

# HinemosでAWS運用を実現するメリット

NTTデータ先端技術株式会社

**NTT DATA**  
Trusted Global Innovator



# INDEX

1. 運用管理における課題とFAQ AWS編
2. 運用管理における課題とFAQ パブリッククラウド編
3. 運用管理における課題とFAQ ハイブリッドクラウド編
4. appendix

# 1

## 運用管理における課題とFAQ AWS編

## AWS環境の監視の課題

### モニタリングサービスと連携したシステム監視

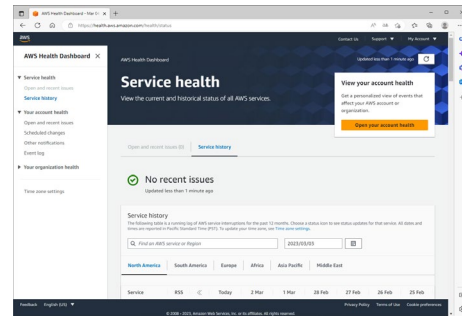
AWS環境の監視にはAmazon CloudWatchは必要です。しかし、それだけでは実際の運用要件を満たせず、何かしらの監視ツール・サービスとの連携/作りこみが必要になります。

CloudWatchの必要性	CloudWatchの課題
<ul style="list-style-type: none"> <li>CloudWatch経由で取得が必要なリソース値</li> <li>AWSの各種PaaS</li> <li>EBS(ネットワークストレージ)</li> </ul>	<ul style="list-style-type: none"> <li>保存期間が最大でも15ヶ月</li> <li>業務カレンダーとして連携した監視/通知抑制</li> <li>業務層の監視(Webシナリオ監視等)</li> <li>OS上のリソース値は取得不可(ディスク使用率等)</li> </ul>

### AWSというプラットフォームの監視

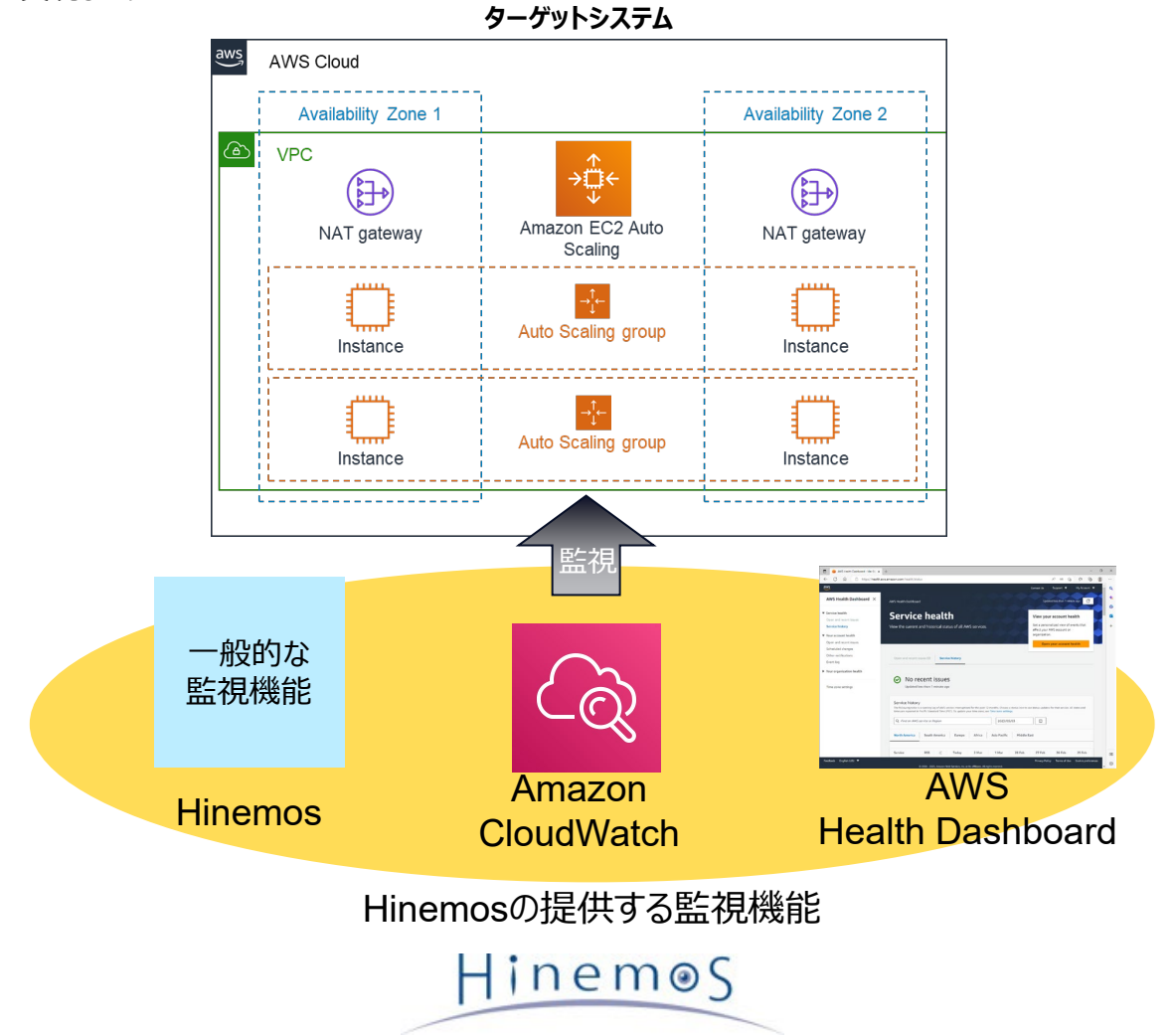
AWS環境はクラウドサービスとして提供されたインフラですが、このインフラ・サービスが正常に動作していることを監視する必要があります。これはAWSの外部から監視する必要があり、AWSの提供するAWS Service Health DashboardからRSS経由で独自で監視する必要があります。

AWS Health Dashboard



## HinemosによるAWS環境の統合管理

Hinemosを導入することで、ユーザが作りこみをする必要なく、AWSの監視に必要な機能の連携を実現します。

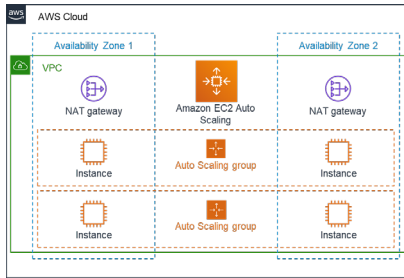


## AWS環境の運用自動化の課題

### 環境変更への追従

柔軟なリソースコントロールができるAWS環境においては、その環境変更による運用コストの低減が課題になります

- ・EC2インスタンスの検出(初期構築/オートスケーリング)
- ・リージョン、アベイラビリティゾーンへのマッピング(ロケーション)
- ・VPC、サブネットへのマッピング(ネットワーク構成)



運用対象の構成把握に  
運用コストが掛かる



運用者

### 監視・ジョブの運用自動化

自動検出した対象について、対象別に監視・ジョブを自動的に開始する仕組みが必要です。これを実現するには次の3ステップの自動化が必要です。

① 自動検出

オートスケールや計画的な機器増設による仮想マシンの変更は、オペレーションミス为了避免、自動検出の仕組みが必要です。

② 識別

新規仮想マシンを検出した際に、何の属性(OSやミドルウェア)を持ったものかを識別できる仕組みが必要です。

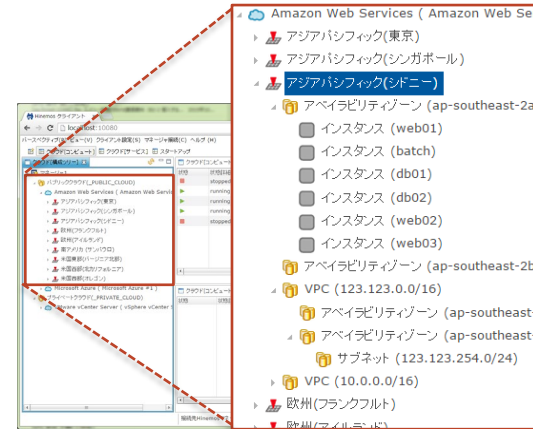
③ 監視・ジョブ開始

仮想マシンを識別した後によく属性にあった監視やジョブを設定することができます。

## Hinemosによる運用自動化

HinemosにAWSへの接続情報(アクセスキー・シークレットキー)を登録すると、AWS上のEC2インスタンス情報を取得し、リージョン、アベイラビリティゾーン、VPC、サブネットの単位で自動グルーピングします。運用開始後も自動追尾して、環境変更時に伴う運用操作は必要ありません。

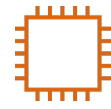
自動構成管理



EC2インスタンスにタグ情報を付与することでHinemosは自動的にその属性を識別し、属性に合わせたグルーピング(スコープ割当て)を行います。グループ別の監視やジョブ定義を事前定義することで、環境変更に影響なく監視・ジョブの運用自動化が可能です。

タグを使った自動識別

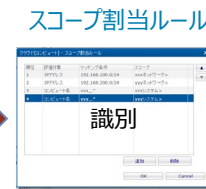
EC2インスタンス



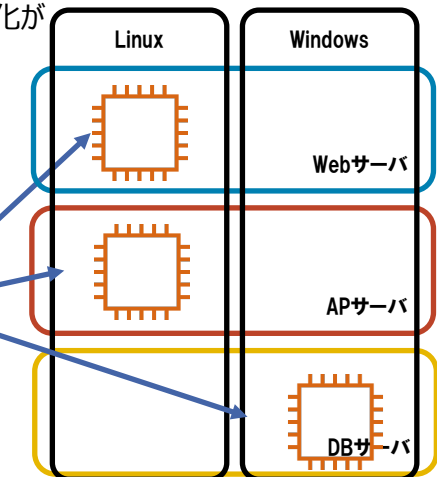
+ タグ情報



検出



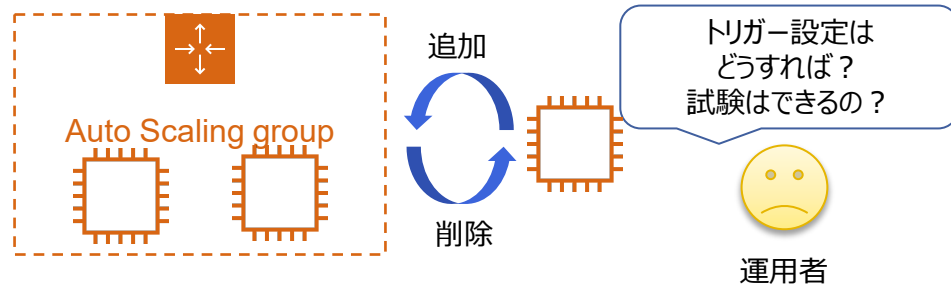
スコープ割当てルール



## AWSのリソース制御の課題

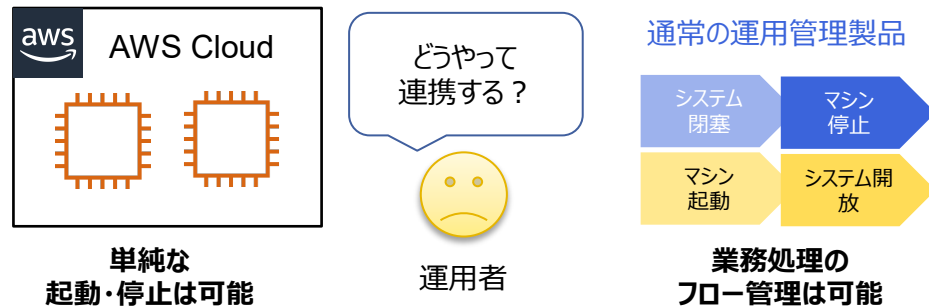
### AWS Auto Scalingの課題

AWS Auto Scaling は非常に有用なソリューションですが、スケールイン・スケールアウトのトリガーの条件設定が難しく、予期せぬタイミングでトリガーが発動するかもしれません。また、条件によっては、試験ができないケースがあります。そのため、単純にEC2インスタンスを消さずに、必要に応じて起動・停止だけするケースも多くあります。



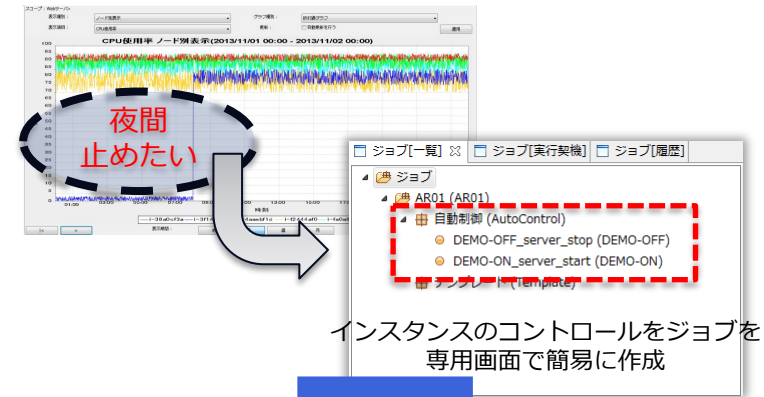
### リソース制御と業務連携

EC2インスタンスの起動・停止のコントロールは単純に実施すればよいものではなく、業務・システムと連動して行われるべきです。そのため、システム閉塞・開放といった簡単な処理でも、業務フローと連携する機能が必要になります。

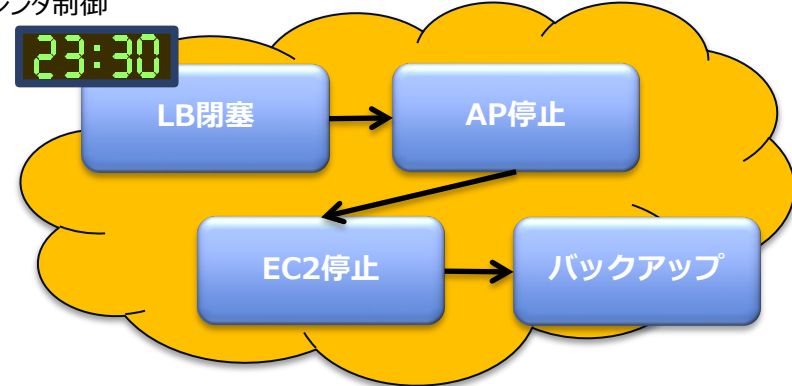


## リソース制御と業務連携

HinemosではEC2インスタンスの起動・停止処理をGUIから簡単にジョブとして登録する機能があります。これにより、システム閉塞・開放といった、リソース制御を伴うことでコスト削減を実現するクラウドのメリットについて、Hinemos単独で作り込をすることなく実現することができます。



- ・スケジュール起動
- ・カレンダー制御



AWSの(主にEC2の)使用料金削減のポイントはEC2の起動時間

$$\text{EC2使用料金の仕組み(概算)} = \text{インスタンス数} \times \text{インスタンスタイプ} \times \text{時間}$$

AWS使用料金の削減にHinemosを使用するメリット

- ①簡易なAWSリソースコントロール機能  
簡易なGUI操作でEC2の起動停止処理をジョブフローに組み込めます
- ②業務閉塞・開放と連動したAWSリソースコントロールのスケジュール管理  
業務処理連動のジョブフローを要件に合わせて実行することができます
- ③業務カレンダーを使用した停止中の監視抑止(監視機能との連動)  
業務停止中には業務の監視を停止するなど、業務カレンダーの連携が簡単に行えます

EC2インスタンス起動時間制限によるAWS料金削減効果

・土日を停止できれば

70%

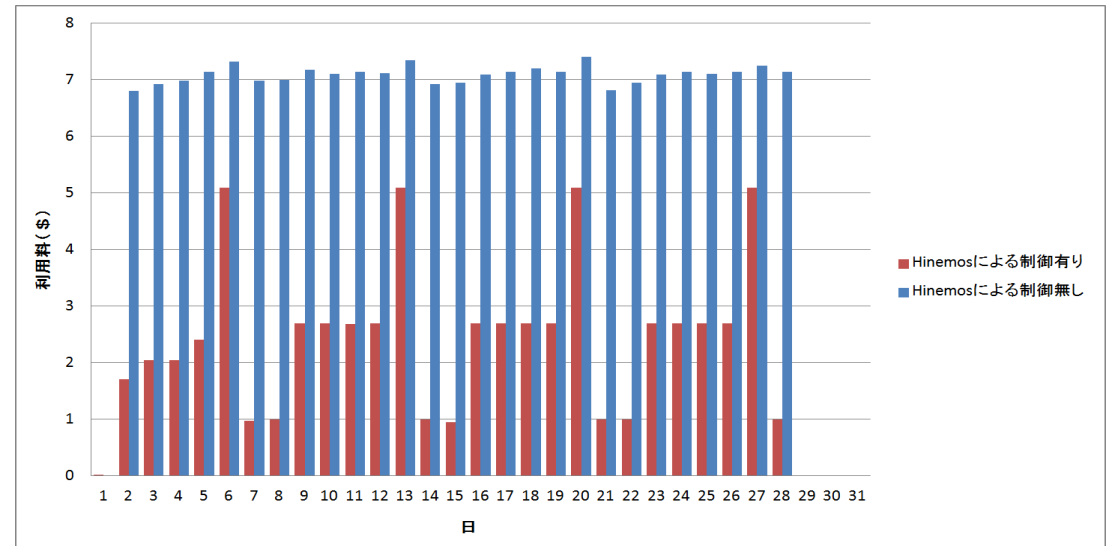
月-金  
→ 5/7

・さらに営業時間を8時~24時に限定すれば

50%

月-金&8:00-24:00  
→ (5/7) x (16/24)

Hinemosの計画停止コントロールによる料金削減検証

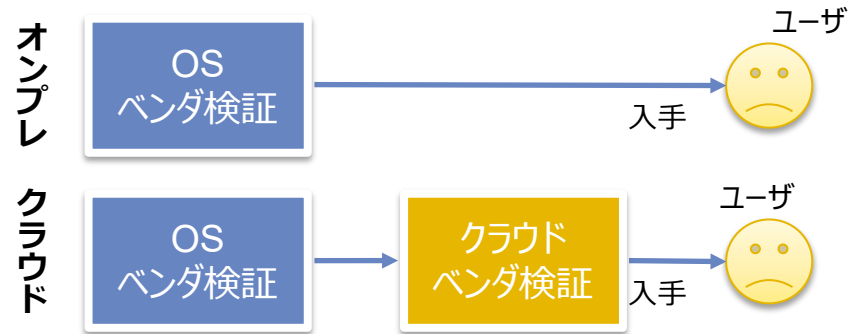


# OSサポートレベル

## OSサポートレベルの課題

### OSのセキュリティ対応

Red Hat Enterprise Linux等の商用OSについては、OSのアップデートやセキュリティパッチの入手が、オンプレミス環境より、クラウドの方がワンテンポ遅くなります。



### Javaサポート

現在でも多くのアプリケーション・ミドルウェアにJavaが必要です。独自のOSが登場すると、そのOS上でJava Runtimeをどうするか、が課題になります。Red Hat Enterprise Linux上のOpenJDKはRed Hatがディストリビュータとして公開・サポートするJavaです。

OS	Java	Javaディストリビュータ	サポート期限
RHEL7	OpenJDK8	Red Hat	2023年6月
独自OS	?	?	?

## Hinemosの動作対応環境

Hinemosマネージャ、Hinemosエージェント共に、AWS上でOLTが提供されるOSのAmazon Linux 2及び、OpenJDK互換のAmazon Correttoに対応しています。これにより、AWS上でセキュアで安定した環境での運用管理が実現できます。

Amazon Linux 2023にも  
順次対応予定

### Amazon Linux 2対応

- Red Hat Enterprise Linux 7ベース(2019.4)
- 長期サポート
- AWS上の他、VMwareやDockerイメージ公開
- EC2に最適化
- セキュリティアップデート

### Amazon Corretto対応

- 2019.4正式リリース
- Amazonによるサポート
- 無料
- マルチプラットフォームサポート

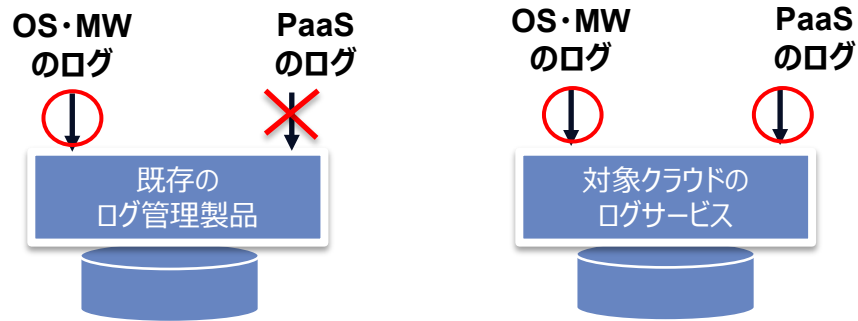


# 監視 クラウドのログは収集と監視の統合管理が必要

## AWSのログ管理の課題

### ログ収集の統合管理

パブリッククラウドの場合、WindowsやLinuxの仮想マシンだけでなくロードバランサといったPaaSがあります。PaaSのログを収集監視しようとするログを集めてくることだけを前提としたログ監視製品では対応できず、そのクラウドのログサービスが必要になります。



### ログ収集の統合管理

しかし、クラウドのログサービスにログを収集・集約したとしても、ログ監視は複雑なフィルタ条件や、業務カレンダーの連携、異常検知後の通知・自動化の連携が重要ですが、一般的な運用管理製品がもつレベルの機能まではクラウド側に用意されていません。

#### ログ監視の要素

##### フィルタ条件の集約

- ・決まった記述ルール (正規表現)
- ・順序性/判定有無

##### 業務カレンダーの連携

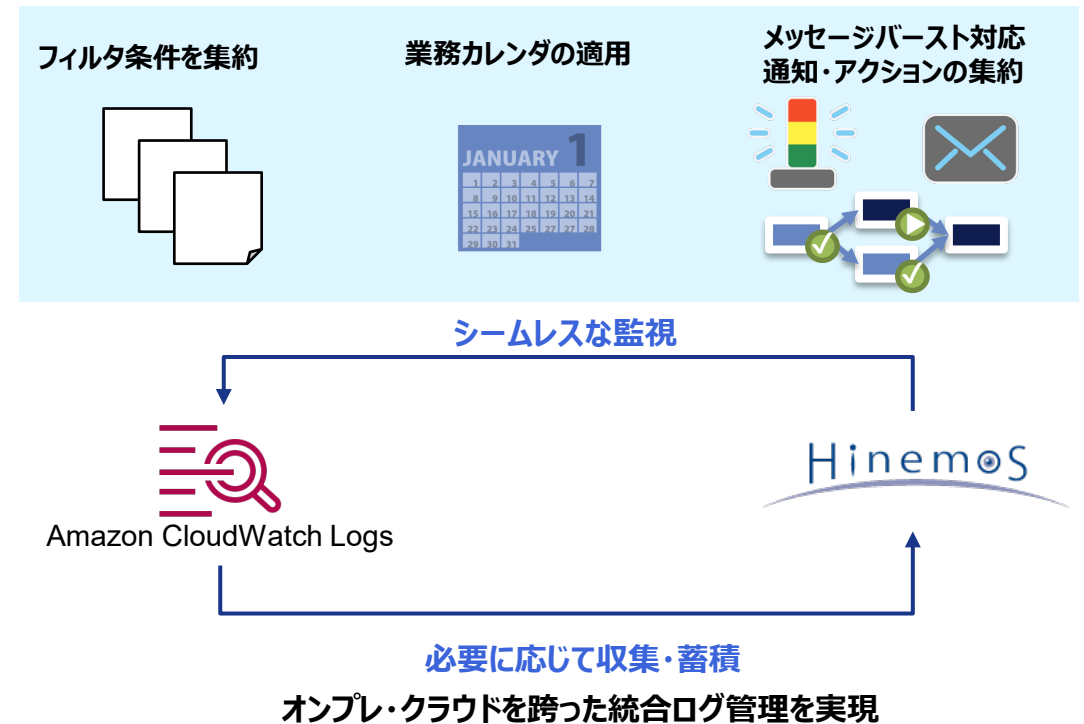
- ・メンテナンス時間の監視無効化

##### 高度な通知や自動化の連携

- ・メッセージバーストを防ぐ抑制機能
- ・通知・自動化先の集約

## Hinemosによるログ統合管理

Hinemosではクラウドのログサービスにあるログを対象に監視が行えます。これにより、他にHinemosで監視しているログファイル・メッセージに対するフィルタ条件や、業務カレンダーによる制御、通知抑制やジョブフロー連携も簡易に可能です。また、Hinemosをログ収集エンジンとして、ログサービスから取得して蓄積管理が、可能になります。



- 注1) 本機能を利用するにはクラウドのログサービスのログを中継するHinemosエージェントの導入が必要です
- 注2) Hinemosが蓄積したイベント・性能実績などを収集蓄積機能により fluentdを介してクラウドのログサービスに転送する事も可能です

# 自動化 運用自動化には双方向の連携が必要

## クラウド環境との双方向連携の課題

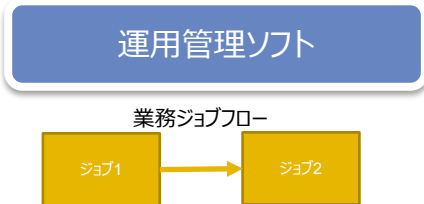
### クラウド内の連携

用途に応じて、クラウド導入においてクラウド側に構築した自動化の仕組みと、運用管理ソフト上にある自動化の仕組みの両方が存在するケースが多々あります。クラウド移行のケースの場合は、主に既存のジョブ定義になります。これらの、相互連携が全体的な自動化・効率化に重要になります。

#### クラウド側の自動化

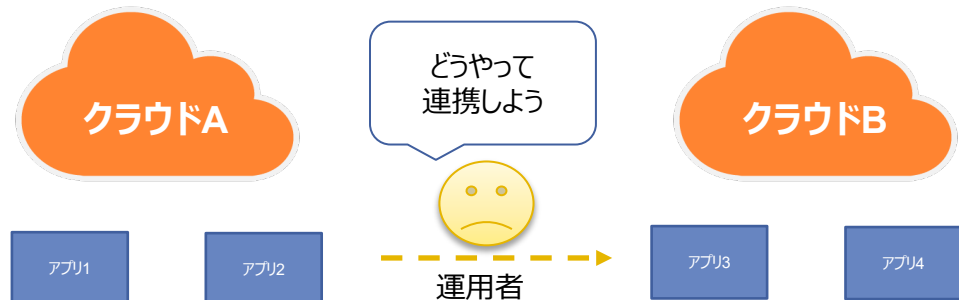
- ・PaaSのアプリケーション
- ・サーバレスのアプリケーション

#### 運用管理ソフト側の自動化



### マルチクラウドの連携

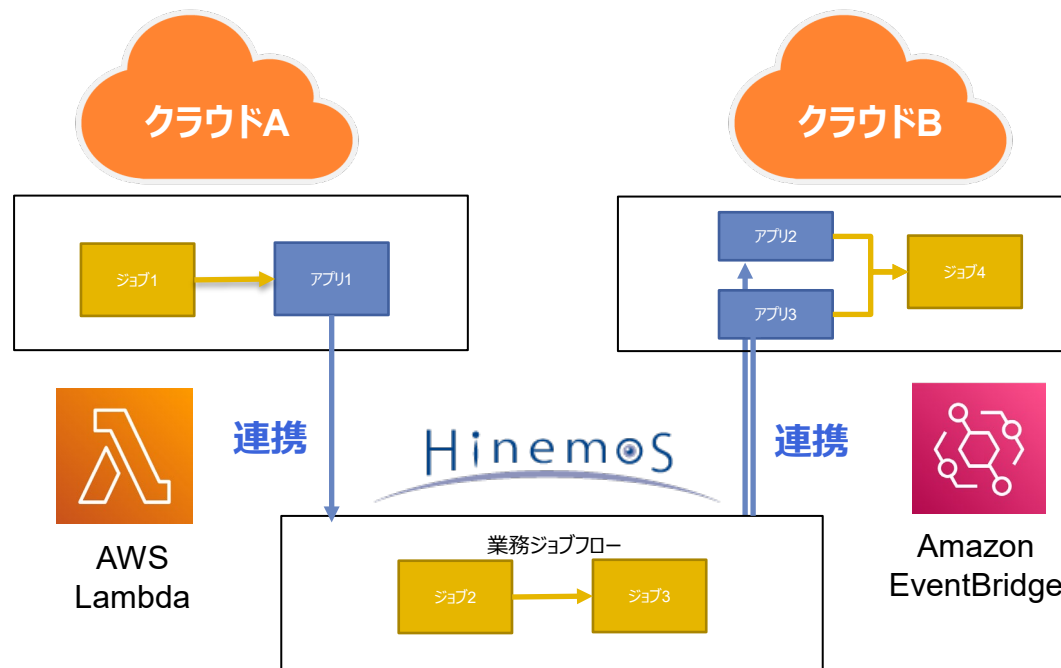
単独のクラウドだけではなく、マルチクラウドで組んだシステム間の連動をどのように実現するかも大きな課題です。



## Hinemosによる双方向連携

Hinemosはクラウド上で発生した汎用的なイベントを受け取り、指定のジョブフローを起動する事ができます。また、Hinemosより監視や業務ジョブの中から、クラウド側に汎用的な通知メッセージを送信し、指定の処理を起動する事ができます。

### マルチクラウドに対応した双方向連携

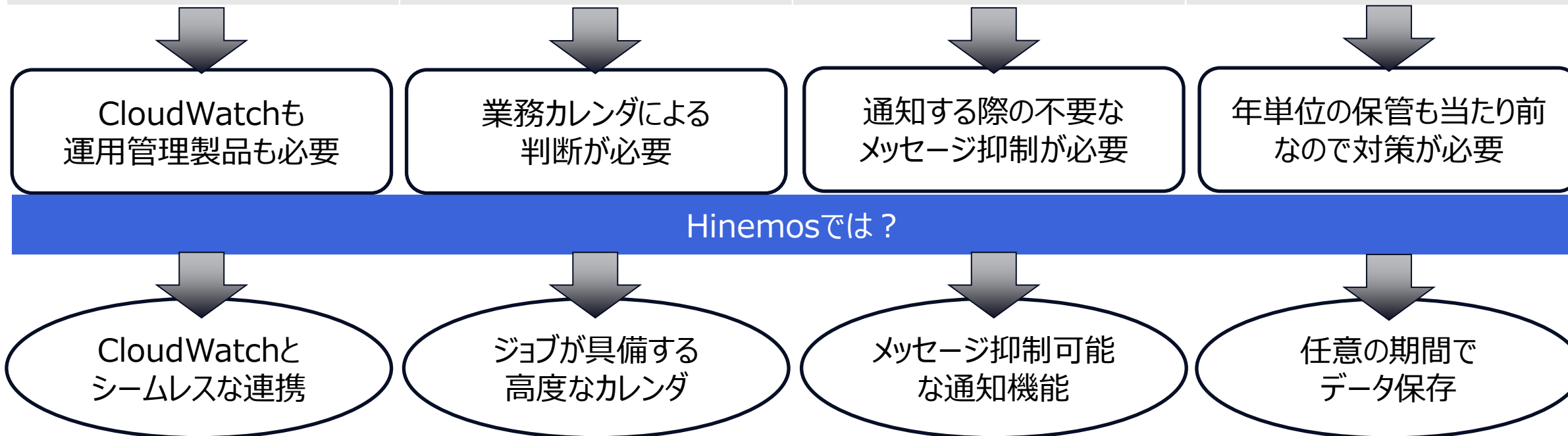


### 通知の集約とクラウド間のブリッジによりシームレスな自動化を実現

注1) AWS Lambdaからの連携は、カスタムトラップ監視設定のサンプルを提供

# FAQ CloudWatchがあるのにHinemosは必要？

性能値のカバー範囲	営業日・営業時間の判断	通知の抑制	データの保存期間
<b>CloudWatchは、</b> ・PaaSのため必須 ・OS上の性能値は苦手 (ディスク使用率が取れない)	<b>CloudWatchは、</b> ・性能値の取得可否はリソースが起動してるか否か	<b>CloudWatchは、</b> ・異常と判断する閾値等の指定のみで通知や抑制は別の機構	<b>CloudWatchは、</b> ・1分毎のデータ→15日間 ・5分毎のデータ→63日間 ・1時間毎のデータ→455日間
<b>一般の運用管理製品では、</b> ・OS上の性能値は得意 ・サービス監視も得意	<b>一般の運用管理製品では、</b> ・営業日・営業時間に性能値が取れていないと異常と判断可能	<b>一般の運用管理製品では、</b> ・瞬間的な異常や、バーストして発生する異常通知を抑制可能	<b>一般の運用管理製品では、</b> ・ディスク容量次第で設定可能



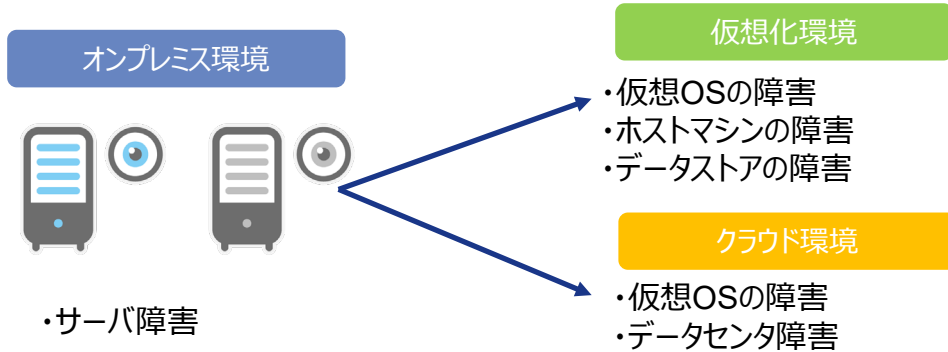
# 2

## 運用管理における課題とFAQ パブリッククラウド編

## 運用管理の可用性の課題

### 可用性の要件の違い

オンプレミス環境では物理サーバの障害を考慮が必要でしたが、仮想化環境やパブリッククラウド環境では、考えるべき要件が異なります。その条件により、運用管理製品の対応有無を確認するというフェーズが必要になります。



### クラスタ構成の様々な制約

オンプレミス環境で構築してきたクラスタ構成は、パブリッククラウド上で同様な構成を取ることが非常に難しく、クラウド毎で提示される「ベストプラクティス」を実装する必要があります。

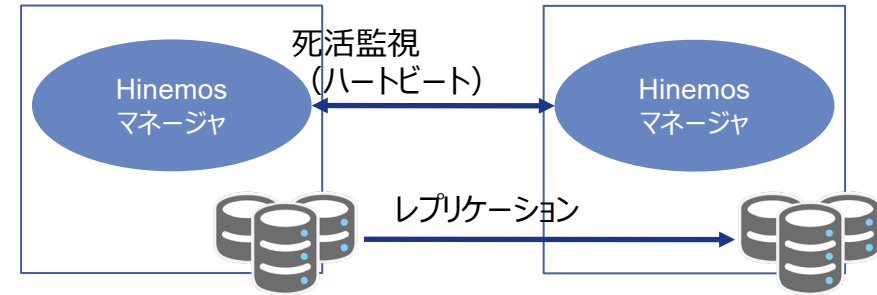


※参考情報（クラウドにおけるクラスタ構成の様々な制約）に詳細を記載

## Hinemosによるクラスタ機能

Hinemosでソフトウェアとして可用性構成を実現します（ミッションクリティカル機能）

### Hinemosミッションクリティカル機能



**クラスタリングソフトや共有ディスクは不要**

#### 環境非依存/FIP不要

オンプレミス、仮想化、クラウド環境で同じ構成で可用性構成が組めます。構成がシンプルのため、設計・構築時のSE/CEコストの削減が可能です。FIPを使用せず可用性構成を組めます。

#### ワンストップ保守

ソフトウェアで実現する可用性構成のため、障害発生時の対応がHinemosサポートに投げるだけ、のワンストップ保守が実現できます。

#### ロストなしの監視

クラスタミドルを使用しない独自の構成により、syslogやsnmptrapもロストなしで監視ができます。

#### ジョブ管理の冗長化

クラウド黎明期の課題であるジョブ管理マネージャの可用性構成が簡単に実現できます。

運用管理の可用性の課題		共有ディスクの問題	
<p>可用性の要件の違い</p> <p>クラウドミドルはセンシティブな製品です。目的のクラウドに対応しているか確認が必要です。</p> <p>運用管理製品の対応 ○ ?</p> <p>クラスタミドルの対応 ○ or X</p>	<p>目的の障害に対応できるか</p> <p>クラウド上のHAはデータセンタ障害などオンプレミス構成とは違う範囲の障害も検討が必要です。</p> <p>NWセグメント障害(VPC等) ○ ?</p> <p>データセンター障害(AZ等) ○ ?</p>	<p>何を利用するのか</p> <p>クラウドサービスでは目的により選択肢があります。どれを採用するかが製品により変わります。</p> <p>ブロックストレージ(EBS等) ○ ?</p> <p>DBサービス(RDS等) ○ ?</p>	<p>目的の障害に対応できるか</p> <p>クラウド上のHAはデータセンタ障害などオンプレミス構成とは違う範囲の障害も検討が必要です。</p> <p>NWセグメント障害(VPC等) ○ ?</p> <p>データセンター障害(AZ等) ○ ?</p>
FIP(Floating IP)の問題		運用管理製品の問題	
<p>AWSの問題</p> <p>VPC PeerやDirect Connectを跨ぐ環境ではFIPは利用できないため、古典的なHA構成がとれません。</p> <p>VPC Peer間のFIP X</p> <p>Direct Connect間のFIP X</p>	<p>Azureの問題</p> <p>AzureではIP付け替えにAPIコールで何分も必要なので、ロードバランシング方式が推奨されています。(SI対応)</p> <p>IP付け替え処理 時間が掛かる</p> <p>ロードバランサ方式推奨 クラウド依存 SI構築が発生</p>	<p>組み合わせサポート</p> <p>各課題のため未だにクラウド対応が遅れています。単機能製品の場合は、ここでサポート有無を判断する必要があります。</p> <p>運用管理製品</p> <p>クラスタミドル</p> <p>共有ディスク</p> <p>セットで判断 × 製品数</p>	<p>構築・障害時解析の簡易さ</p> <p>一般に製品の範囲だけで解析が困難な複雑な構成になるため、初期構築も障害発生時の解析も困難です。</p> <p>運用管理製品技術者</p> <p>クラスタミドル技術者</p> <p>対象クラウド技術社</p> <p>多くのスキルが必要 × 製品数</p>

# 3

## 運用管理における課題とFAQ ハイブリッドクラウド編

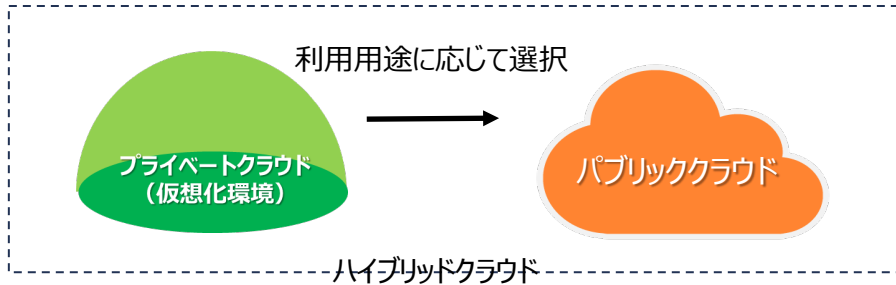


## ハイブリッドクラウドの運用の課題

### ハイブリッドクラウドの流れ

異なるレイヤのクラウドを組み合わせるハイブリッドクラウドですが、プライベートクラウドを基礎基盤として用途に応じてパブリッククラウドと連携する組み合わせが多いです。

- ・クラウドベンダロックイン排除
- ・重要データの持ち出しの困難性
- ・適材適所でのクラウド利用
- ・スモールスタート



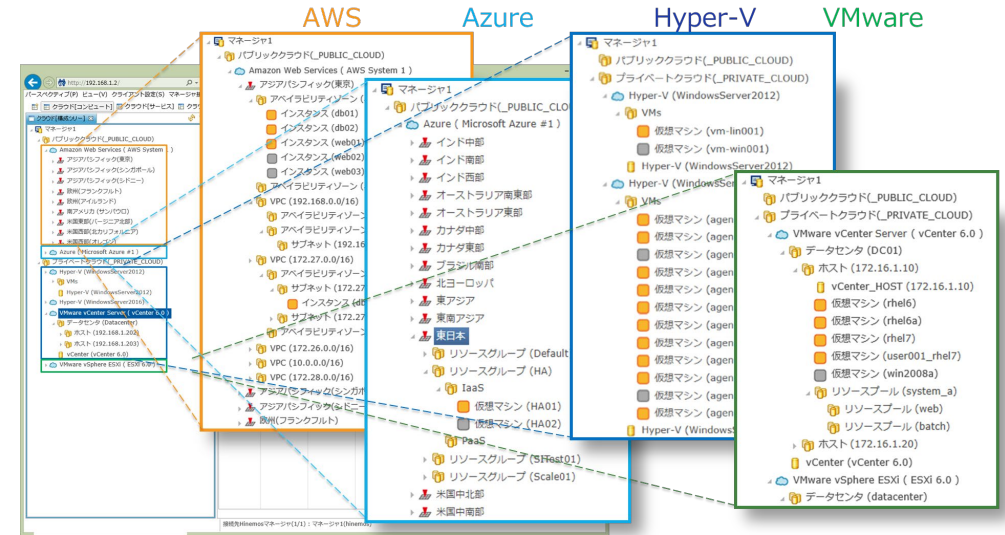
### ハイブリッドクラウドの運用要件

ハイブリッドクラウドの運用を行う際に、次の2つが重要な運用要件になります。当初、プライベートクラウドのみを運用を行い、順次パブリッククラウドを活用していくようなケースにおいても、本要件を満たす製品を選定しておくことが重要です。

- ・ **俯瞰的な管理が出来ること**  
各クラウド個別の製品・サービスを導入しての運用管理は、エンジニアの習得や日々のオペレーションなど様々なカットで運用コストが増大します
- ・ **マルチインフラでの動作サポート**  
対象環境で運用製品のマネージャやエージェントが動作することは最低限の要件です。商用製品の場合は特に、可用性やライセンス費用も含めた確認が必要です。

## ハイブリッドクラウド統合管理

Hinemosは単一画面でオンプレミス、仮想化、クラウドなどを組み合わせたヘテロな環境を俯瞰的に管理することができます。クラウドについては、動的に変更するインフラ運用を吸収し、オンプレミス環境のときと同一のイメージで運用を行えます。



Hinemosは多種多様なクラウドサービス上で動作します。運用管理を行うために、その環境にHinemosマネージャを構築するという制約はありません。VMware環境上のHinemosからAWSの運用管理も可能です。

- ・ **動作確認済みクラウドサービス**  
AWS, Microsoft Azure, IIS, GIO Managed Cloud Platform, Enterprise Cloud, ニフクラ, Google Cloud Platform, IBM Cloud
- ・ **動作確認済みハイパーバイザ**  
VMware, Hyper-V, Xen, KVM

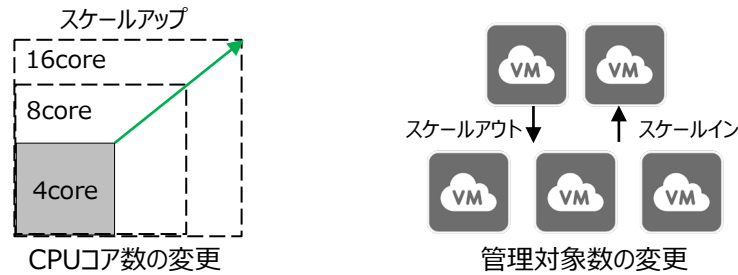


# 費用体系

## CPU数・管理対象数変更によるライセンス変動

### リソース拡張・追加に影響

一般的な運用管理製品はCPUコア数、もしくは管理対象台数に依存して費用が変動します。そのため、クラウドのメリットである柔軟なリソースのスケールアップ・ダウン、スケールイン・アウトを実現しようとすると運用管理製品のライセンス費用という价格的な課題が生じます。



運用管理製品の費用に変動が生じてしまう！

### マシンスペック・配置設計に影響

プライベートクラウドにコアライセンスの運用管理製品を導入する場合、エージェントを導入する仮想マシンのCPU数はもちろんのこと、仮想マシンが動作するサーバのCPU数にも、ライセンスコストが影響するケースもあります。

#### ・ハイスペックマシンの利用に影響

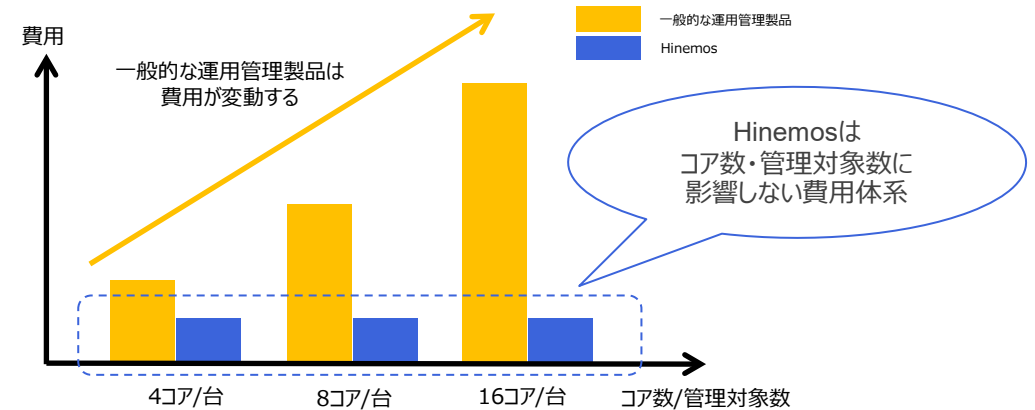
コアライセンス製品では管理対象数が少なくてもCPUコア数の多いハイスペックマシンでは費用が高額になり、簡易に処理能力の高いマシンを利用できるというクラウドのメリットを享受できない。

#### ・仮想マシンの配置場所の設計に影響

あるホストサーバのリソースが厳しくなった場合に、その上で動作する仮想マシンを別のホストサーバに移動しようとした際に、ライセンスコストの変動がないかを確認するといった、配置設計を意識する必要がでてくる。

## リソース・管理対象数に依存しない費用体系

Hinemosはコア数や管理対象数に依存する費用体系ではありません。そのため、マネージャやエージェントが動作するサーバのスペックや管理対象数に依存せず利用することが可能です。



そのため、一旦導入した環境において、仮想CPU数の変更や動作するホストサーバの変更、必要に応じた仮想マシンの追加（スケールイン・アウト）について、費用について意識することなく実施することができます。これにより、柔軟にリソースを変更できるクラウドのメリットを享受することができます。



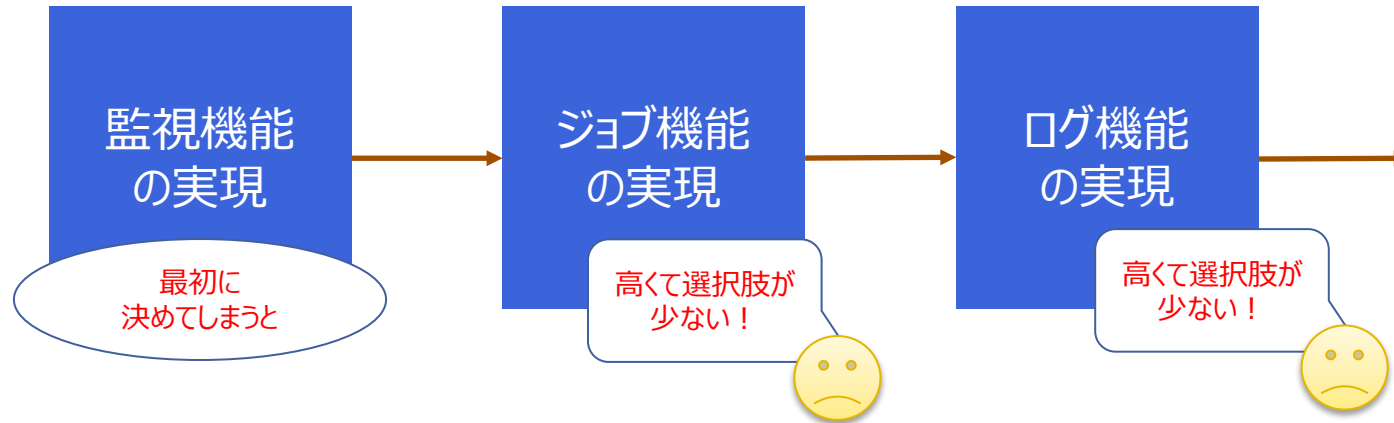
## クラウド・VMware等の仮想環境

構成変更に対する運用製品コストの変動を意識する必要がない

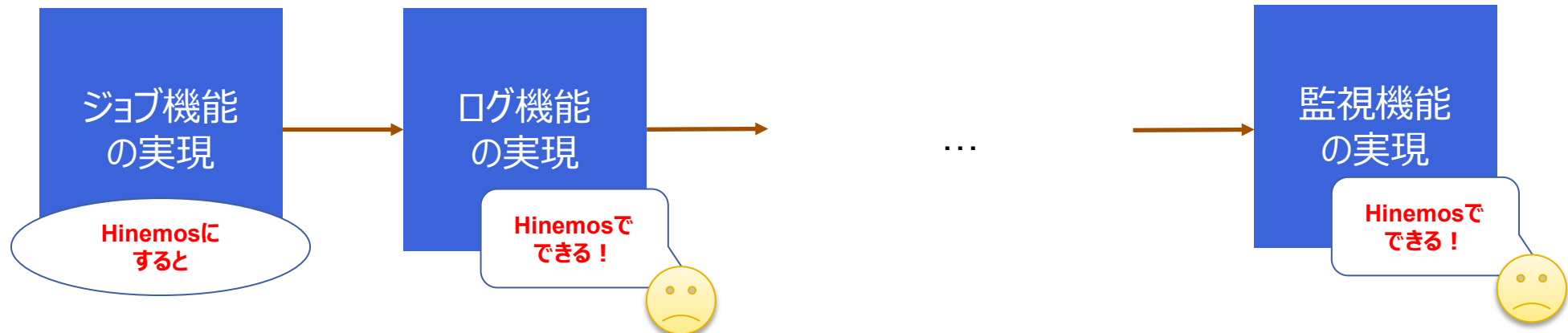
## FAQ クラウドにおける運用管理基盤の検討順は？

監視はコモディティ化され、一番最後に選択したとしても選択肢は残ります。

そのため、監視は必須だといっても、「監視製品から」選定を、では全体最適となる構成は組めません。



高価、クラウド対応が進んでいない製品（例：ジョブ管理製品）の検討・製品選定を、まず初めに行うことをお勧めします。Hinemosを選択すると、その時点で全体最適（費用・構成）な運用管理基盤ができます。



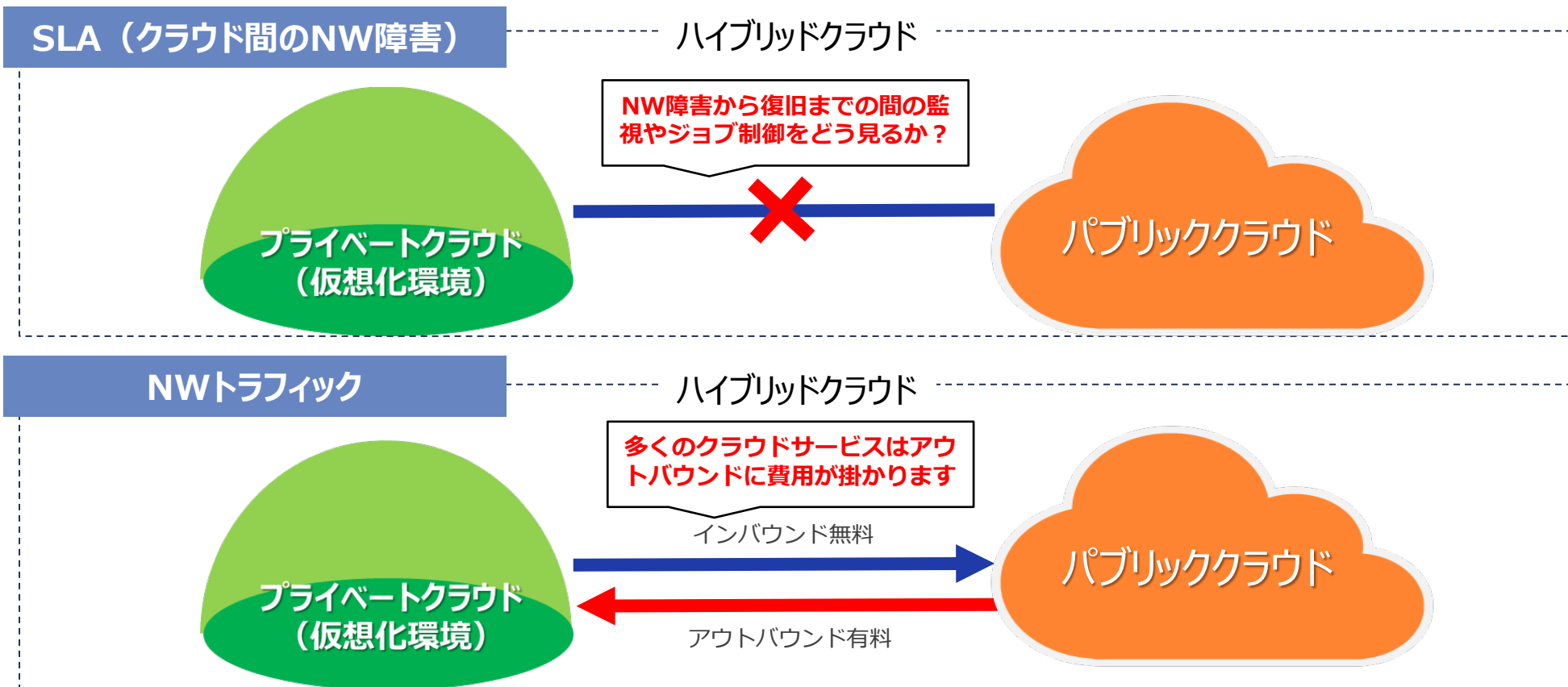
# FAQ 監視におけるHinemos採用メリットは？

基盤専用の監視ツール/サービスと比べてもHinemos採用のメリットは多々あります。

観点	Hinemos	基盤専用ツール/サービス
リソース情報の保存期間	ユーザ指定 (ディスク容量次第)	保存期限ありのケースが多い 古いデータは間引きされるケースが多い
カレンダー連携	高度なカレンダー定義可能	詳細が設定できないケースが多い
外部通知	高度な通知設定可能	外部サービスとの連携、作り込が発生し、流量制限ありのケースがある
ジョブ連携	可能	外部サービスとの連携、作り込みが必要 高度なジョブ定義は不可のケースがある
イベント集約	基盤・システム、様々なカットでイベント集約	ジョブ管理や業務管理を別の仕組みを用意すると、それらの集約が必要になる
マルチクラウド ハイブリッドクラウド	基本操作変更なし	個別に仕組みを検討、再度作り込が必要の可能性はある

## FAQ Hinemosマネージャはどこに配置すればよい？

ハイブリッドクラウド運用の場合にHinemosマネージャをどこに配置すべきかは、SLA（クラウド間のNW障害）、NWトラフィックの観点で決定します。



上記理由より、次のような配置設計を採用されるケースが多いです。

- ・プライベートクラウド側に親（集約）Hinemosマネージャを配置
- ・拠点毎に子（テナント別）Hinemosマネージャを配置

# FAQ マルチテナント運用におけるHinemosマネージャの配置は？

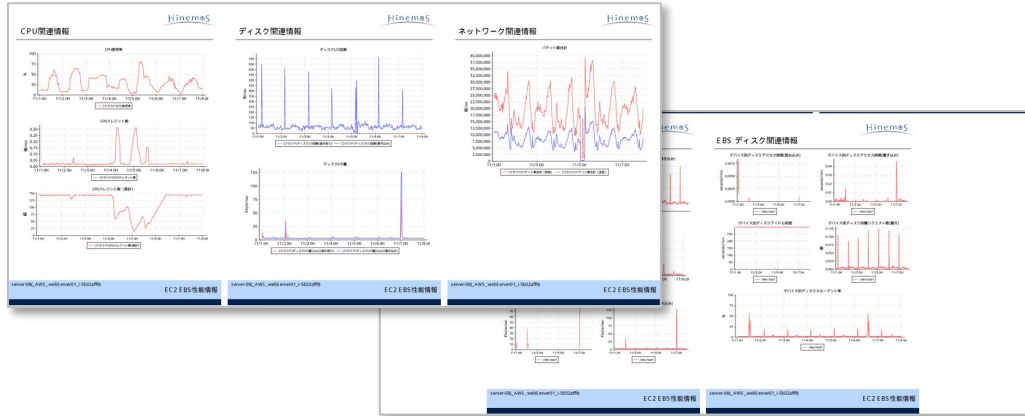
クラウド環境を活用して、小規模から大規模の様々なテナントに対して、Hinemosマネージャをどのように配置すべきかは、テナントの規模・SLAの観点でカテゴライズしたルールを決定することが多いです。（以下、カテゴライズ例）

梅コース	竹コース	松コース
 <p>Hinemosマネージャ シングル構成</p>	 <p>Hinemosマネージャ シングル構成</p>	 <p>Hinemosマネージャ HA構成・その他</p>
<p>【梅コース】</p> <ul style="list-style-type: none"><li>・Hinemosシングル構成</li><li>・1マネージャで複数テナント共用</li></ul>	<p>【竹コース】</p> <ul style="list-style-type: none"><li>・Hinemosシングル構成</li><li>・1マネージャ1テナント占有</li></ul>	<p>【松コース】</p> <ul style="list-style-type: none"><li>・Hinemos HA構成</li><li>・1マネージャ1テナント占有</li><li>・規模により複数マネージャ占有</li></ul>
<p>【用途】</p> <ul style="list-style-type: none"><li>・小規模向け（台数）</li><li>・低SLA</li></ul>	<p>【用途】</p> <ul style="list-style-type: none"><li>・中規模向け（台数）</li><li>・NWセキュリティ要件（NW独立）</li><li>・中SLA</li></ul>	<p>【用途】</p> <ul style="list-style-type: none"><li>・大規模向け（台数）</li><li>・NWセキュリティ要件（NW独立）</li><li>・ジョブ/監視の高可用性要件</li></ul>

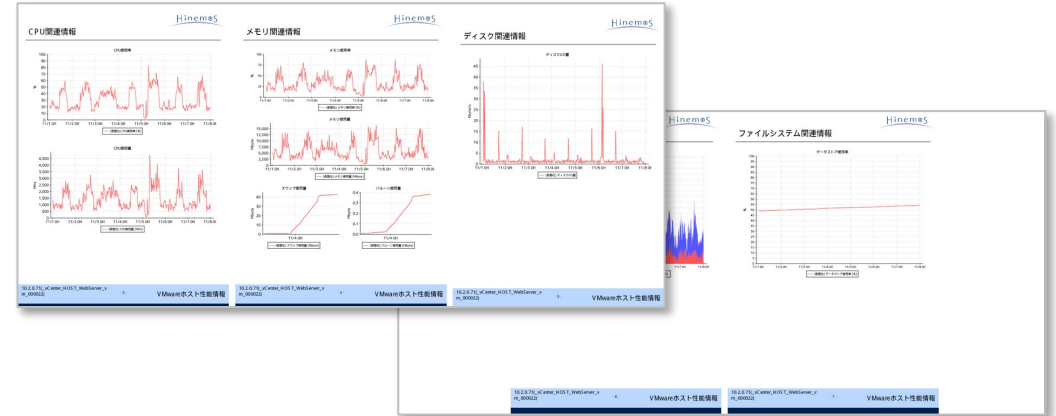
# FAQ VMwareやAWS環境向けの運用報告書・レポートテンプレートは？

Hinemosのレポート機能で対応しています。インストール後、直ぐに利用可能です。

## AWS EC2,EBS性能情報



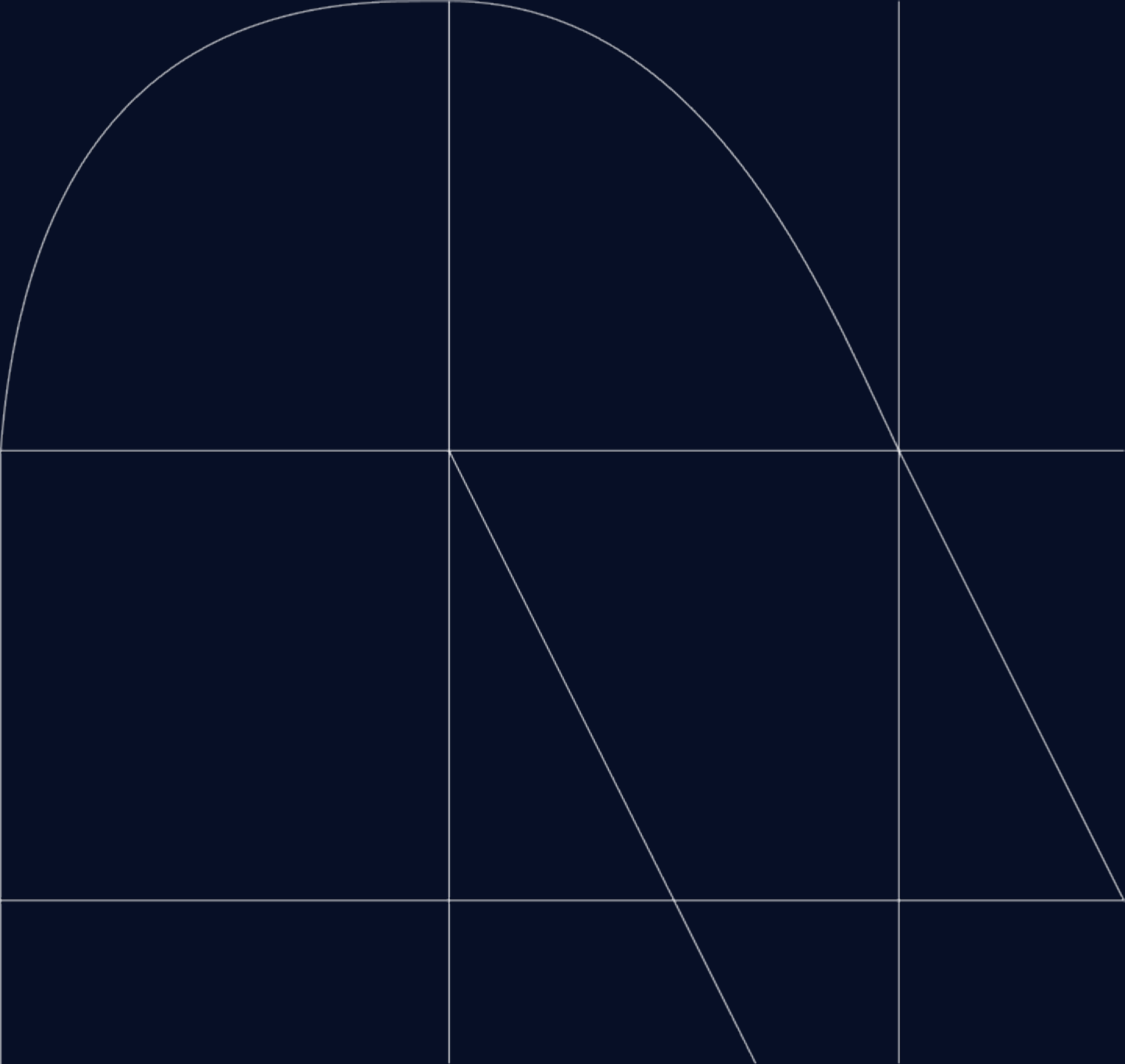
## VMwareホスト性能情報



環境	レポート	説明
AWS	AWS クラウド課金情報 AWS EC2,EBS性能情報 AWS ELB, Auto Scaling性能情報 AWS RDS性能情報	クラウド環境(AWS)専用の課金情報と 性能情報をグラフ形式で出力します。
VMware	VMwareホスト性能情報 VMware仮想マシン性能情報	仮想化環境(VMware)専用の性能情報を グラフ形式で出力します。

# 4

## appendix



Hinemosではじめる

## 実践ジョブ管理・自動化入門 技術評論社

2023.3.9 発売開始

最新バージョン ver.7.0に対応

Amazonから購入可能





# お問い合わせはこちら

## Hinemosに関するお問合せ

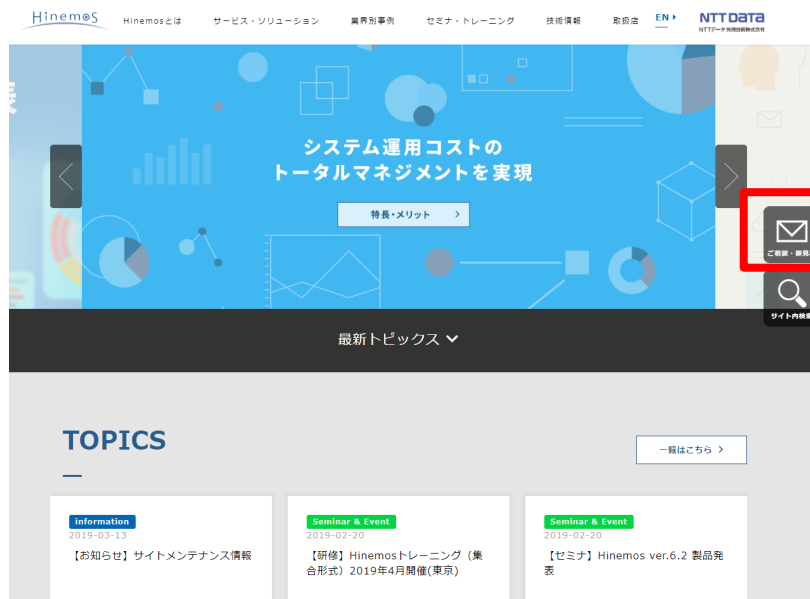
お気軽にお問合せください。

Hinemosポータルサイト

URL : <https://www.hinemos.info/contact>

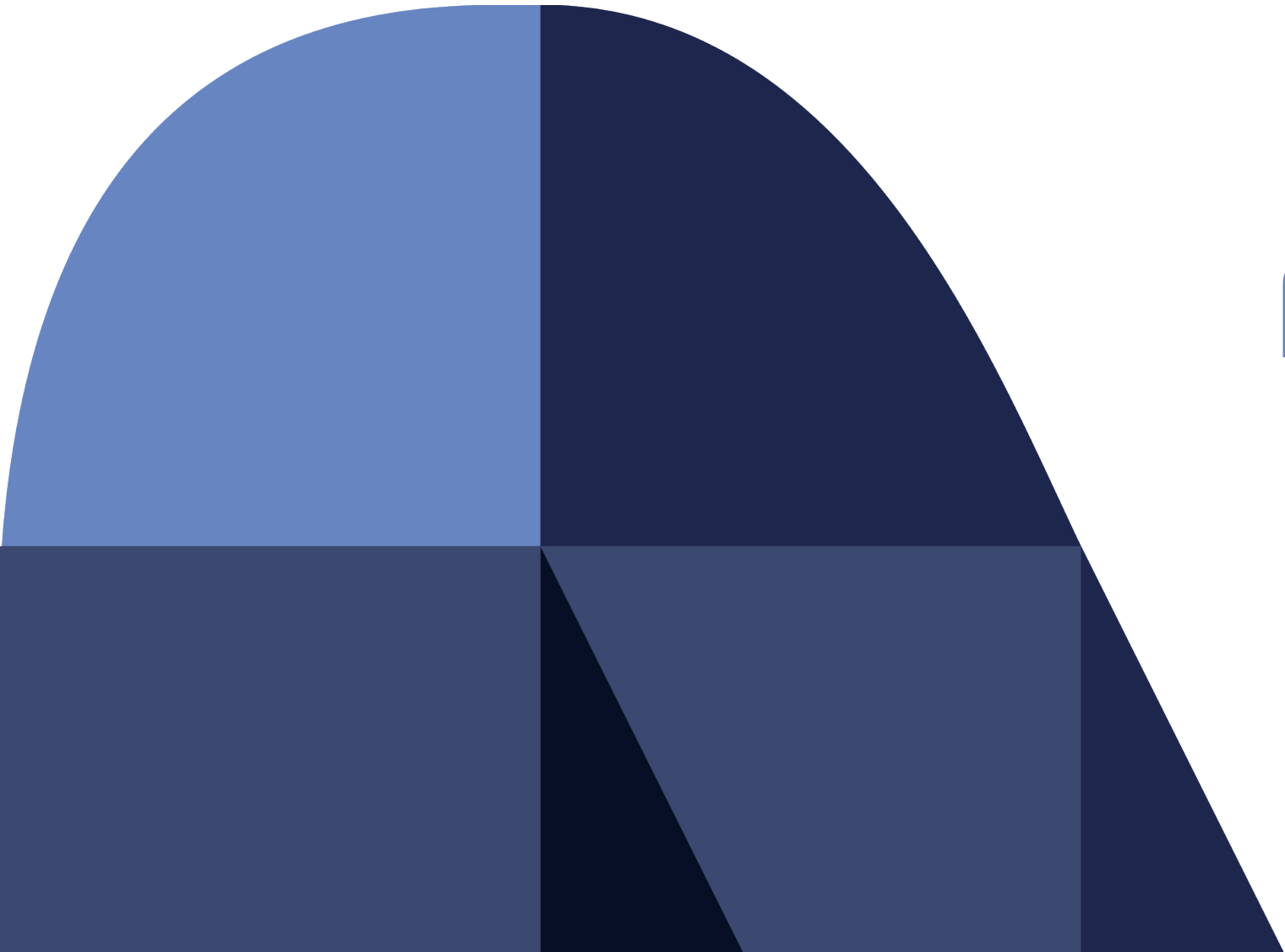
 

お待ちしているもに！



ご相談フォーム





**NTT DATA**  
Trusted Global Innovator