

AWS Blackbelt Container

メトリクス入門 CloudWatch Container Insights

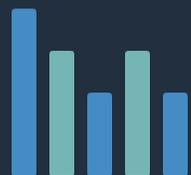
2021-09

分散システムを「観測可能」にする三つの柱



ログ

- システム内で発生したイベント情報
- 各イベントが独立したレコードとして記録される
- タイムスタンプを伴うことが多い
- e.g. アクセスログ、エラー情報、...



メトリクス

- ある時点のなんらかのシステム状態を表現する数値情報
- 一定間隔ごとの時系列データとして記録される
- 1つ以上のディメンジョン/ラベルで識別される
- e.g. CPU 使用率、エラー率、ストレージ残容量

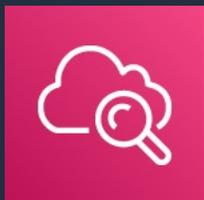


トレース

- 1つのトランザクションを複数システムで構成するフロー情報
- トランザクションごとにユニークな識別子をもって記録される
- システム間のやりとりに関するメタ情報付与も
- e.g. ある HTTP リクエストの受け取りからレスポンスまで
- 「リクエスト・パス」

分散システムを「観測可能」にする三つの柱

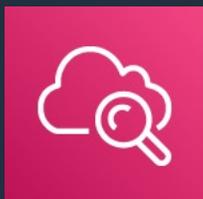
Amazon
CloudWatch



ログ

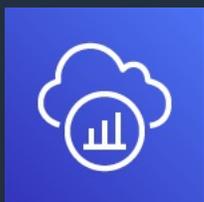
- システム内で発生したイベント情報
- 各イベントが独立したレコードとして記録される
- タイムスタンプを伴うことが多い
- e.g. アクセスログ、エラー情報、...

Amazon
CloudWatch



メトリクス

- ある時点のなんらかのシステム状態を表現する数値情報
- 一定間隔ごとの時系列データとして記録される
- 1つ以上のディメンジョン/ラベルで識別される
- e.g. CPU 使用率、エラー率、ストレージ残容量、...



トレース

- 1つのトランザクションを複数システムで構成するフロー情報
- トランザクションごとにユニークな識別子をもって記録される
- システム間のやりとりに関するメタ情報付与も
- e.g. ある HTTP リクエストの受け取りからレスポンスまで
- 「リクエスト・パス」

このセッションで扱うこと

- AWS CloudWatch Container Insights 概要
- AWS CloudWatch Container Insights の準備
- AWS CloudWatch Container Insights の利用例

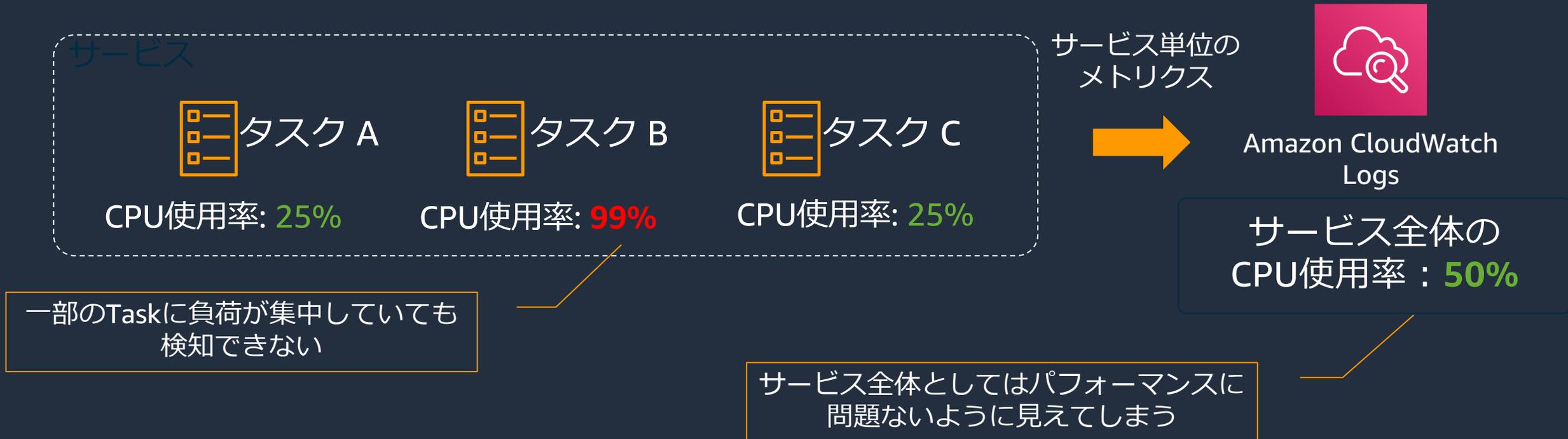
(注) AWS CloudWatch Logs Insights および AWS X-Ray とも統合されていますが、メトリクス中心に扱います

Container Insights 概要

Container Insights登場前のコンテナモニタリングの例

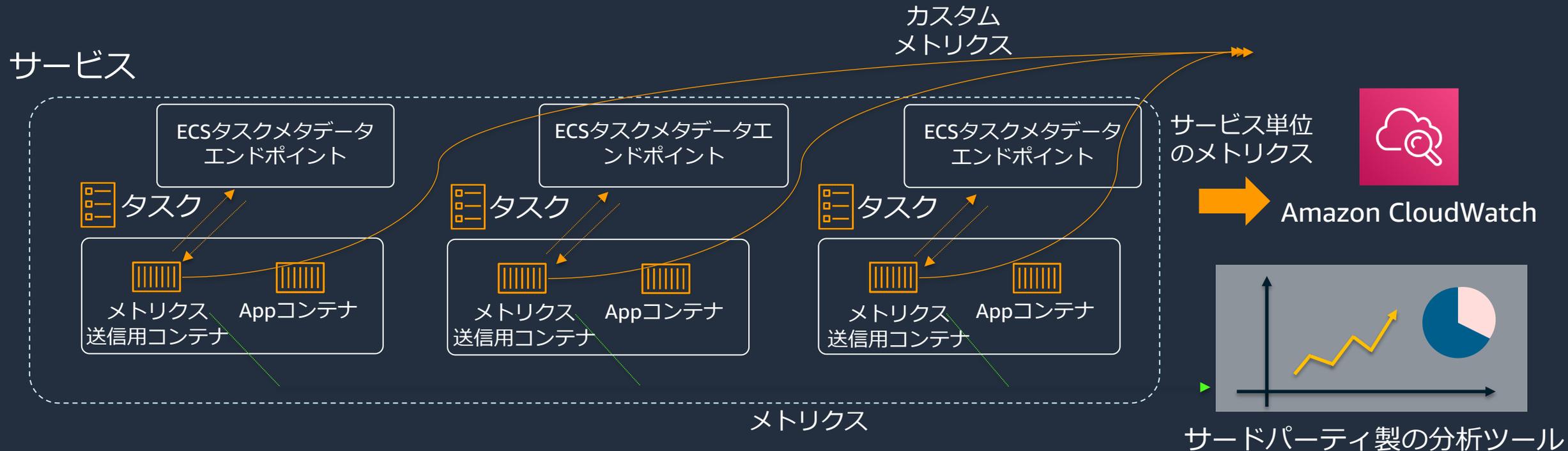
- CloudWatchのデフォルトの設定ではタスク、コンテナ単位のメトリクスが取得できなかった
- タスク、コンテナ単位レベルの問題を検知する際に、より詳細なメトリクス情報が必要となる場合があった

タスク、コンテナ単位レベルの問題の検知が難しい例)



Container Insights登場前のコンテナモニタリングの例

- 各コンテナからECSタスクメタデータエンドポイントを呼び出し、カスタムメトリクスとしてCloudWatchにメトリクスを送信する
- サードパーティー製のメトリクス送信用のコンテナをサイドカーとして配置する



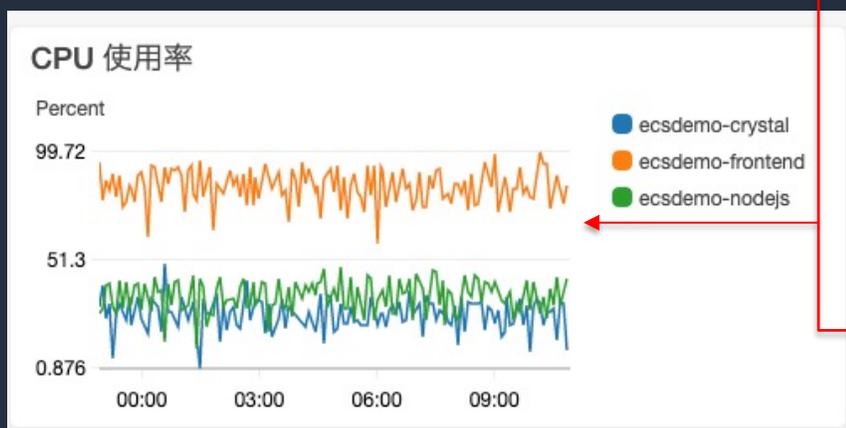
タスクやコンテナレベルのメトリクスを取得するには何かしらの工夫が必要だった

Amazon CloudWatch Container Insightsとは

- コンテナ化されたアプリケーションのメトリクスとログを収集、集計、要約できるCloudWatchの機能の一つ
- CloudWatchにてタスク、コンテナレベルでのモニタリングが可能
- Container Insights が収集するメトリクスは自動的に作成されるダッシュボードに集約され、より鋭い洞察を行うことが可能
- AWSが提供するコンテナオーケストレーションツールであるAmazon ECSや、Amazon EKS、および Amazon EC2 の Kubernetes プラットフォームでご利用可能

Amazon CloudWatch Container Insightsの概要

- Amazon CloudWatchと統合された、タスクやコンテナレベルでメトリクスやログを取得することが可能



Task performance (9)

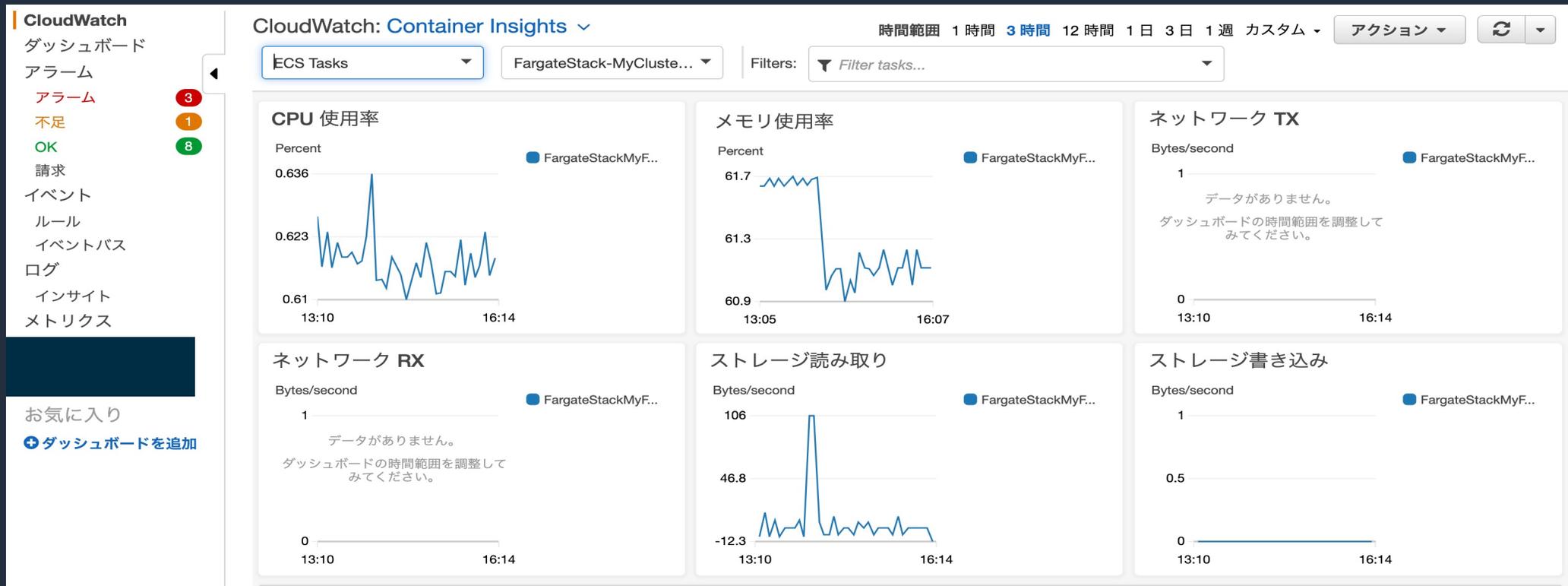
Q ecsdemo-frontend X

< 1

	Task definition	サービス	平均 CPU (%)	平均メモリ (%)
<input type="checkbox"/>	ecsdemo-frontend	ecsdemo-frontend	1.6258	16.0156
<input type="checkbox"/>	ecsdemo-frontend	ecsdemo-frontend	99.574	96.0156
<input type="checkbox"/>	ecsdemo-frontend	ecsdemo-frontend	1.5725	16.0156

Amazon CloudWatch Container Insightsの概要

- Container Insights が収集するメトリクスは自動的にダッシュボードに集約され、可視化を行うことが可能

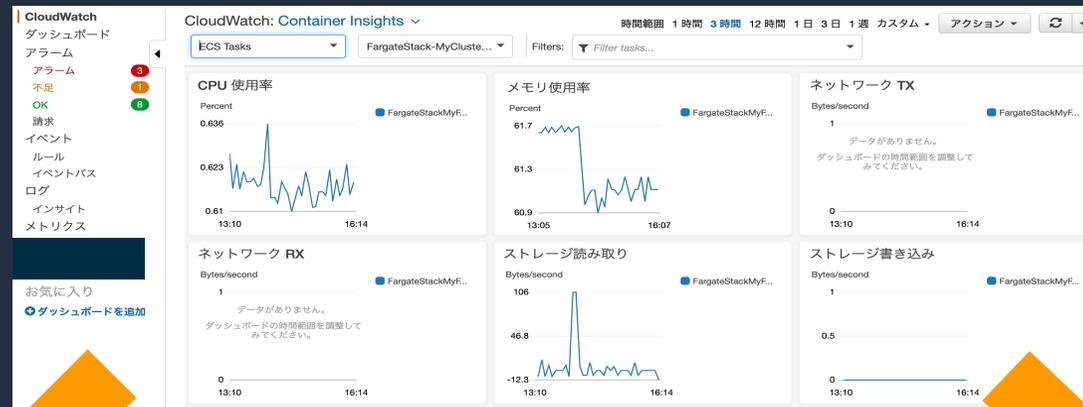


Amazon CloudWatch Container Insightsの概要

- CloudWatch Logs InsightsやX-Rayとも統合されている
- Container Insightsのダッシュボードを起点により詳細な分析が可能

CloudWatch Logs Insightsの要件の例

- グラフのより詳細な値を見たい
- ログに対し分析クエリを発行したい



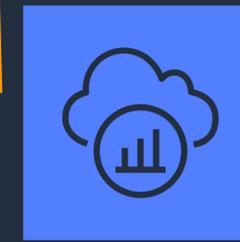
X-Ray を使う要件の例

- タスク間の通信をトレースしたい



Amazon CloudWatch
Logs Insights

Container Insights



AWS X-Ray

Container Insightsの使い方

Container Insightsを有効にする ～ ECS編 ～

新規のクラスタに対し有効化する

- マネジメントコンソールの「**ステップ 2: クラスターの設定**」画面にて「**Container Insights を有効にする**」にチェックを入れる

CloudWatch Container Insights

CloudWatch Container Insights はコンテナ化されたアプリケーションとマイクロサービスのモニタリングとトラブルシューティングのためのソリューションです。CPU、メモリ、ディスク、ネットワークなどの計算使用率、およびクラスターの問題を切り分けて迅速に解決するのに役立つ、コンテナ再始動の失敗などの診断情報を情報を収集、集約、および要約します。

[🔗 詳細はこちら](#)

CloudWatch Container Insights Container Insights を有効にする

Container Insightsを有効にする ～ ECS編 ～

- Container Insightsのオプトインについて

- Container Insightsは以下の単位でデフォルトで有効化することが可能

- アカウント全体
- IAMロール
- IAMユーザ



画面下部へ移動



適用範囲でContainer Insightsがデフォルトで有効になる

Container Insightsを有効にする ～ ECS編 ～

- Container Insightsが有効かどうかは、ECSのクラスター画面で「Container Insights」に緑色のチェックが入っているかどうかで確認できる

The screenshot shows the AWS ECS console interface for a cluster named 'container-demo'. At the top, under 'CloudWatch モニタリング', the 'Container Insights' status is shown as 'ON' with a green checkmark, highlighted by a red box. Below this, the cluster is identified as 'FARGATE'. The dashboard displays the following metrics:

Category	Service	Running Tasks	Reserved Tasks	CPU Utilization	Memory Utilization	Container Instances
FARGATE	3	9	0	-	-	-
EC2	0	0	0	No data	No data	0

Container Insightsを有効にする ～ EKS編 ～

Container Insightsを有効にするための前提条件を確認

- Container Insightsが利用可能なリージョンでクラスタが起動している
- kubectl がインストール、権限設定が完了しており、 kubectl apply が実行可能な状態になっている
- IAM のアクセス許可
 - ワーカーノードの IAM ロールにポリシーをアタッチor
 - クラスタでサービスアカウントの IAM ロールを使用し、このロールにポリシーをアタッチ

■Amazon EKS と Kubernetes で Container Insights をセットアップする 前提条件の確認

https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/Container-Insights-prerequisites.html

© 2021, Amazon Web Services, Inc. or its Affiliates.



Container Insightsを有効にする ～ EKS編 ～

Container Insightsを有効にする手順

- 必須手順
 - CloudWatch にメトリクスを送信する CloudWatch エージェントまたは、または AWS Distro for OpenTelemetry を DaemonSet としてセットアップ
 - CloudWatch Logs にログを送信する DaemonSet として FluentD/Fluent Bit をセットアップ
- オプション手順
 - Amazon EKS コントロールプレーンのログ記録を有効にする(コントロールプレーンのログをCloudWatch Logsで監査する必要がある場合)
 - StatsD エンドポイントとして CloudWatch エージェントをセットアップする(StatsDを使用している場合)
 - App Mesh Envoy アクセスログを有効にする

■Amazon EKS と Kubernetes で Container Insights をセットアップする

https://docs.aws.amazon.com/ja_jp/AmazonCloudWatch/latest/monitoring/deploy-container-insights-EKS.html

© 2021, Amazon Web Services, Inc. or its Affiliates.



ユースケース

ECSでタスクに配置するコンテナのリソースを適切なサイズにチューニングしたい

ユースケースの詳細

- コンテナごとの適切なリソースの配分は非常に重要
- 適切でないリソース配分はパフォーマンスの劣化や、過剰なリソースの確保、リソース不足によるプロセスの停止につながる可能性がある

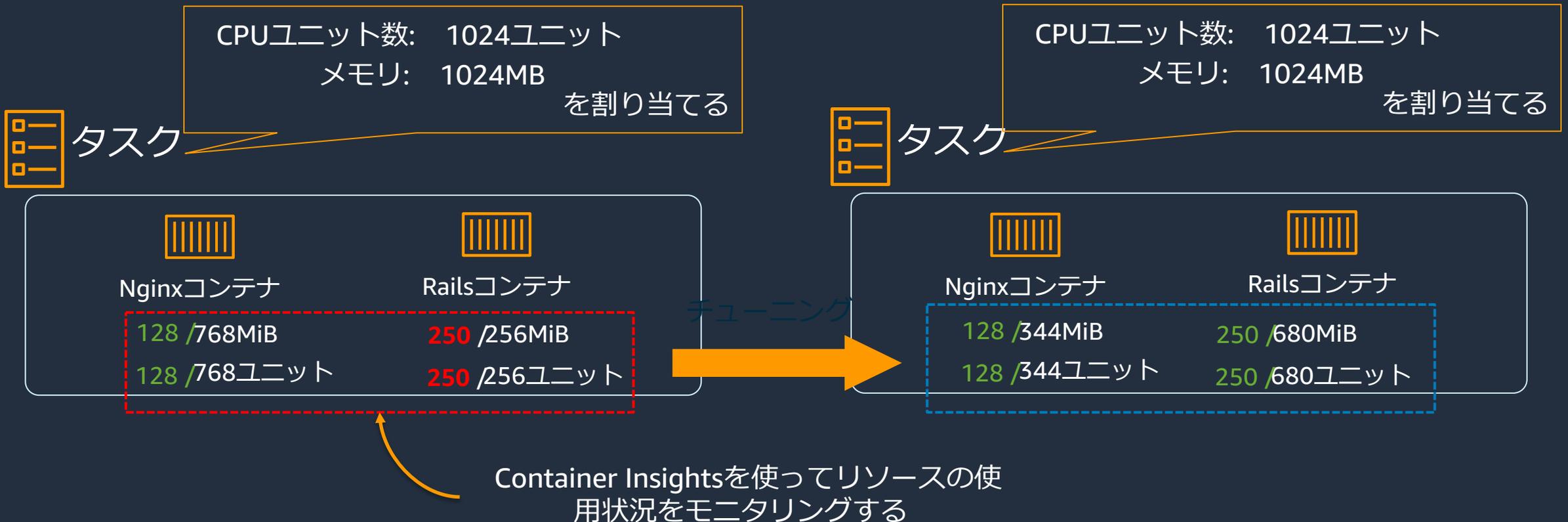
例) リソースが適切に配分できていない例

- タスク全体ではリソースに余力があるが、片方のコンテナでリソースが足りていない
- どちらのコンテナにも余力があるが、タスク全体でリソースを効率的に使えていない状態



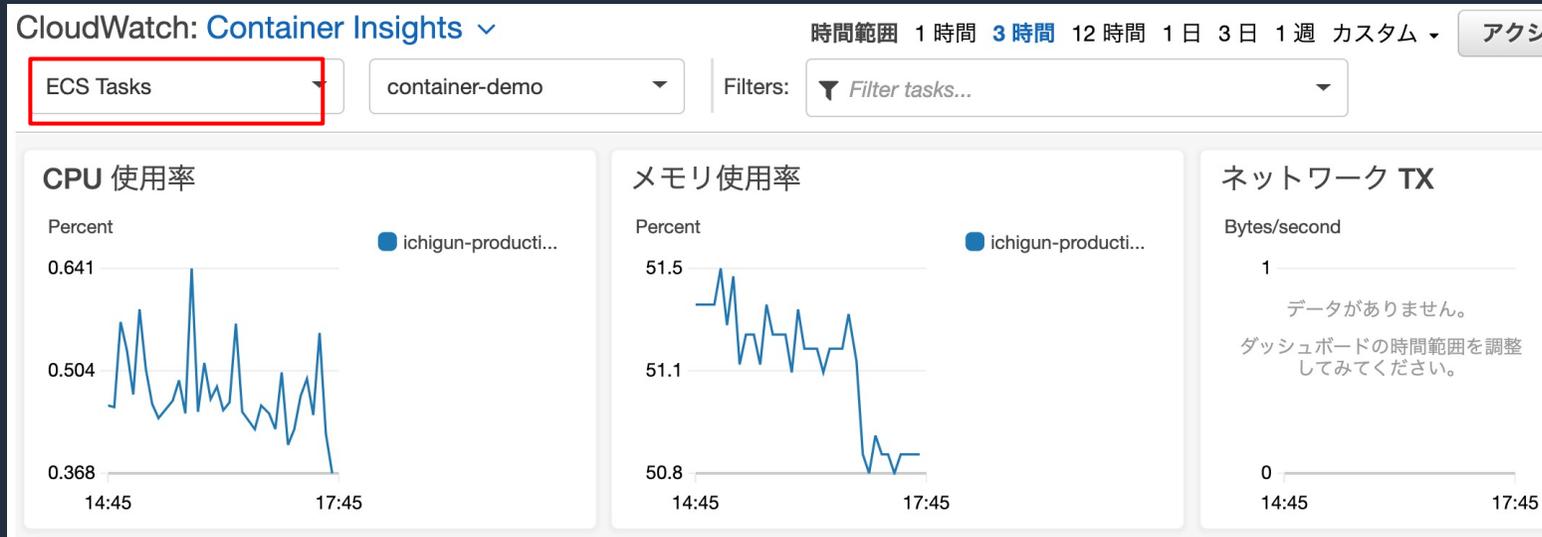
Container Insightsを使って適切なリソース配分を行う

- Container Insightsはコンテナ単位のリソースモニタリングが可能であるため、各コンテナに適切なリソース配分が行われているかを確認できる
- 適切なリソース配分により、効率的かつ安全に運用できるリソースを割り当てる



ステップ1. Containerごとのメトリクスを確認する

- Container Insightsの自動ダッシュボードから「ECS Tasks」を選択し、Containerごとのパフォーマンスを確認する



片方のコンテナに負荷が偏っていることが判明した！

Container performance (4)

アクション

Filter task definitions, container names, tasks...

Task definition	コンテナ名	平均 CPU (%)	平均メモリ (%)
task-demo	container-a	89.1227	85.6548
task-demo	container-b	9.6542	11.6954

ステップ.2 Containerリソース配分を変更する

- ECSコンソールのタスク定義からタスク内のコンテナリソース配分を変更する

container-a : メモリ256MB/CPU 128ユニット
container-b : メモリ256MB/CPU 128ユニット

container-a : メモリ384MB/CPU 192ユニット
container-b : メモリ128MB/CPU 64 ユニット

タスクサイズ

タスクサイズにより、タスクの固定サイズを指定できます。Fargate 起動タイプを使用したタスクにはタスクサイズが必須で、EC2 起動タイプではオプションです。タスクサイズが設定されている場合、コンテナレベルのメモリ設定はオプションです。タスクサイズは Windows コンテナではサポートされません。

タスクメモリ (GB) 0.5GB
0.25 vCPU の有効なメモリ範囲: 0.5GB - 2GB。

タスク CPU (vCPU) 0.25 vCPU
0.5 GB メモリの有効な CPU: 0.25 vCPU

コンテナメモリの予約用のタスクメモリの最大割り当て

0 512 MiB

コンテナへの CPU の最大割り当て

0 256 CPU ユニット数

コンテナの定義

コンテナの追加

コンテナ名	イメージ	ハード/ソフトメモリ...	CPU ユニ...	GPU	Inference Accelerat...	基本	
container-a	888727018440.dkr.ec...	256/256	128			true	✕
container-b	888727018440.dkr.ec...	256/256	128			true	✕



タスクサイズ

タスクサイズにより、タスクの固定サイズを指定できます。Fargate 起動タイプを使用したタスクにはタスクサイズが必須で、EC2 起動タイプではオプションです。タスクサイズが設定されている場合、コンテナレベルのメモリ設定はオプションです。タスクサイズは Windows コンテナではサポートされません。

タスクメモリ (GB) 0.5GB
0.25 vCPU の有効なメモリ範囲: 0.5GB - 2GB。

タスク CPU (vCPU) 0.25 vCPU
0.5 GB メモリの有効な CPU: 0.25 vCPU

コンテナメモリの予約用のタスクメモリの最大割り当て

0 512 MiB

コンテナへの CPU の最大割り当て

0 256 CPU ユニット数

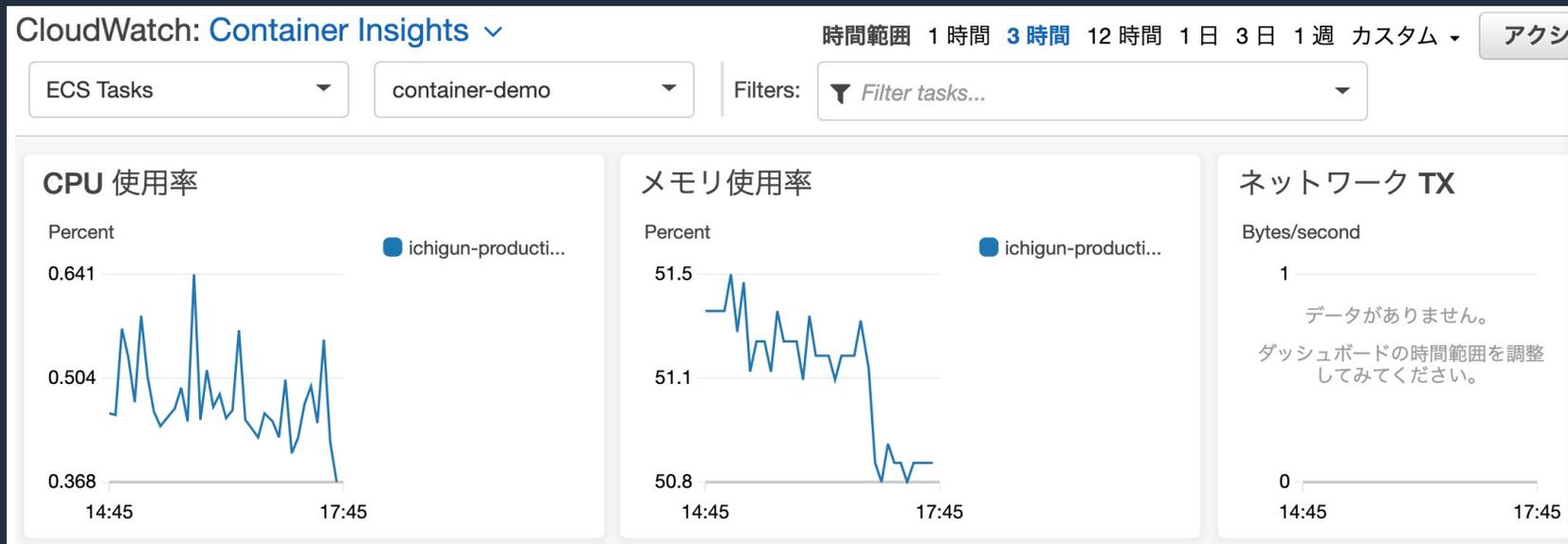
コンテナの定義

コンテナの追加

コンテナ名	イメージ	ハード/ソフトメモリ...	CPU ユニ...	GPU	Inference Accelerat...	基本	
container-a	888727018440.dkr.ec...	384/384	192			true	✕
container-b	888727018440.dkr.ec...	128/128	64			true	✕

ステップ3. 再びContainerごとのリソースを確認

- あらためて Container Insights からコンテナごとのリソース使用状況を確認する



The screenshot shows the 'Container performance' table with 4 containers. The table is filtered to show two containers, 'task-demo' in 'container-a' and 'task-demo' in 'container-b'. The columns '平均 CPU (%)' (Average CPU %) and '平均メモリ (%)' (Average Memory %) are highlighted with a red box. The values are 45.3721% and 50.2441% for container-a, and 42.9787% and 44.1687% for container-b.

Task definition	コンテナ名	平均 CPU (%)	平均メモリ (%)
task-demo	container-a	45.3721	50.2441
task-demo	container-b	42.9787	44.1687

適切なリソース割り当てができたことを確認！

このセッションで扱ったこと

- AWS CloudWatch Container Insights 概要
 - ✓ パフォーマンス変化、リソース使用率、運用状況把握
- AWS CloudWatch Container Insights の準備
 - ✓ ECS ではクラスタ作成時に選ぶだけ（あとからでも可）
 - ✓ EKS では DaemonSet として CloudWatch エージェント
- AWS CloudWatch Container Insights の利用例
 - ✓ コンテナ単位のメトリクスを確認して最適な配置