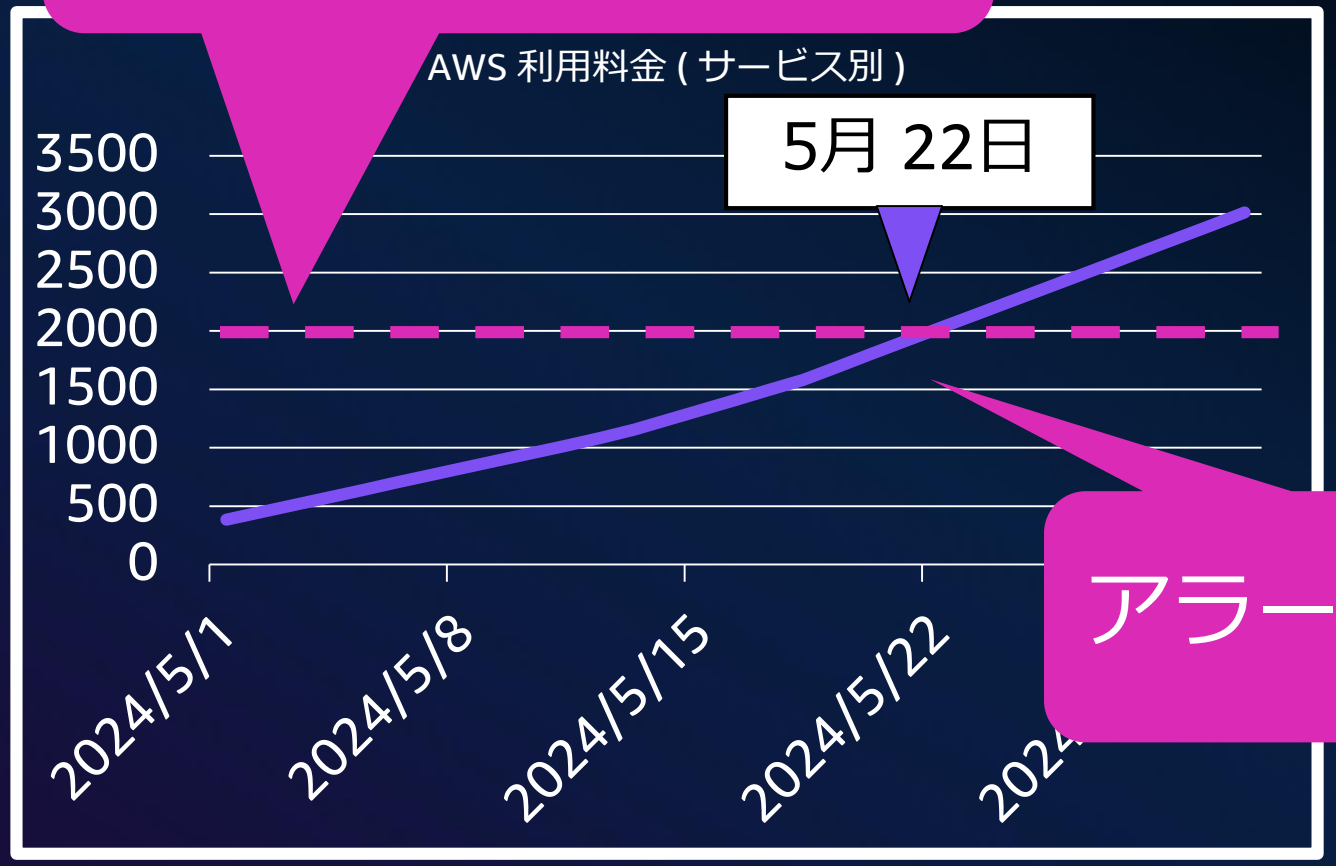


予算額の 80% でアラート通知

予算・予算アラートを設定すると こうなります

ミッション 3
計画・予測

予算 : $2,500
予算アラート : $2,000



アラート発報

これなら
安心!

経理
A さん

ミッション 3
計画・予測

異常検知アラートも設定しましょう
AWS Cost Anomaly Detection

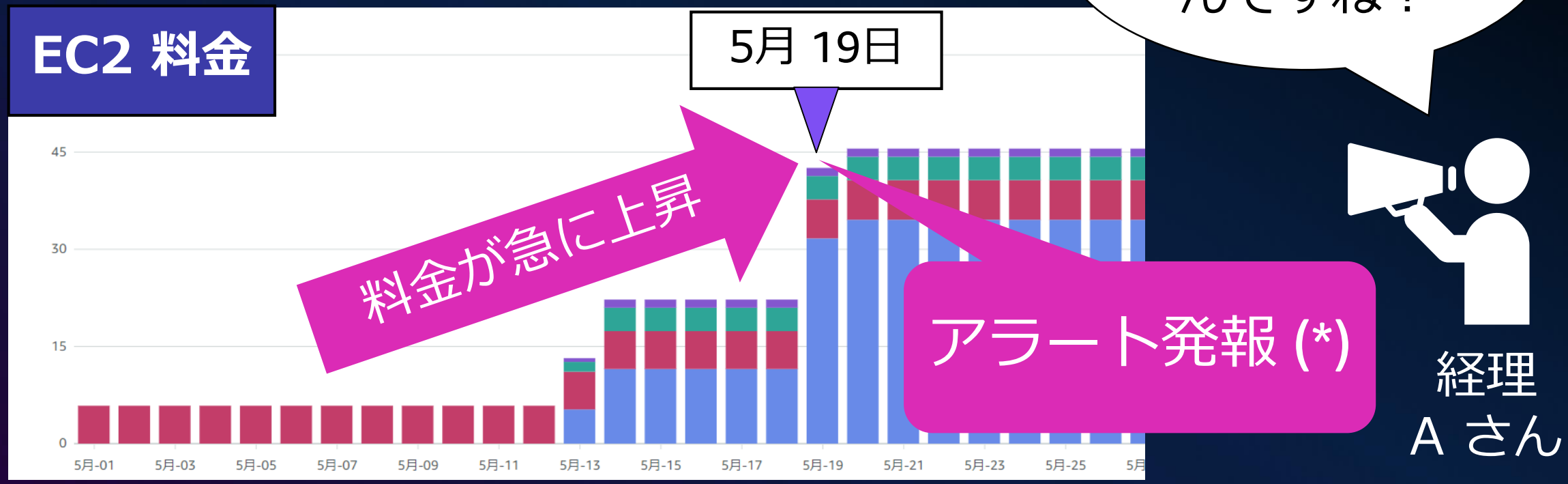
AWS サービスの利用状況を機械学習モデルによって学習して、
ベースラインを大きく逸脱した値を検知することができます。

モニターの設定

モニタータイプは AWS サービスがおすすめ！

異常検知アラートを設定すると こうなります

ミッション 3
計画・予測



サービス別の
異常がわかる
んですね！

経理
A さん

* AWS サービスの利用状況を機械学習モデルによって学習し、ベースラインを大きく逸脱した値を検知します。

ここまでのまとめ

ミッション 3
計画・予測

- **予算アラート**と**異常検知アラート**を設定しよう！

  - AWS Budgets を使って予算と予算アラートを作成し、利用料金全体や、プロジェクトごとの利用料金が計画と異なる推移をしたらすぐ検知できるよう監視しましょう
  - AWS Cost Anomaly Detection を使って、自動でサービス毎のコストの異常検知をしましょう

まとめ

まとめ

このセッションではユースケースを通して
CFM のフレームワークを使った
AWS のコスト最適化の進め方を
ご紹介しました。

まとめ

1. 可視化

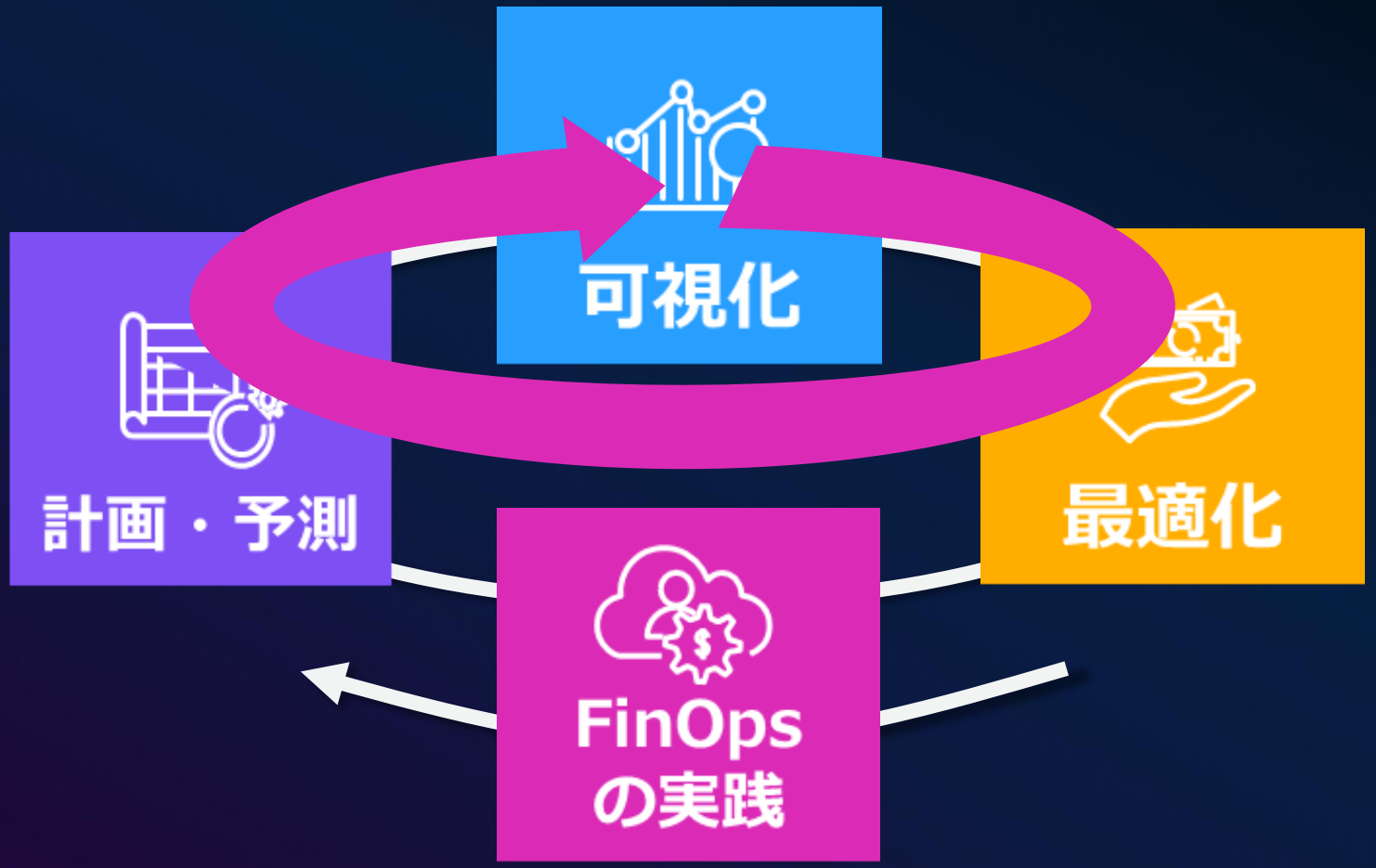
   - AWS Cost Explorer / AWS Cost and Usage Reports で分析
   - タグを使ったリソースの管理、タグ無リソースの検索・是正方法
   - データエクスポートと CUR2.0 で簡単にダッシュボードを作る
2. 最適化

   - Cost Optimization Hub でコスト最適化の機会を知る
3. 計画・予測

   - AWS Budgets を使って予算を立てる/アラートを設定する
   - AWS Cost Anomaly Detection を使ってコストの異常を検知する

CFM の「 FinOps の実践」に取り組みましょう！

可視化、最適化、計画・予測に継続的に取り組みましょう。



まとめ

コストを見るところから始めてみてはいかがでしょうか？
まずは AWS Cost Explorer にアクセスしてみてください！

Thank you!

石王　愛