

PostgreSQL Monitoring

release date. 2026. 09. 11.

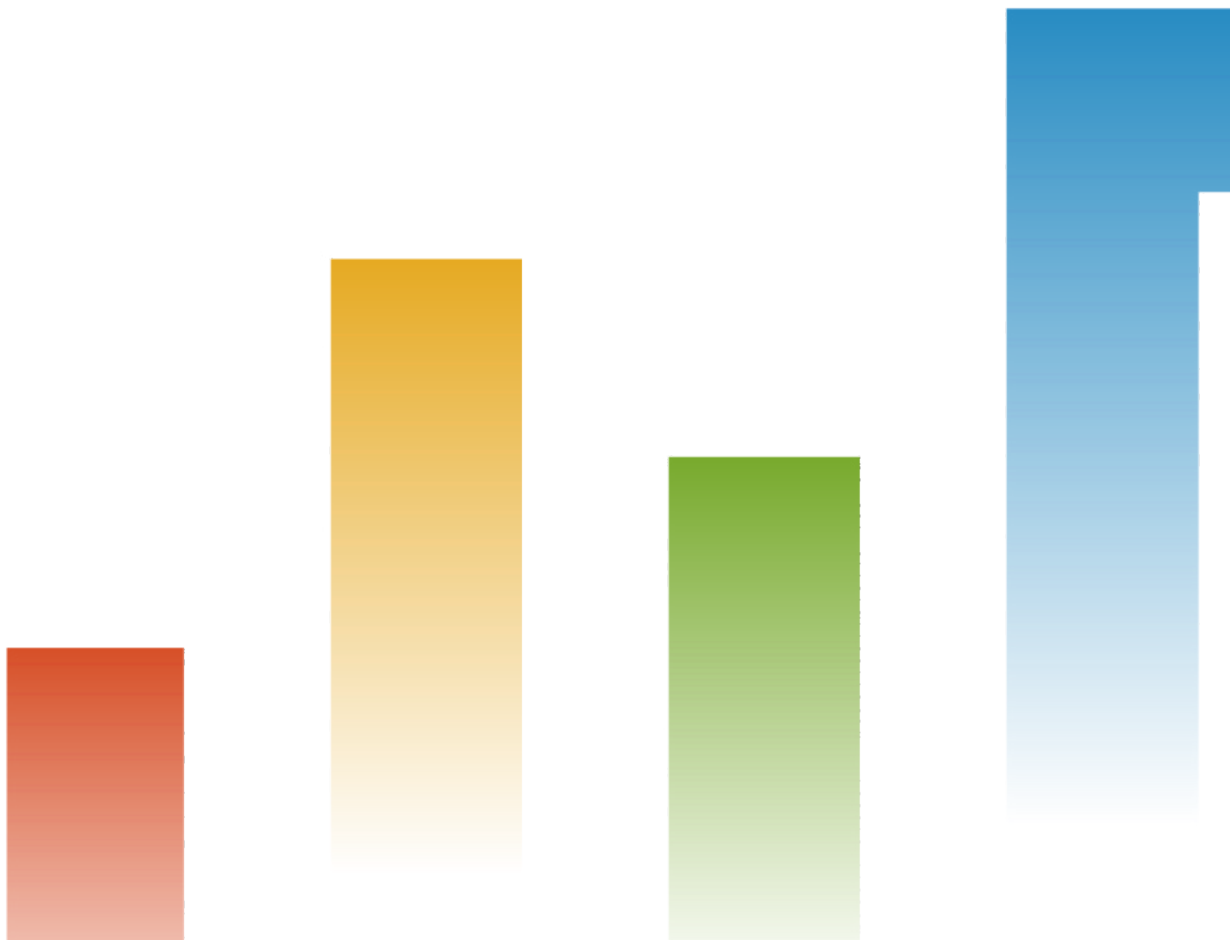


Table of Contents

• PostgreSQLモニタリング	5
◦ はじめに	10
◦ サポート環境	12
◦ インストール	
▪ エージェントのインストール	
▪ 手動インストール	19
▪ パッケージマネージャーインストール	32
▪ Dockerインストール	39
▪ インストールチェックリスト	46
▪ インストール問題の解決	48
◦ 設定	51
▪ 迅速なエージェント設定	54
▪ DBXエージェントの設定	58
▪ XOSエージェント設定	68
▪ クラウドエージェント設定	74
▪ インストールと使用	87
▪ ログ収集設定	93
▪ サーバーレスエージェント設定	96
▪ サーバーレスログ収集設定	115
◦ 管理	122
◦ ダッシュボード	126
▪ インスタンスリスト	128
▪ インスタンスマップ	131
▪ トポロジ	140

▪ インスタンスモニタリング	143
▪ マルチインスタンス	171
▪ スロークエリ	196
▪ Flexボード	201
▪ Flexボードの使用方法	204
▪ ウィジェットの管理	215
◦ 分析	
▪ カウント推移の比較	220
▪ ロックツリー	228
▪ セッション履歴	236
▪ Wait分析	243
▪ Vacuum分析	246
▪ Topオブジェクト	256
▪ Top SQL比較	266
▪ データベースパラメータ	274
◦ メトリクス	276
▪ PostgreSQLメトリクス	278
▪ メトリクスチャート	286
▪ メトリクスチャート	309
▪ メトリクス照会	312
▪ メトリクス異常検出	314
▪ メトリクストレンド分析	317
◦ 統計/レポート	
▪ Top SQL	320
▪ PG SQL統計	328
▪ データベースのサイズ	337

▪ レポート	343
◦ ログ	346
▪ 適用方法	348
▪ ライブテール	350
▪ ログエクスプローラー	362
▪ ログトレンド	369
▪ ログの検索	377
▪ ログの設定	391
▪ ログパーシング	427
◦ アラート通知	442
▪ イベント設定	0
▪ メトリクス	449
▪ 複合メトリクス	470
▪ 異常値	477
▪ リアルタイムログ	481
▪ 複合ログ	493
▪ イベント設定	507
▪ 通知設定	511
▪ イベント通知フォーマット	533
▪ イベント履歴	541
▪ リアルタイムアラート	549
▪ システムイベント	552
◦ 管理	
▪ 連携プロジェクト管理	554
◦ 実験室	562

PostgreSQLモニタリング

データベースのパフォーマンス管理は、データベースのパフォーマンスをリアルタイムでモニタリングし、問題点を診断、分析します。問題のあるSQLをチューニングして適用し、問題が解決されたかをリアルタイムでモニタリングします。

WhaTapデータベースモニタリングサービスは、これら一連のタスクを循環管理できるように設計されています。データベースの動作をモニタリング、診断、分析し、可用性とパフォーマンスを一定のレベルに維持できるように構成されています。データベースに問題が発生した場合、担当者にアラームを送信して状況を認識し、収集されたデータからデータベースの状況をよりに分析することで、Root Causeを明確にする支援サービスです。

❗ データベースプラットフォームごとの対応機能の詳細については、[次の文書](#)を参照してください。

PostgreSQLモニタリングの特長

- リアルタイムDBモニタリング

リアルタイムのDBパフォーマンス指標と実行中のclient(アクティブセッション)の情報を収集します。稼働状況を一目で確認しながら負荷発生状況をすぐ検知できるし、通知設定で問題発生をすぐに認識できます。

- 特定時点の分析

指標の短期推移、長期推移、指標比較などの機能を提供し、過去の状況を分析できます。データベースのパラメータ変更履歴を追跡します。

- 大容量規模でも効率的な構造

複数のDBサーバーを管理する場合でも、一元化された性能データ収集および効率的なデータストレージ構造により、最適な分析環境を提供します。

- クラウドDBモニタリング可能

クエリベースのモニタリングデータ収集方式で、DBサーバーの種類に制限があり暗線。AWS、GCP、Azureおよび、OCI環境のDBもモニタリングができ、クラウドのリソース指標との統合モニタリングにも対応しています。

- 統合モニタリング

Application、Server、Databaseモニタリング製品をすべてインストールすると、連携されたビューを使用して障

害の原因を素早く分析し特定できます。

- WhaTap SaaS

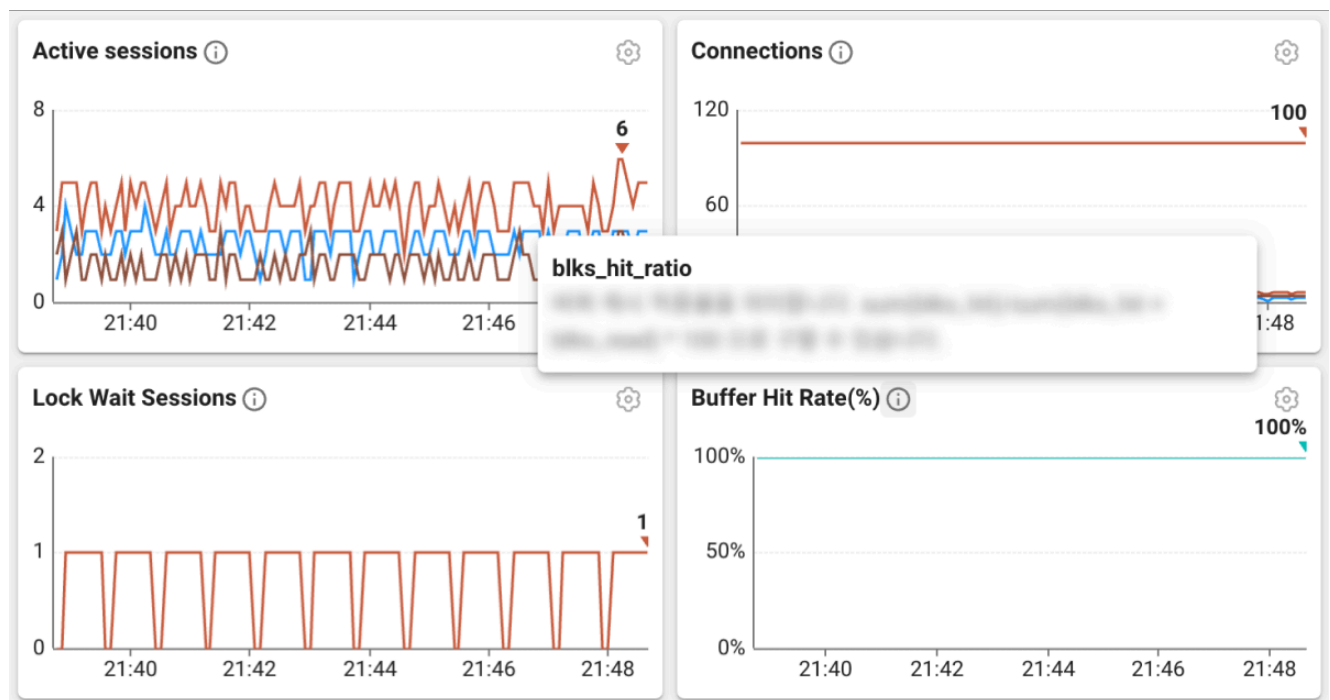
多様なDatabase製品を選択でき、サブスクリプションで利用期間を自由に設定できます。(オンプレミスタイプも提供可能)

データベースモニタリングの新機能

チャートおよび収集機能を強化したPostgreSQLモニタリングをアップデートしました。最新の機能を利用するには、エージェントをアップデートしてください。

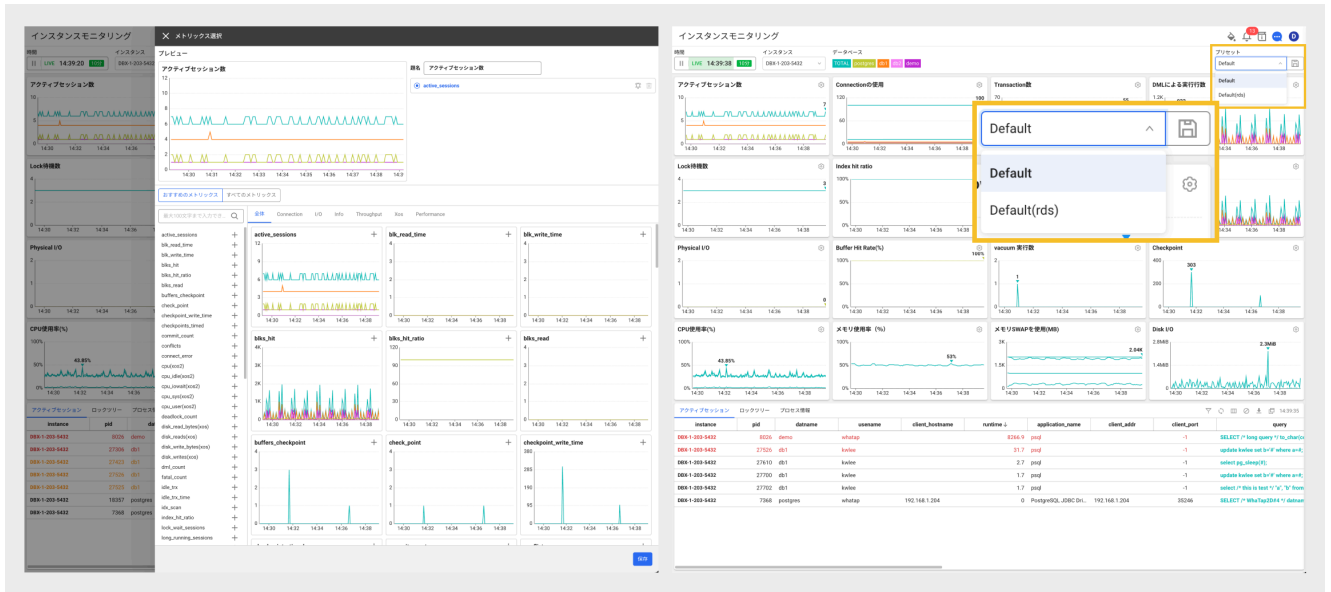
❗ PostgreSQLモニタリングを利用するエージェントの最小バージョンは1.6.7です。 エージェント更新の詳細については、[次の文書](#)を参照してください。

- DB性能指標の情報提供



指標のヘルプとチューニングに関するヒントを追加し、ユーザーが数多くの指標を簡単に理解できるようにカテゴリ化して利便性を提供します。(多言語に対応予定)

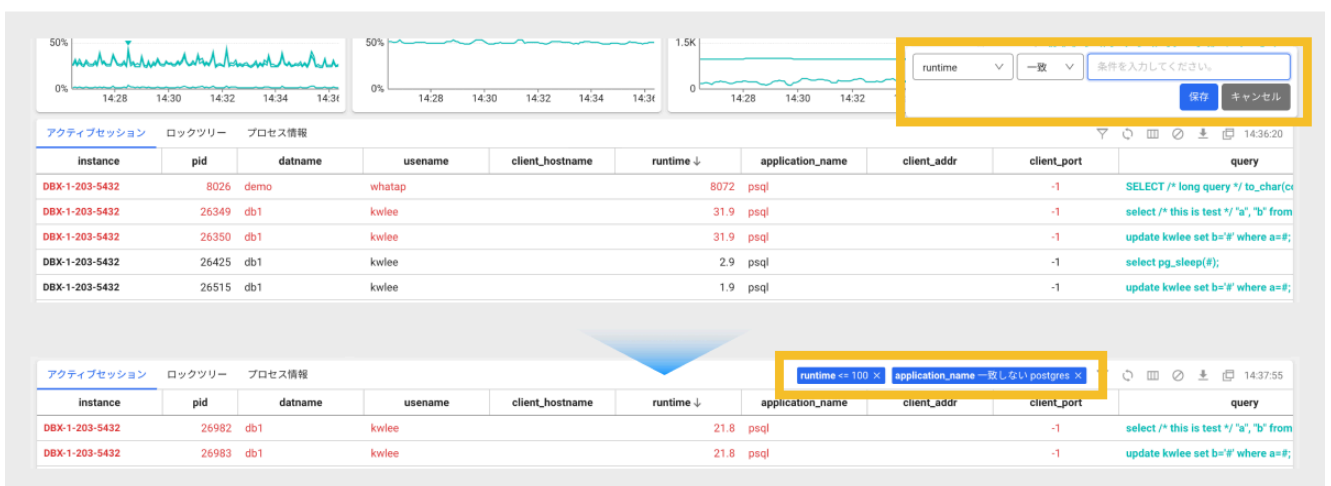
ダッシュボード機能の改善



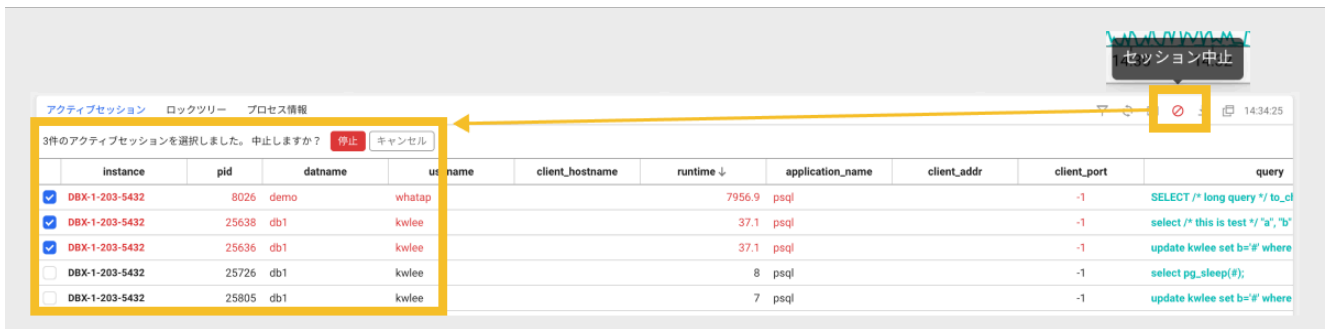
ダッシュボードでは、1つのチャートに複数の指標を追加する機能が追加されました。また、プリセット機能を追加してユーザーが必要な指標を素早く確認できます。

UI改善

インスタンスモニタリングフィルター機能の改善

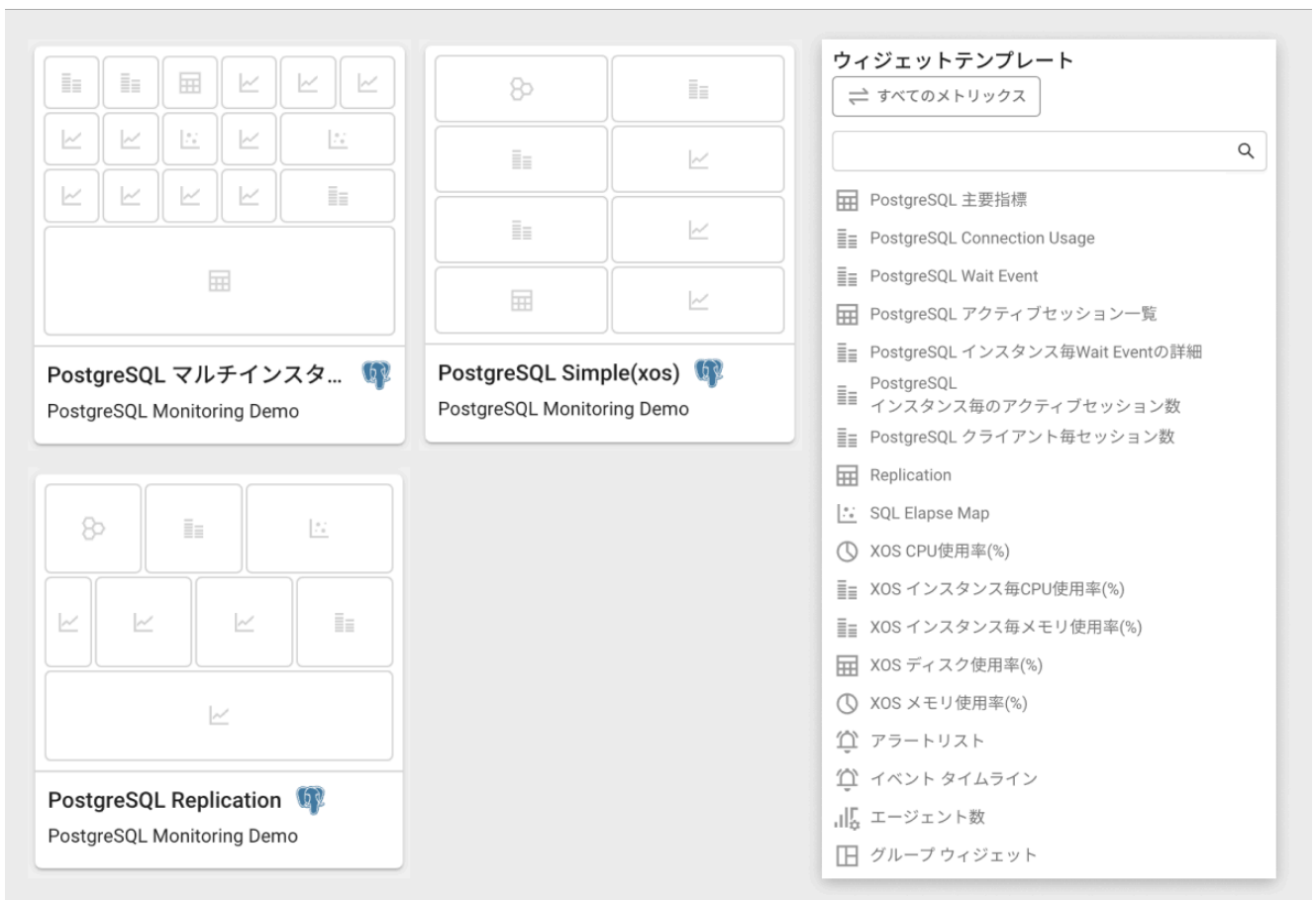


セッションキル機能UIの改善



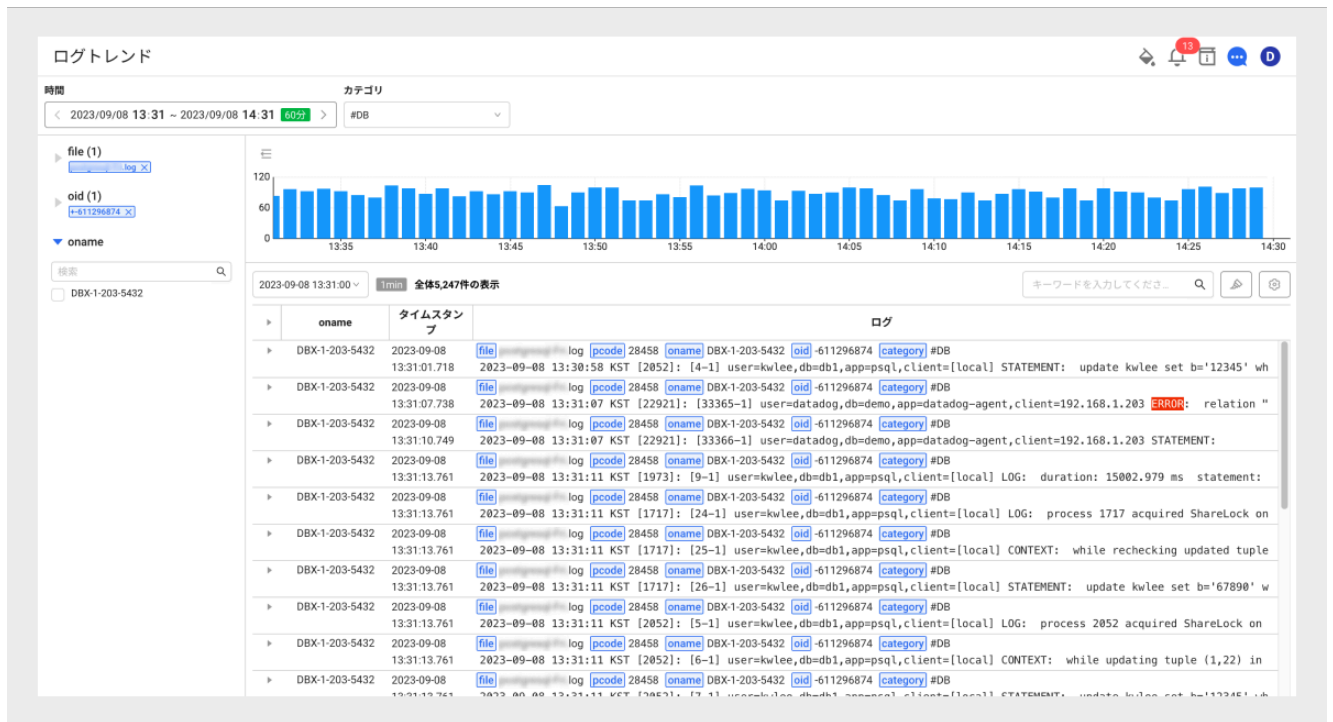
インスタンスモニタリングのフィルター機能を改善し、セッション強制終了機能と関連するユーザーインターフェース(User Interface)を改善しました。

- Flexボードテンプレートおよびウィジェットの追加提供



DB指標の多くをメトリックスに変換し、追加のテンプレートとウィジェットを提供します。

- ログ



ログTailとログトレンド、ログ検索、ログ通知機能で、DBログを検索して推移を確認できるし、ログ関連の通知を受信できます。

はじめに

WhaTapモニタリングは次の手順で開始します。

[ステップ1. 会員登録](#)

[ステップ2. 製品選択](#)

[ステップ3. プロジェクト作成](#)

[ステップ4. アクセスキー確認](#)

その後、サポート環境を確認してからエージェントのインストール手順に進みます。

会員登録

WhaTapモニタリングサービスを利用するには、まず[会員登録](#)が必要です。会員登録に関する詳細は、[アカウント管理 > 会員登録](#)ドキュメントを参照してください。

製品選択

1. [WhaTapモニタリングサービス](#)にログインします。
2. ≡ [プロジェクト全体](#) > [+ Project](#)ボタンをクリックします。
3. [商品選択](#)画面でモニタリングするGo製品を選択してから作成ボタンをクリックしてください。

プロジェクト作成

モニタリングする製品を選択したら、エージェントをインストールする前にまずプロジェクトを作成します。

1. [プロジェクト作成](#)画面でプロジェクトを作成するには、次の項目を入力または選択してください。
 - [プロジェクト名](#)(必須): プロジェクト名を入力します。
 - [データサーバーリージョン](#)(必須): データサーバーが位置するリージョン(地域)を選択します。リージョンはクラウドサービスを提供するデータセンターのグループです。選択したリージョンにユーザーのデータが保存されます。

- **プロジェクトタイムゾーン(必須):** アラートおよびレポート生成時の基準となる時間を設定します。
- **通知言語の設定(必須):** プロジェクトで発生する警告通知メッセージの言語を設定します(韓国語、英語対応)。
- **プロジェクトグループ:** 複数のプロジェクトをグループにまとめて管理できます。所属するグループがあれば選択してください。
- **プロジェクト説明:** プロジェクトに関する追加説明を入力します。

2. すべての設定を完了したら**プロジェクトを作成する**ボタンをクリックしてください。

❗ 組織を選択した状態でプロジェクトを追加する場合、**組織下位のグループ**を必須で設定する必要があります。グループに関する詳細は、[次のドキュメント](#)を参照してください。

アクセスキー確認

アクセスキーは、WhaTapサービスを有効化するための固有IDです。

インストールガイドセクションで**プロジェクトアクセスキーの発行ボタン**を選択してください。アクセスキーを自動的に発行した後、次のステップに進みます。

❗ プロジェクトを作成すると、自動的に**エージェントインストールページ**に移動します。エージェントインストールページに移動しない場合は、画面左側のメニューから**管理 > エージェントインストール**を選択してください。

サポート環境

PostgreSQLモニタリングを開始する前に、次のサポート環境を確認してください。

データベース対応バージョン

PostgreSQLベース(EDB、Tmax OpenSQLなど) 9.2バージョン以降に対応します。

- ❗ エージェントをインストールするサーバーには、Java 8以降のインストールが必要です。
- データベースサーバーは、DBXエージェントがDBポートにアクセスできるように、Inbound接続を許可する必要があります。
- DBXエージェントがWhaTapサーバーにデータを転送できるように、6600ポートのOutboundを開く必要があります。

エージェントの最小インストール仕様

- CPU:** 1コアあたり約20個のエージェントのインストールが可能(DBサーバーの負荷によって変動可能)
- メモリー:** 200MB
- ディスク:** 100MB(エージェントスペース50MB + ログファイル作成スペース50MB)

❗ エージェント数による最小インストール仕様

エージェントインストール数	CPU	メモリ	ディスク
10個	1コア	2GB	1GB
20個	1コア	4GB	2GB
40個	2コア	8GB	4GB

エージェント方式

DBエージェントは、次のような方式で基本動作します。

- 遂行周期: 5秒(既定値)
- 実行クエリ数: 5~6個
- 速度: サーバーの負荷が正常な場合、1秒以内に終了
- 最大2つのセッション(5秒収集セッション、その他の情報収集セッション)

収集クエリのタイムアウトが発生すると、DBエージェントは負荷を最小化するために次の方法で動作します。

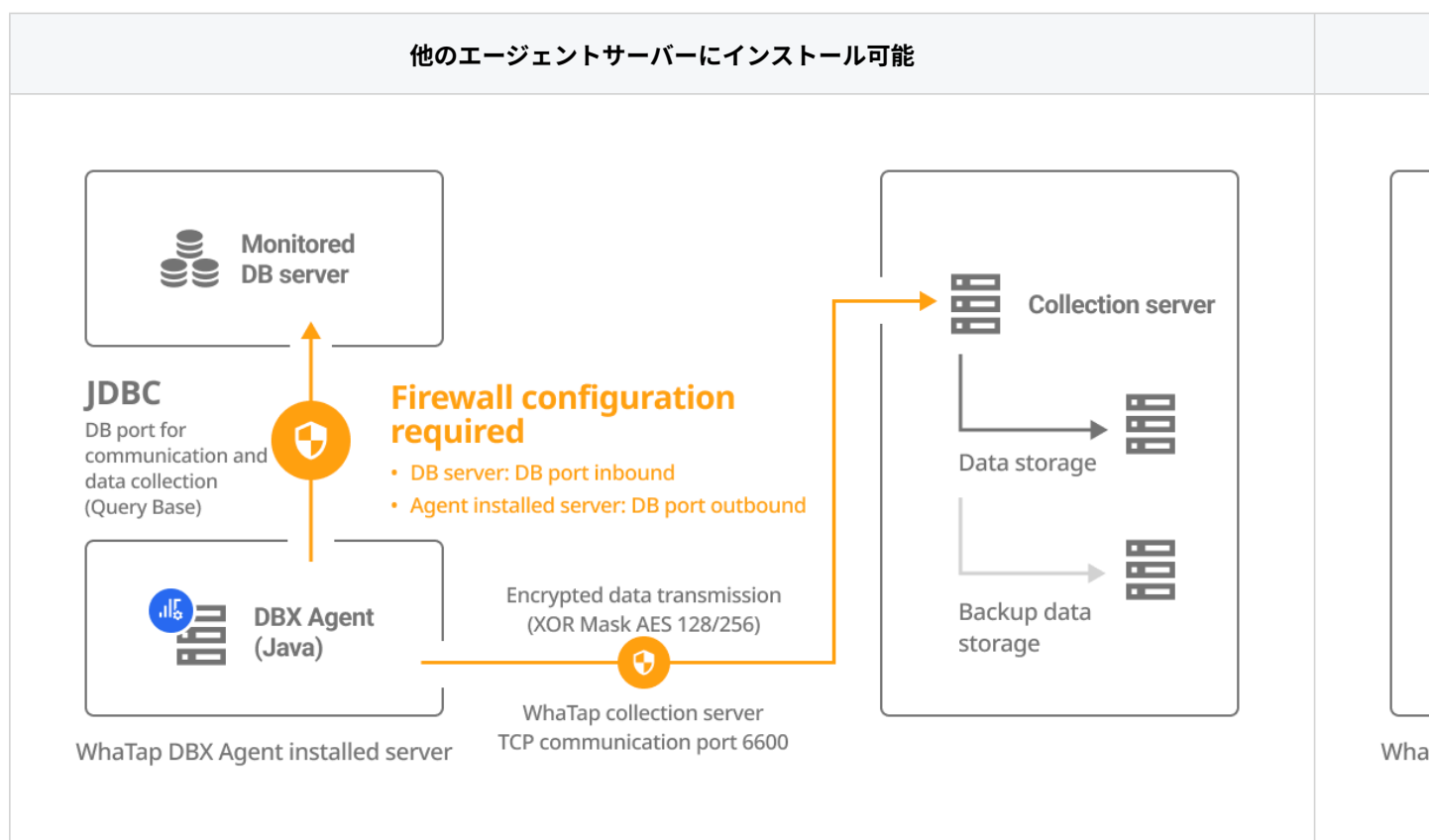
- 以前の収集クエリが終了するまで、新しい収集クエリを実行しません。
- タイムアウトが12秒を超えた場合(リフレッシュ周期が2回を超えた場合)は、セッション終了後に再接続します。
- 3回以上タイムアウトが発生した場合は、10秒待機してから再試行します。

❗ DBサーバーの負荷状態によって収集クエリの結果がWhaTapモニタリングサービス画面に出力されるまでに長い時間がかかったり、タイムアウトが発生することがあります。

エージェント構成

WhaTapのデータベースモニタリングサービスは、エージェントと収集サーバーで構成され、エージェントと収集サーバーの間でセキュリティのためにデータを暗号化します。外部ライブラリを使用していないため、インストールやアップデートのためにシステムを再起動する必要はありません。

- WhaTapデータベースモニタリングサービスは、データベースサーバーにエージェントをインストールしなくても、別のエージェントサーバーでモニタリングすることができます。ユーザーの環境に柔軟に対応できます。



- XOSエージェントは、データベースサーバーのプロセス使用状況をモニタリングできる付加オプションエージェントです。データベースサーバーでプロセス使用状況をモニタリングする場合は、データベースサーバーに別のエージェントを起動してデータを収集できます。
- エージェントとサーバーは、バイナリベースの通信プロトコルを使用します。また、データを選択的に暗号化することで、データのセキュリティを確保しながら、暗号化の負荷を低減します。

- ⓘ • モニタリング対象のデータベースサーバーとネットワーク通信が可能なサーバーにWhaTapエージェントをインストールする必要があります。
- WhaTap DBXエージェントサーバには、**Java 8バージョン以降のJDK**をインストールする必要があります。

サービス提供形態

WhaTapモニタリングは、SaaSまたはオンプレミス（インストール型）方式でご利用いただけます。どちらを選択しても、同じエージェントで同じモニタリング機能をご利用いただけます。

Deployment	Support	Description
SaaS		WhaTapが運用する収集サーバーに接続。リージョン別アドレスはファイアウォール項目を参照
オンプレミス（インストール型）		収集サーバーを自社で構築して運用（閉域網環境に対応）。導入範囲は営業担当者と協議

❗ インターネットに接続しない閉域網環境でも、オンプレミスで構築できます。

提供形態による機能の違い

どの形態を選択しても、モニタリング機能は同じです。モバイルアプリのみが例外で、service.whatap.io SaaSでのみご利用いただけます。

❗ オンプレミス環境で一部の機能をご利用いただくには、別途契約が必要です。

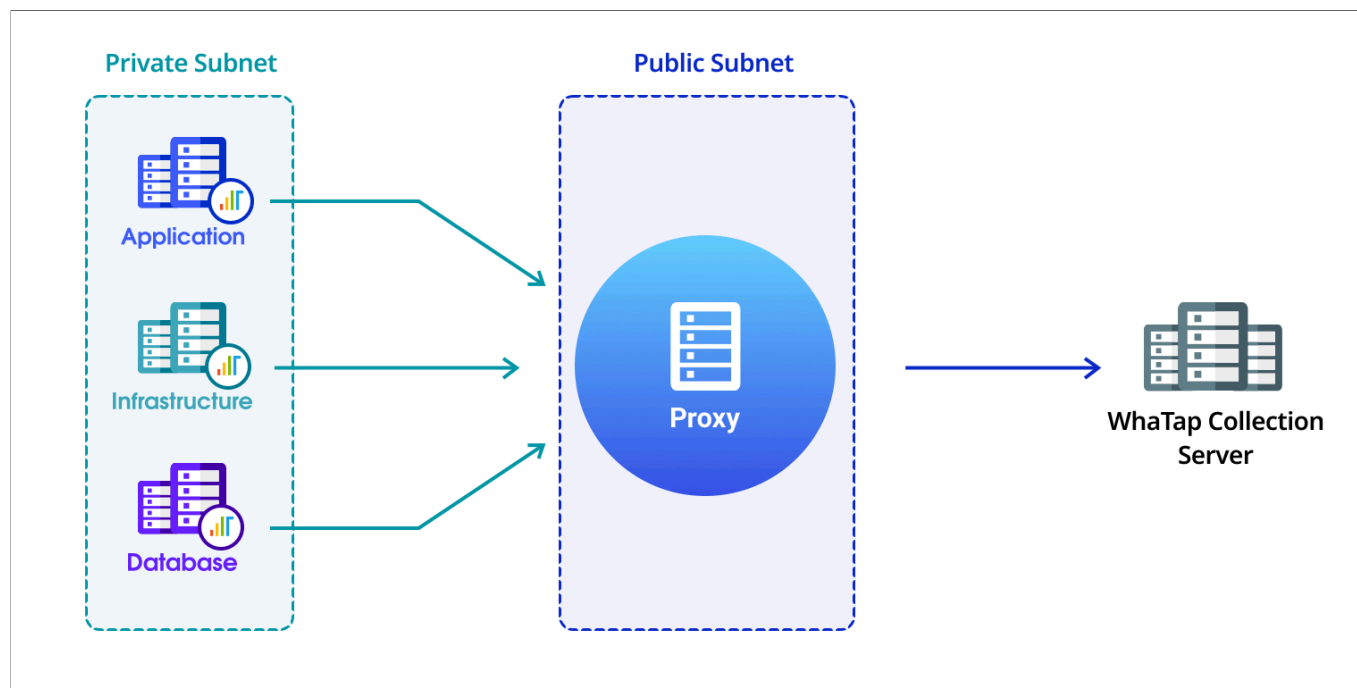
ファイアウォール

WhaTap エージェントは、収集サーバーの **TCP 6600** ポートにアクセスする必要があります。監視対象に最も近い収集サーバーのアドレスを許可してください。

出発地：WhaTapエージェント

クラウド	目的地	目的地IP	ポート番号
AWS	WhaTapソウル収集サーバー	13.124.11.223 / 13.209.172.35	TCP 6600
	WhaTap東京収集サーバー	52.68.36.166 / 52.193.60.176	TCP 6600
	WhaTapシンガポール収集サーバー	18.138.0.93 / 18.139.67.236	TCP 6600
	WhaTapジャカルタ収集サーバー	108.136.91.69 / 108.137.158.44	TCP 6600
	WhaTapタイ収集サーバー	43.209.109.219 / 43.209.48.86	TCP 6600
	WhaTapムンバイ収集サーバー	13.127.125.69 / 13.235.15.118	TCP 6600
	WhaTapカリフォルニア収集サーバー	52.8.223.130 / 52.8.239.99	TCP 6600
	WhaTapバージニア収集サーバー	107.23.220.101 / 54.236.221.105	TCP 6600
	WhaTapフランクフルト収集サーバー	3.125.142.162 / 3.127.76.140	TCP 6600
Azure	WhaTapソウル収集サーバー	52.231.66.38 / 20.194.5.115	TCP 6600
	WhaTap東京収集サーバー	52.246.169.54 / 20.210.27.232	TCP 6600
Kakao	WhaTapソウル収集サーバー	61.109.237.237 / 61.109.238.166	TCP 6600

エージェントから収集サーバーに直接接続できない場合は、提供するProxyモジュールを使用して経由します。



対応ブラウザ

WhaTapモニタリングサービスは、ウェブブラウザとモバイルアプリで利用できます。

Webブラウザ	推奨	対応バージョン
Google Chrome	O	111以降
Mozilla FireFox	X	最新バージョン
Edge	X	最新バージョン
Safari	X	最新バージョン

ⓘ サポート案内

- ブラウザの互換性および性能の観点から、Chrome の最新バージョンの使用を推奨します。
- ユーザーインターフェース（User Interface, UI）は HTML5 標準技術で実装されているため、Internet



Explorer はサポートしていません。

制約事項

WhaTap の Web インターフェースはモバイルブラウザをサポートしていません。モバイル端末から WhaTap にアクセスするには、Android または iOS 用の WhaTap アプリをインストールしてください。WhaTap モバイルアプリはモバイル環境に最適化されています。WhaTap モバイルアプリの詳細については、[こちらのドキュメント](#) を参照してください。

モバイルアプリ

WhaTap モバイルアプリは、Android および iOS 環境をサポートしています。リンクへ移動するか、QR コードをスキャンして WhaTap アプリをインストールできます。モバイルアプリの詳細な説明は、[次の文書](#)を参照してください。

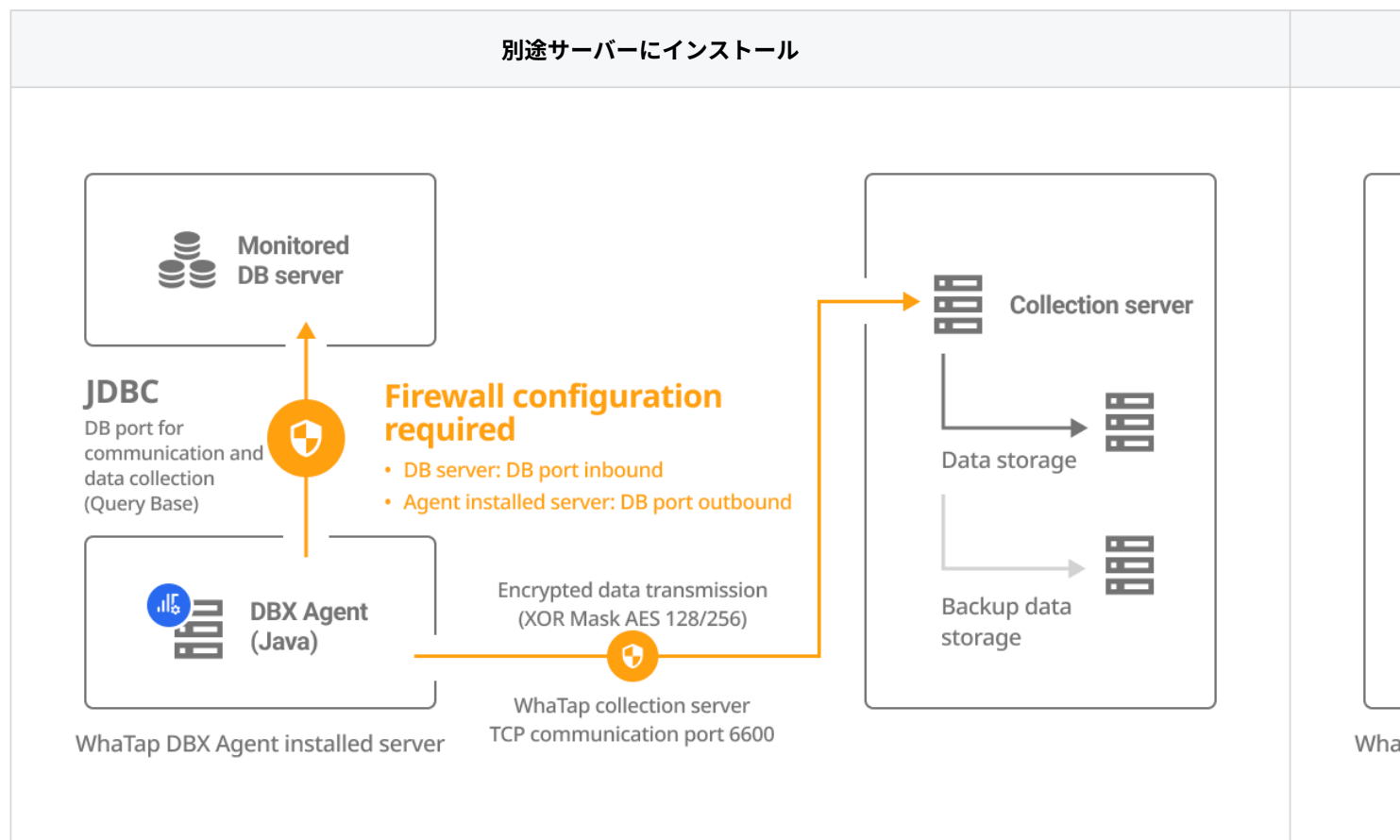
iOS	Android
	
iOS 12バージョン以降	Android 5.0バージョン以降

手動インストール

WhaTapデータベースモニタリングサービスを使用するための基本的なインストール方法について説明します。

構成図の確認

DBXエージェントは、別のサーバーにインストールするか、DBサーバーにインストールできます。インストールする方法を選択し、構成とファイアウォールを確認します。



アカウントの作成

データベースモニタリングに必要な権限を持つアカウントを作成してください。root(admin)アカウントでログインしてアカウントを作成してください。

- ① • 必要な権限を持つアカウントがすでにある場合は、このステップをスキップして[次のステップ](#)に進んでください。
- サンプルコードの DB_User は、データベースユーザーアカウント名です。使用するユーザーアカウント名に変更してください。
- サンプルコードの DB_Password には、ユーザーパスワードを入力してください。

機能と使用環境に応じて追加権限を付与してください。

- PostgreSQL 10バージョン以降

SQL

```
create user DB_User with password 'DB Password';  
grant pg_monitor to DB_User;
```

- PostgreSQL 10バージョン未満

SQL

```
create user DB_User with password 'DB Password';  
alter user DB_User with superuser;
```

- Kill sessionを使用する場合 (バージョン10以上)

データベースモニタリングでは、進行中のセッションを終了できる**Kill Session**機能を提供します。この機能を使用するには、次のように権限を設定してください。

SQL

```
grant pg_signal_backend to DB_User;
```


- テーブルサイズを収集する場合 (統計/レポート > DBサイズ > テーブルサイズ(トップ50))

SQL

```
grant usage on schema {schema_name} to DB_User;
```

- ① 個別のDBごとにコマンドを実行する必要があります。

- オブジェクト情報を照会する場合(分析 > Topオブジェクト)

SQL

```
grant select on all tables in schema {schema_name} to DB_User;
```

権限による対応機能を確認する



基本権限外の権限は、ユーザーの環境によって付与する必要はありません。ただし、一部の機能が動作しないことがあります。

権限	command	メニューと機能	詳細履歴
基本権限	grant pg_monitor to whatap;	インスタンスモニタリング / マルチインスタンス / SQL統計	grant pg_read_all_stats, pg_read_all_settings to DB_User; と同一
詳細権限	grant usage on schema {schema_name} to whatap;	DBサイズ / テーブルサイズ(トップ50)	当該DBのすべてのスキーマに対して権限が必要
	grant select on all tables in schema {schema_name} to whatap;	テーブルレイアウトを照会	-

権限	command	メニューと機能	詳細履歴
	grant pg_read_all_settings to whatap;	パラメータ照会	特定のパラメータ値が見えないことがある
セッション中止	grant pg_signal_backend to whatap;	アクティブセッション> セッション中止	PostgreSQL 9.6バージョン以下は superuser権限が必要

DBXエージェントダウンロード

DBXエージェントファイルをダウンロードします。次の2つの方法を利用します。

- `wget` コマンドを使用してダウンロードできます。

BASH

```
wget -O whatap.agent.database.tar.gz "https://service.whatap.io/download/dbx_agent?type=postgresql&format=tar.gz"
```

- コマンドでダウンロードできない場合は、WhaTapモニタリングサービス画面で**ダウンロード**ボタンを選択します。

DBXエージェントファイルをダウンロードして圧縮を解凍してください。

BASH

```
tar -zxvf whatap.agent.database.tar.gz
```

❗ セキュリティ設定によりTAR形式のファイルをダウンロードできないユーザーのためにZIP形式のファイルも一緒に提供します。インストール画面で**zip ダウンロード**ボタンを選択します。

DBXエージェントの設定

エージェントのインストール画面のガイドに従って、エージェントの設定を進めます。画面に各項目ごとに必要な情報を入力すると、設定内容とコマンドが自動的に完成されます。

✔ DBXエージェントの構成ファイルで**setup.sh**ファイルを実行すると、エージェントの設定が簡単に行われます。詳細については、[次の文書](#)を参照してください。

whatap.conf設定

圧縮を解除したフォルダに移動し、**whatap.conf**ファイルを確認します。**whatap.conf** ファイルに、プロジェクトのアクセスキー情報、WhaTapサーバー情報、DB接続情報を入力します。

whatap.conf

```
license={AccessKey}
whatap.server.host=13.124.11.223/13.209.172.35 # WhaTap server information
dbms=postgresql
db=postgres
db_ip={DB_Server_IP}
db_port={DB_Server_Port}
```

JDBCダウンロード

圧縮ファイルを解凍したパスのjdbcフォルダーにデータベースサーバーのオペレーションシステムとバージョンに合うJDBCドライバーをダウンロードします。 `/圧縮を解凍したフォルダ/jdbc`

❗ PostgreSQL 9.2バージョン以降：<https://jdbc.postgresql.org/download>

❗ /解凍されたフォルダ/jdbc/README.mdファイルで各データベースのJDBCドライバのインストールパスを確認することができます。

DBユーザーの作成とパスワードの変更

データベース接続用に暗号化されたUIDを作成します。usernameとpasswordを入力し、シェルスクリプト(またはバッチファイル)を実行してください。

✔ パスワードを変更する場合も、変更するパスワードを入力して同じ方法で進めてください。

• Linux Windows

BASH

```
./uid.sh {DB_USER} {DB_PASSWORD}
```

- ❗ 一度の設定で暗号化されたUIDを使用してモニタリング対象データベースサーバーからデータを収集します。
- DBユーザーファイルを生成するには、プロジェクトアクセスキーを`whatap.conf`ファイルに入力する必要があります。 [アクセスキーの確認](#)
- Azure Database環境の場合、DB_USER をDB_USER@DB_nameの形式で入力します。
- DB_USER または DB_PASSWORD に特殊文字が含まれている場合、特殊文字の前にEscape Character(\)を入力します。

Example

```
./uid.sh whatap whatap\!pwd
```

特殊文字が2つ以上の場合、それぞれエスケープ文字を追加

```
./uid.sh whatap whatap\!\@pwd
```

CMD

```
uid.bat "DB_USER" "DB_PASSWORD"
```

- ❗ 一度の設定で暗号化されたUIDを使用してモニタリング対象データベースサーバーからデータを収集します。
- DBユーザーファイルを生成するには、プロジェクトアクセスキーを`whatap.conf`ファイルに入力する必要があります。[アクセスキーの確認](#)
- Azure Database環境の場合、`DB_USER` を`DB_USER@DB_name`の形式で入力します。
- Windowsでは、パスワードの特殊文字にEscape Character(\)が必要ありません。ただし、パスワードに大きな引用符(")を使用する場合には必要です。

デフォルトプランを照会する

デフォルトプランユーザーファイル作成時、エージェントが事前定義されたDB情報を使用してプランを自動的に照会するように設定できます。

• Linux Windows

1. エージェント設定(`whatap.conf`)ファイルに `plan_db` オプションを追加し、DB名を入力します。

```
whatap.conf
```

```
plan_db=(dbname)
```

2. DB設定後 `planuid.sh` スクリプトを使用してユーザーとパスワードを設定します。

```
./planuid.sh user pwd
```

- ❗ ユーザー(`user`)とパスワード(`pwd`)に特殊文字が含まれている場合、各特殊文字の前にバックスラッシュ(\)を追加します。

例、 `user$example` → `user\$example`

1. エージェント設定(`whatap.conf`)ファイルに `plan_db` オプションを追加し、DB名を入力します。

```
whatap.conf
```

```
plan_db=(dbname)
```

2. DB設定後 `planuid.bat` スクリプトを使用してユーザーとパスワードを設定します。

```
planuid.bat user pwd
```

モニタリングの開始

エージェントをインストールしたパスからシェルスクリプト(またはバッチファイル)を実行してください。

- [Linux](#) [Windows](#)

```
./start.sh
```

デーモンのように使用する場合、次のコマンドを実行してください。ただし、**nohup**をインストールした環境のみで動作します。

```
./startd.sh
```

```
start.bat
```

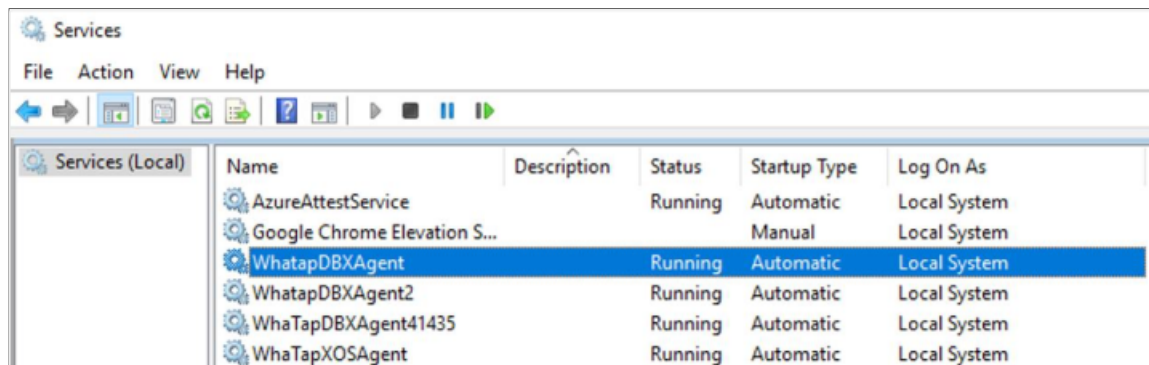
画面に表示された実行ログにエラーがないか確認し、ダッシュボード画面(**インスタンスリスト**)にチャートが正常に表示されることを確認します。異常がなければCtrl+Cを押して実行を終了し、次の手順に従ってウィンドウサービスに登録します。

サービスを登録する

次のコマンドを実行して、サービスを登録します。

```
install_WindowsService.bat create WhatapDBXAgent
```

サービスを登録すると、Windowsのサービス管理ツールで以下のように確認できます。該当サービスを探して右クリックし、スタート(Start) オプションを選択してサービスを開始します。



① コントロールパネル > Windows ツール > サービス (services.msc) で WhatapDBXAgent サービスを開始または停止できます。使用する Windows バージョンによってサービス経路は異なる場合があります。

サービスを削除する

次のコマンドを実行して、サービスを削除できます。

```
install_WindowsService.bat delete WhatapDBXAgent
```

データベースモニタリングのエージェントのインストールが完了しました。[次の文書](#)でインストール後、チェックリストを確認してください。

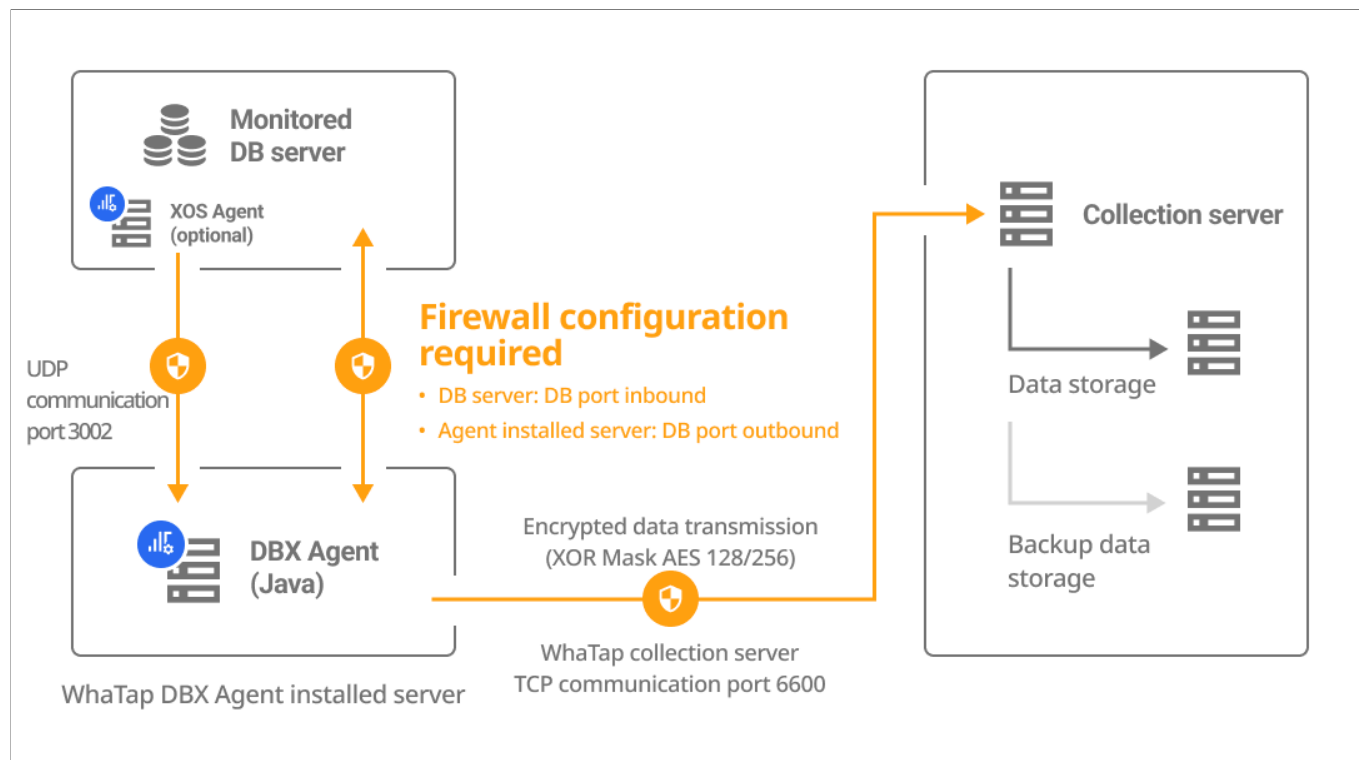
XOSエージェントのインストール及びその他のオプションを適用する

データベースサーバーのリソースをさらにモニタリングする場合は、データベースサーバーに別のXOSエージェントを実行してデータを収集できます。

- ① • x86アーキテクチャで動作するOS環境にのみ適用されます。
- 追加エージェントのインストールは、必須ではなく選択事項です。
- XOSエージェント設定オプションの詳細については、[次の文書](#)を参照してください。



- クラウドサービスが提供するモニタリングメトリクスを、データベースプロジェクトのダッシュボードで追加で収集してモニタリングするには、[次の文書](#)を参照してください。



whatap.confファイルを設定する

DBXエージェントがインストールされたパスの[whatap.conf](#)ファイルに次のオプションを設定します。

whatap.conf

```
xos=1
xos_port=3002
```

[xos](#)フォルダー(/解凍されたフォルダ/xos/)をデータベースサーバーに移動してください。

xos.confファイルを設定する

データベースサーバーに移動し、[xos](#)パスの[xos.conf](#)ファイルに次のオプションを設定します。

xos.conf

```
dbx_ip={DB_Agent_IP}  
dbx_port=3002 # default 3002  
cpu_limit=0  
mem_limit=10240
```

- ✔ エージェントインストールメニューのDB Agent IP、DB Agent Port項目にDB情報を入力すると、エージェントオプションを自動生成できます。

XOSエージェントを実行する

• Linux Windows

XOSエージェントを実行します。

```
./start.sh
```

- ❗
- DBXエージェントでモニタリングしたデータを転送するために dbx_port に設定したポート(default 3002)が開いている必要があります。(UDP Outbound)
 - XOSエージェントをバックグラウンドで実行するには*./startd.sh*ファイルを実行します。

xosフォルダの配置ファイルでエージェントを実行します。

```
start.bat
```

画面に表示される実行ログにエラーがないことを確認し、実行を終了した後、次の手順に従ってサービスに登録します。

❗ msvc120.dll、msvcr120.dll欠落時の解決方法

xosエージェント実行中msvc120.dllとmsvcr120.dllファイルが欠落して実行できない場合は、Microsoft Visual C++ 2013 Redistributable 2013がインストールされていないか、損傷しているためです。Microsoft Visual C++ 2013 Redistributableパッケージをインストールして問題を解決できます。

- ❗ 次のリンクで自分のシステム環境に合ったパッケージをダウンロードしてインストールします。
[Microsoft Visual C++ 2013 Redistributable \(x64\)](#)
- `vcredist_x64.exe`: 64ビットプログラム実行用

• サービスを登録

xosフォルダにある[install_WindowsService.bat](#)ファイルを次のように実行して、サービスに登録します。もし `create` オプションの次に名前を入力しないと、**WhatapXOSAgent**名前でサービスに基本登録されます。

```
install_WindowsService.bat create WhatapXOSAgent
```

• サービスを削除

```
install_WindowsService.bat delete WhatapXOSAgent
```

- ❗ DBXエージェントでモニタリングしたデータを転送するために `dbx_port` に設定したポート(default 3002)が開いている必要があります。(UDP Outbound)

次のステップ

インストールの点検

プロジェクトの作成やエージェントのインストール、基本エージェントのオプションまですべて適用した場合は、[次の文書](#)で点検事項を確認します。

インストール問題の解決

エージェントをインストールする際に発生する可能性がある問題と、これを解決するための具体的なガイドを提供します。詳細については、[次の文書](#)を参照してください。

エージェント設定

エージェント設定([whatap.conf](#))ファイルにオプションを適用して、モニタリングのための様々な機能を提供します。詳細については、[次の文書](#)を参照してください。

データベースのサーバーリソースを追加で監視するには、追加のエージェント(XOS)に追加オプションを設定します。詳細については、[次の文書](#)を参照してください。

クラウド設定

クラウドサービスが提供するモニタリングメトリクスを、データベースプロジェクトのダッシュボードで追加で収集してモニタリングするには、[次の文書](#)を参照してください。

モニタリングの開始

すべての設定を完了したら、エージェントがデータベースサーバーのメトリクスに対する情報収集を開始します。まず、**インスタンスリスト**メニューでモニタリングデータが収集されていることを確認します。**インスタンスリスト**の詳細については、[次の文書](#)を参照してください。

パッケージマネージャーインストール

パッケージマネージャーを利用したWhaTapデータベースモニタリングエージェントのインストール方法をご案内します。

✔ インストール方法の選択

- apt パッケージマネージャーを使用するDebian/Ubuntu系システムの場合は、このドキュメントを参照してください。
- Docker環境では、[Dockerインストール](#)を参照してください。
- その他の環境では、[手動インストール](#)を参照してください。

❗ サポートされているデータベース

MySQL、PostgreSQL、SQL Server

WhaTapモニタリングサービスを利用するには、[サインアップ](#)後にプロジェクトを作成し、モニタリング対象のサーバーにエージェントをインストールする必要があります。サインアップの詳細については、[次のドキュメント](#)を参照してください。

プロジェクト作成

エージェントをインストールする前にプロジェクトを作成してください。

1. [WhaTapモニタリングサービス](#)にログインしてください。
2. 左側のサイドメニューで **プロジェクト全体** > **+ Project** ボタンをクリックします。
3. **商品選択** 画面でインストールする製品を選択します。
4. 以下の項目を入力または選択します。
 - **プロジェクト名**：プロジェクト名を入力します。
 - **データサーバーリージョン**：データサーバーの設置されているリージョンを選択します。リージョンとは、クラウドサービスを提供するデータセンターの集合体です。選択したリージョンにユーザーデータ

が保存されます。

- **プロジェクトタイムゾーン**：アラートおよびレポート生成時の基準となる時刻を設定します。
- **通知言語の設定**：アラート通知メッセージの言語を設定します。（韓国語、英語に対応）
- **プロジェクトグループ**：複数のプロジェクトをグループとしてまとめて管理できます。所属するグループがある場合は選択してください。
- **プロジェクト説明**：プロジェクトに関する補足説明や詳細情報を入力します。

5. すべての設定が完了したら、**プロジェクトを作成する** ボタンをクリックします。

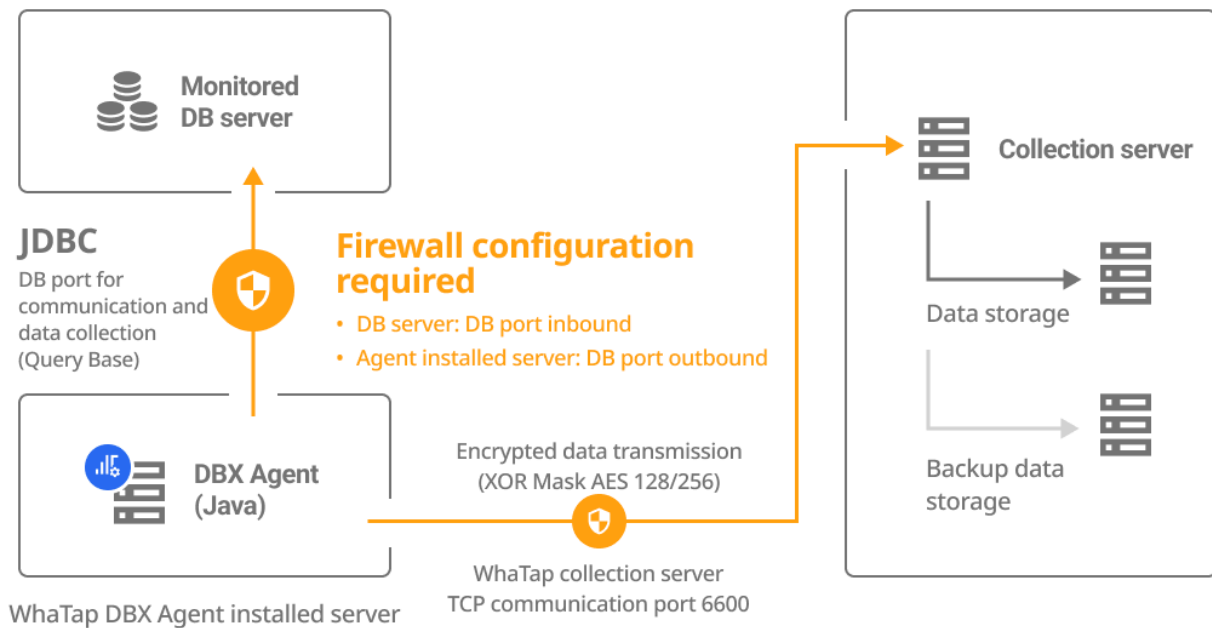
① 組織を選択してプロジェクトを追加する場合は、**組織下位のグループ**を必須に設定する必要があります。

グループに関する詳細は、[次のドキュメント](#)を参照してください。

構成図の確認

DBXエージェントは、別のサーバーにインストールするか、DBサーバーにインストールできます。インストールする方法を選択し、構成とファイアウォールを確認します。

別途サーバーにインストール



アカウントの作成

データベースモニタリングに必要な権限を持つアカウントを作成してください。root(admin)アカウントでログインしてアカウントを作成してください。

- ① 必要な権限を持つアカウントがすでにある場合は、このステップをスキップして次のステップに進んでください。
- サンプルコードの `DB_User` は、データベースユーザーアカウント名です。使用するユーザーアカウント名に変更してください。
- サンプルコードの `DB_Password` には、ユーザーパスワードを入力してください。

機能と使用環境に応じて追加権限を付与してください。

- PostgreSQL 10バージョン以降

SQL

```
create user DB_User with password 'DB Password';
grant pg_monitor to DB_User;
```

- PostgreSQL 10バージョン未満

SQL

```
create user DB_User with password 'DB Password';
alter user DB_User with superuser;
```

- Kill sessionを使用する場合 (バージョン10以上)

データベースモニタリングでは、進行中のセッションを終了できる**Kill Session**機能を提供します。この機能を使用するには、次のように権限を設定してください。

SQL

```
grant pg_signal_backend to DB_User;
```

- テーブルサイズを収集する場合 (統計/レポート > DBサイズ > テーブルサイズ(トップ50))

SQL

```
grant usage on schema {schema_name} to DB_User;
```

① 個別のDBごとにコマンドを実行する必要があります。

- オブジェクト情報を照会する場合(分析 > Topオブジェクト)

SQL

```
grant select on all tables in schema {schema_name} to DB_User;
```

① 権限による対応機能を確認する



基本権限外の権限は、ユーザーの環境によって付与する必要はありません。ただし、一部の機能が動作しないことがあります。

権限	command	メニューと機能	詳細履歴
基本権限	<code>grant pg_monitor to whatap;</code>	インスタンスモニタリング / マルチインスタンス / SQL統計	<code>grant pg_read_all_stats, pg_read_all_settings to DB_User;</code> と同一
詳細権限	<code>grant usage on schema {schema_name} to whatap;</code>	DBサイズ / テーブルサイズ(トップ50)	当該DBのすべてのスキーマに対して権限が必要
	<code>grant select on all tables in schema {schema_name} to whatap;</code>	テーブルレイアウトを照会	-
	<code>grant pg_read_all_settings to whatap;</code>	パラメータ照会	特定のパラメータ値が見えないことがある
セッション中止	<code>grant pg_signal_backend to whatap;</code>	アクティブセッション > セッション中止	PostgreSQL 9.6バージョン以下は superuser権限が必要

アクセスキーの確認

アクセスキーは、WhaTapサービスを有効にするための固有IDです。

インストール手順セクションでプロジェクトアクセスキーの発行ボタンをクリックします。アクセスキーを自動的に発行した後、次のステップを進みます。

❗ プロジェクトを作成すると、自動的にエージェントインストールページに移動します。エージェントインストールページが移動しない場合は、画面左側のメニューから管理 > エージェントインストールを選択します。

エージェントのインストール

WhaTapリポジトリの追加

次のコマンドを実行して、WhaTap公式リポジトリをシステムに追加します。

```
curl -s https://repo.whatap.io/debian/release.gpg | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/whatap-release.gpg > /dev/null
echo "deb [signed-by=/etc/apt/trusted.gpg.d/whatap-release.gpg] https://repo.whatap.io/debian unstable/" | sudo tee /etc/apt/sources.list.d/whatap.list
sudo apt-get update
```

パッケージのインストール

次のコマンドでWhaTapデータベースモニタリングエージェントをインストールします。

```
sudo apt-get install whatap-db
```

エージェントの設定と起動

設定ファイルを作成し、エージェントを起動します。

```
sudo bash -c 'cat > /usr/whatap/db/whatap.conf <<EOF
license=ライセンスキー
whatap.server.host=13.124.11.223/13.209.172.35
dbms=mysql'
```

```
db_ip=127.0.0.1
db_port=3306
EOF'
cd /usr/whatap/db && sudo /usr/whatap/db/uid.sh whatap whatap && cd -
sudo service whatap-db restart
```

- ❗ • `whatap.conf` ファイルに必要なデータベースオプションを追加できます。
- `uid.sh` スクリプトのパラメータ値(データベースアカウントIDおよびパスワード)を実際の環境に合わせて修正してください。

エージェントの管理

エージェントのアップデート

エージェントを最新バージョンにアップデートするには、次のコマンドを実行します。

```
sudo apt-get update
sudo apt-get install whatap-db
sudo service whatap-db restart
```

エージェントの停止とパッケージの削除

WhaTapデータベースモニタリングエージェントを削除するには、次のコマンドを実行します。

```
sudo service whatap-db stop
sudo apt-get remove whatap-db
```

エージェントログの確認

エージェントログを確認するには、次のコマンドを使用します。

```
sudo service whatap-db log
```

Dockerインストール

Dockerコンテナを利用したWhaTapデータベースモニタリングエージェントのインストール方法をご案内します。

✔ インストール方法の選択

- Docker環境で使用する場合は、このドキュメントを参照してください。
- Debian/Ubuntuシステムでは、[パッケージマネージャーインストール](#)を推奨します。
- その他の環境では、[手動インストール](#)を参照してください。

ⓘ サポートされているデータベース

MySQL、PostgreSQL、SQL Server

WhaTapモニタリングサービスを利用するには、[サインアップ](#)後にプロジェクトを作成し、モニタリング対象のサーバーにエージェントをインストールする必要があります。サインアップの詳細については、[次のドキュメント](#)を参照してください。

プロジェクト作成

エージェントをインストールする前にプロジェクトを作成してください。

1. [WhaTapモニタリングサービス](#)にログインしてください。
2. 左側のサイドメニューで **プロジェクト全体** > **+ Project** ボタンをクリックします。
3. **商品選択** 画面でインストールする製品を選択します。
4. 以下の項目を入力または選択します。
 - **プロジェクト名**：プロジェクト名を入力します。
 - **データサーバーリージョン**：データサーバーの設置されているリージョンを選択します。リージョンとは、クラウドサービスを提供するデータセンターの集合体です。選択したリージョンにユーザーデータが保存されます。
 - **プロジェクトタイムゾーン**：アラートおよびレポート生成時の基準となる時刻を設定します。

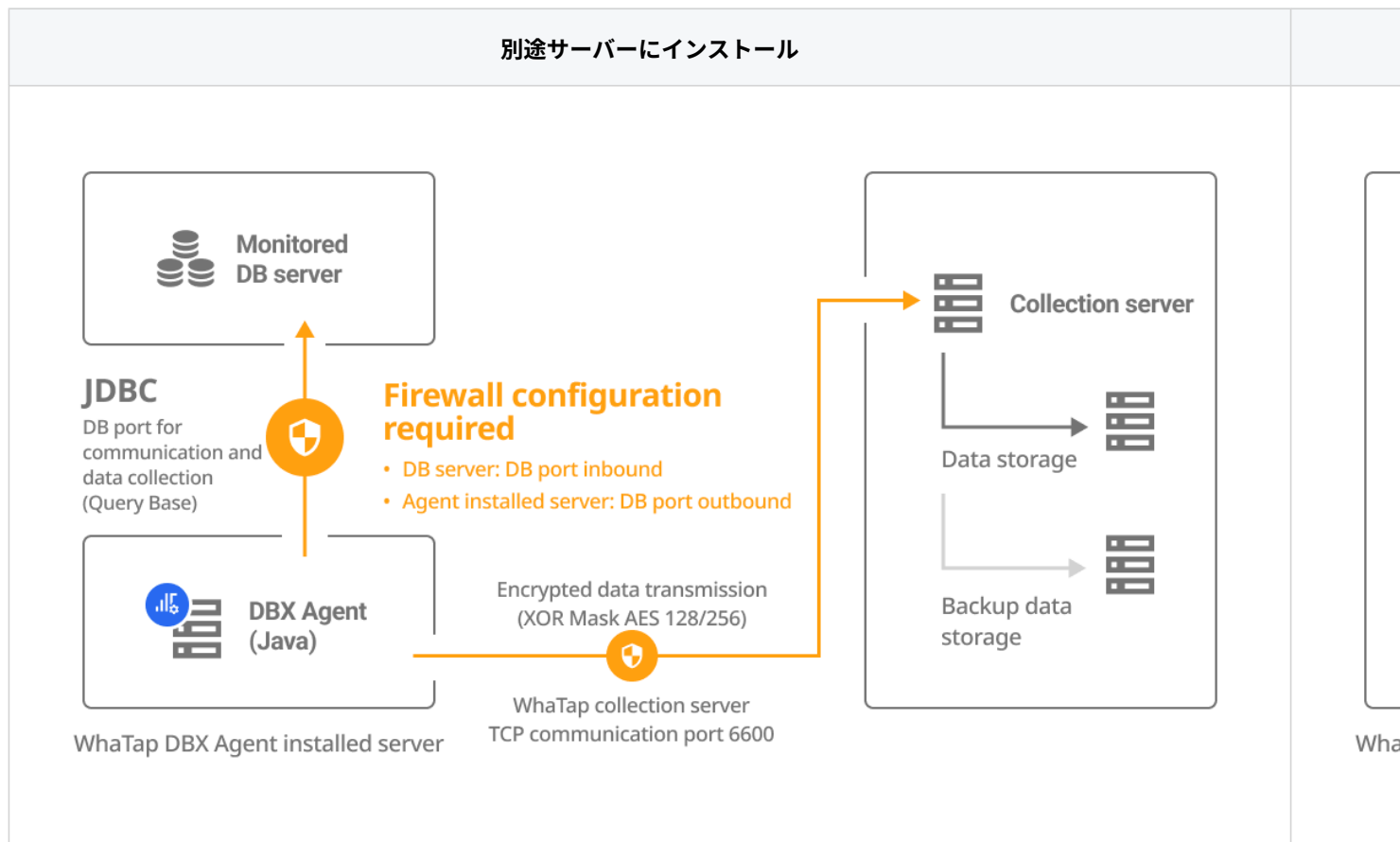
- **通知言語の設定**：アラート通知メッセージの言語を設定します。（韓国語、英語に対応）
- **プロジェクトグループ**：複数のプロジェクトをグループとしてまとめて管理できます。所属するグループがある場合は選択してください。
- **プロジェクト説明**：プロジェクトに関する補足説明や詳細情報を入力します。

5. すべての設定が完了したら、**プロジェクトを作成する** ボタンをクリックします。

❗ 組織を選択してプロジェクトを追加する場合は、**組織下位のグループ**を必須に設定する必要があります。
グループに関する詳細は、[次のドキュメント](#)を参照してください。

構成図の確認

DBXエージェントは、別のサーバーにインストールするか、DBサーバーにインストールできます。インストールする方法を選択し、構成とファイアウォールを確認します。



アカウントの作成

データベースモニタリングに必要な権限を持つアカウントを作成してください。root(admin)アカウントでログインしてアカウントを作成してください。

- ① 必要な権限を持つアカウントがすでにある場合は、このステップをスキップして次のステップに進んでください。
- サンプルコードの `DB_User` は、データベースユーザーアカウント名です。使用するユーザーアカウント名に変更してください。
- サンプルコードの `DB_Password` には、ユーザーパスワードを入力してください。

機能と使用環境に応じて追加権限を付与してください。

- PostgreSQL 10バージョン以降

SQL

```
create user DB_User with password 'DB Password';
grant pg_monitor to DB_User;
```

- PostgreSQL 10バージョン未満

SQL

```
create user DB_User with password 'DB Password';
alter user DB_User with superuser;
```

- Kill sessionを使用する場合 (バージョン10以上)

データベースモニタリングでは、進行中のセッションを終了できる**Kill Session**機能を提供します。この機能を使用するには、次のように権限を設定してください。

SQL

```
grant pg_signal_backend to DB_User;
```

- テーブルサイズを収集する場合 (統計/レポート > DBサイズ > テーブルサイズ(トップ50))

SQL

```
grant usage on schema {schema_name} to DB_User;
```

① 個別のDBごとにコマンドを実行する必要があります。

- オブジェクト情報を照会する場合(分析 > Topオブジェクト)

SQL

```
grant select on all tables in schema {schema_name} to DB_User;
```

① 権限による対応機能を確認する



基本権限外の権限は、ユーザーの環境によって付与する必要はありません。ただし、一部の機能が動作しないことがあります。

権限	command	メニューと機能	詳細履歴
基本権限	<code>grant pg_monitor to whatap;</code>	インスタンスモニタリング / マルチインスタンス / SQL統計	<code>grant pg_read_all_stats,pg_read_all_settings to DB_User;</code> と同一
詳細権限	<code>grant usage on schema {schema_name} to whatap;</code>	DBサイズ / テーブルサイズ(トップ50)	当該DBのすべてのスキーマに対して権限が必要
	<code>grant select on all tables in schema {schema_name} to whatap;</code>	テーブルレイアウトを照会	-
	<code>grant pg_read_all_settings to whatap;</code>	パラメータ照会	特定のパラメータ値が見えないことがある
セッション中止	<code>grant pg_signal_backend to whatap;</code>	アクティブセッション > セッション中止	PostgreSQL 9.6バージョン以下は superuser権限が必要

アクセスキーの確認

アクセスキーは、WhaTapサービスを有効にするための固有IDです。

インストール手順セクションでプロジェクトアクセスキーの発行ボタンをクリックします。アクセスキーを自動的に発行した後、次のステップを進みます。

❗ プロジェクトを作成すると、自動的にエージェントインストールページに移動します。エージェントインストールページが移動しない場合は、画面左側のメニューから管理 > エージェントインストールを選択します。

エージェントの実行

次のコマンドを実行して、WhaTapデータベースモニタリングエージェントをDockerコンテナとして起動します。

```
docker run -d --name whatap-db-agent \
-e WHATAP_CONF="
license=x6051ro88nrc9-x4hmr76jvvlfnq-z52cnv1dfa8n5c
whatap.server.host=13.124.11.223/13.209.172.35
dbms=mysql
db_ip=10.0.0.1
db_port=3306
connect_option=?useSSL=true&verifyServerCertificate=false&serverTimezone=SERVER_TIMEZONE" \
-e DB_USERNAME=mydbuser \
-e DB_PASSWORD=mypassword \
whatap/db-agent:latest
```

設定項目

必須設定

- `license` : プロジェクトアクセスキー
- `whatap.server.host` : WhaTap収集サーバーIP (13.124.11.223/13.209.172.35)
- `dbms` : データベースタイプ (mysql, postgresql, mssql)
- `db_ip` : データベースサーバーIPアドレス
- `db_port` : データベースサーバーポート番号

- `DB_USERNAME` : データベースアカウントID
- `DB_PASSWORD` : データベースアカウントパスワード

選択設定

- `connect_option` : データベース接続オプション
 - `useSSL=true&verifyServerCertificate=false` : SSLを使用してデータベースに接続する場合
 - `serverTimezone=SERVER_TIMEZONE` : JDBCドライバー接続時にサーバータイムゾーン設定が必要な場合

❗ 上記コマンドの設定値を実際の環境に合わせて修正してください。

エージェント動作確認

次のコマンドでエージェントが正常に動作しているか確認します。

```
docker logs -f whatap-db-agent
```

エージェントの削除

次のコマンドを実行して、WhaTapデータベースモニタリングエージェントコンテナを停止し、削除します。

```
docker stop whatap-db-agent && docker rm whatap-db-agent
```

インストールチェックリスト

データベースモニタリングのインストールを完了しました。次の項目をチェックして正しくインストールされているか確認してください。

エージェントのインストールチェック

データベースエージェントをインストールしたパスから、[start.sh](#) シェルスクリプトファイルまたは [start.bat](#) バッチファイルを実行します。エージェントを実行すると、データベースのモニタリング情報の収集が開始されます。シェルスクリプトを実行した後、ターミナルにWhaTapロゴが出力されている場合は、正常に実行されている状態です。

start.sh

```
$ ./start.sh
```

```
 _ _ _ _ _
|||/||| _/_ _/_ _ _
|||/_V_`|||/_/_\
|_|/_||/_\,||/_\,/_\
      |/_
```

Just Tap, Always Monitoring

WhaTap DBX version X.Y.Z YYYYMMDD

- ❗ デーモンのように使用したい場合は、`startd.sh` コマンドを実行してください。ただし、**nohup**をインストールされている環境のみで動作します。

ログの確認

- [whatap.log](#)

エージェントの起動および収集に関するすべてのログは、`$WHATAP_HOME /logs/whatap.log` ファイルで確認できます。

毎日0時(UTC)になると、既存の[whatap.log](#) ファイルは `whatap_ yymmdd .log` 形式に自動的にリネームされて日付

別に保存され、新しいwhatap.logファイルが作成されてログが記録されます。

日付別に保存されたログファイルは、7日が経過すると自動的に削除されます。

- xos.log

XOSの起動に関するログは、\$XOSのインストール先 /xos yymmdd .logファイルに記録され、日付別に自動生成されます。

エージェントプロセスの確認

次のコマンドを使用して、動作中のWhaTap DBXエージェントプロセスを確認できます。

```
$ ps -ef | grep dbx
```

モニタリングの確認

サーバーで正常にログが確認され、コンソールに正しく登録されたことを確認するには、WhaTapモニタリングサービスの初期画面に移動します。確認するプロジェクトを選択し、画面の左側の管理 > エージェントリストメニューを選択します。エージェントリストからデータベースサーバーの名前を確認できます。データベースサーバーの名前を別途設定する場合、次の文書を参照してください。

エージェントリスト											
Csv Download											
Agent Name, OID											
ステータス	名前 (初期)	ID	開始時間	活性 / 非活性	エージェントバージョン	Agent IP	DB IPアドレス	DB ポート	DB バージョン	CPU コア	
● 活性	DBX-2-60-6379	2122651642	2024-03-13 18:36:35	2024-03-13 18:36:35	1.6.32 20240306	172.17.0.1	192.168.2.60	6379	7.0.11	1	📄 🗑
● 活性	DBX-2-60-6380	-2134789231	2024-03-13 18:36:10	2024-03-13 18:36:10	1.6.32 20240306	172.17.0.1	192.168.2.60	6380	7.0.11	0	📄 🗑
● 活性	DBX-2-60-6381	-137985273	2024-03-13 18:36:15	2024-03-13 18:36:15	1.6.32 20240306	172.17.0.1	192.168.2.60	6381	7.0.11	0	📄 🗑
● 活性	DBX-2-60-6382	1859113661	2024-03-13 18:36:20	2024-03-13 18:36:20	1.6.32 20240306	172.17.0.1	192.168.2.60	6382	7.0.11	0	📄 🗑

インストール問題の解決

データベースエージェントのインストール時に発生する問題を確認してください。

ファイアウォール設定の確認

ファイアウォールのブロックを解除してください。WhaTapサーバーのTCPアウトバウンドファイアウォールが設定されていると、モニタリング情報をサーバーに転送できません。

ファイアウォールを確認する方法

telnet コマンドを実行し、接続関連情報をが正常に表示します。(telnet[サーバーIP][サーバーポート])

Example

```
$ telnet 52.193.60.176 6600
Trying 52.193.60.176...
Connected to 52.193.60.176.
Escape character is '^]'.
```

❗ WhaTapの収集サーバー情報は、**管理** > **エージェントインストール**メニューから確認できます。

制約事項

複数リージョンを1つのプロジェクトとして選択することはできません。

- [WhaTapモニタリングサービス](#)サイトでプロジェクトを作成する時にリージョンは重複選択できず、複数のリージョンを活用する場合は別途のプロジェクトを作成する必要があります。
- クラウド環境に複数のリージョン(Region)にサーバが存在する場合、ネットワークのlatencyなどパフォーマンス情報収集の制約を回避するために、リージョン(Region) 単位で収集サーバをインストールする必要があります。

- WhaTapは、顧客のニーズを満たすためにリージョン(Region)ごとに収集サーバーを構築します

エージェント名(ONAME) は重複使用できません

WhaTap DBXモニタリングは、エージェントを認識するために、エージェントのIPアドレスとポート(Port)情報を使用します。ユーザー環境でモニタリングするデータベースサーバーが同一IP、同一ポート(Port)を使用している場合、WhaTapサーバーのサーバーインスタンスを区分できません。

- サーバーインスタンスが動的に容量を可変に拡張される環境でない場合は、[次の文書](#)を参照してデータベース識別用の名前を直接変更することで、サーバーインスタンスを迂回できます。
- モニタリングするデータベースのinternal addressまたはネットワーク仮想化により重複したIPを使用している場合は、[次の文書](#)を参照して、別のデータベース名のパターンを使用してください。

DB権限及びエージェント設定

DB権限及びエージェントの設定によってデータが照会されないメニューがあります。次の内容を参考にして、必要な権限を付与し、エージェントのオプションを設定してください。

❗ モニタリングのアカウントを whatap で生成した場合の例です。
`create user whatap with password 'DBPassword';`

メニュー	DBオプション及び権限	エージェント設定	備考
基本メニュー	<code>grant pg_monitor to whatap;</code>	-	DBXエージェントv1.6.15以降
ダッシュボード>スロークエリ	DBパラメータ： <code>log_min_duration_statement=500</code> (ex. 500ms以上かかるSQL記録) DB権限 <code>grant PG_READ_ALL_SETTINGS to whatap;</code>	xos.conf <code>slow_query_interval=30</code> (既定値30秒) whatap.conf <code>slow_query_log=ON</code>	-

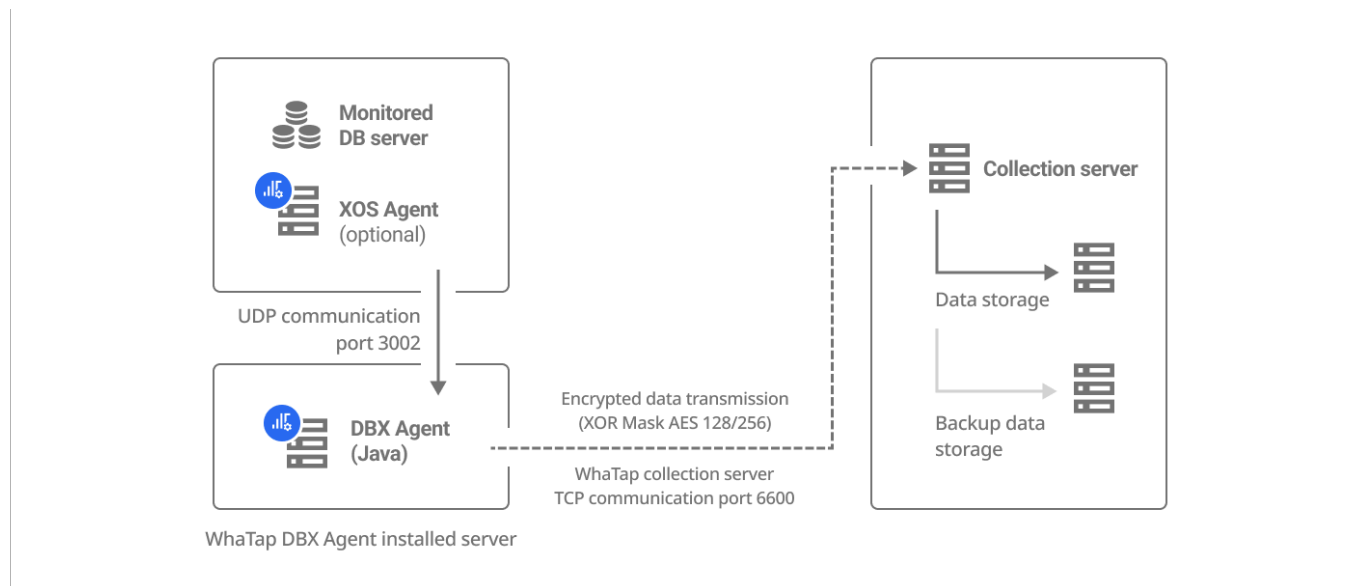
メニュー	DBオプション及び権限	エージェント設定	備考
インスタンス > セッション中止	<code>grant pg_signal_backend to whatap;</code>	-	-
分析 > Top オブジェクト	<code>grant select on all tables in schema {schema_name} to whatap;</code>	<code>whatap.conf</code> <code>pg_object=true</code>	DBXエージェントv1.6.15以降 Bloating 、 Age タブの一覧は、権限及びエージェントオプションが設定されていてもしきい値(size、ratio)を超えない場合、データが照会されない可能性があります。
統計/レポート > データベースサイズ	<code>grant usage on schema {schema_name} to whatap;</code>	<code>whatap.conf</code> <code>tables=true</code>	データベースサイズメニュー画面の下のテーブルサイズを照会するために必要な権限です。
オブジェクト情報 (Object detail) モーダルウィンドウ	<code>grant select on all tables in schema {schema_name} to whatap;</code>	-	Top オブジェクト, データベースサイズ, SQL詳細(Plan) メニューでテーブル名をクリックすると表示されるポップアップウィンドウです。権限がない場合、Column listが照会されません。

設定

このページでは、WhaTapエージェントの構成について説明し、各種機能の設定方法について解説します。また、モニタリング対象となるDBサーバーのリソースを監視するためのオプション機能についても紹介します。

エージェント構成

以下は、WhaTapエージェントの構成図です。



• 収集サーバー

エージェントが収集したデータベースのパフォーマンスデータを収集および保存、統計情報を抽出し、これをユーザーに効率的な形式で提供します。収集サーバーはリージョン（Region）別に設定できます。リージョンごとに収集サーバーのアドレスが異なるように割り当てます。ユーザーが選択したリージョンによって収集サーバーのアドレスは異なる場合があります。プロジェクトを作成する際、リージョンも設定します。

• DBX エージェント(デフォルトエージェント)

Queryベースで性能データを収集し、サーバーに送信します。モニタリング対象データベースサーバーにエージェントを直接インストールしなくても、別のエージェントサーバーを使用してモニタリングすることができます。

• XOSエージェント(追加エージェント)

XOSエージェントは、データベースサーバーのプロセス使用量をモニタリングできる追加のオプションエージェントです。データベースサーバーのプロセス使用量をモニタリングしたい場合は、データベースサーバーにXOSエージェントを実行してデータを収集できます。

① XOSエージェントは、x86アーキテクチャベースで動作するOS環境のみ適用できます。

• ネットワーク

- 基本エージェント：WhaTapのモニタリングエージェントは、収集したモニタリング情報をWhaTap収集サーバーに転送するために外部通信（TCP）6600番ポートを使用します。
- XOSエージェント：エージェントをインストールしたサーバーとデータベースサーバーの間に外部通信（UDP）3002番ポートを使用します。ポートが重複する場合は、`dbx_port` オプションを使用してポートを変更します。

エージェント設定

エージェント設定の詳細については、次の文書を参照してください。

迅速なエージェント設定



DBXエージェントの構成ファイルの中で`setup.sh`ファイルを実行すると、エージェントの設定を迅速かつ簡単に進めることができます。



DBXエージェントの設定

`whatap.conf`ファイルで設定できるオプションについて説明します。

 XOSエージェント設定

データベースサーバーのリソースを追加でモニタリングするための設定方法を提供します。

 クラウドエージェント設定

4 項目

迅速なエージェント設定

DBXエージェントの構成ファイルで`setup.sh`ファイルを実行すると、エージェントの設定が簡単に行われます。

ファイルの圧縮を解凍する

ダウンロードしたDBXエージェントファイルの解凍します。

BASH

```
tar -zxvf whatap.agent.database.tar.gz
```

setup.shを実行する

`setup.sh`ファイルを実行し、画面の案内に従ってエージェントの設定及びDBサーバーの情報入力、JDBCダウンロード、モニタリング用のアカウント作成及び権限設定を行います。必要な情報を入力すると、すべての設定を自動的に完了できます。

BASH

```
./setup.sh
```

```
Hello. This is a monitoring configuration file helper. If an error occurs during the process, please take action according to the instructions.

**Proceed to the next step**
=====
1.Register WhaTap server information
2.Register WhatTap server access key
3.Register DB server information (ip/port check)
4.Download related files (id/pwd check)
5.Register a monitoring account
=====

if you want to stop setup prees 'q' or 'quit'

## STEP1 Register WhaTap server information
Whatap > Management > Agent Installation > 1. Check Whatap Server ip and port
WhaTap Server Ip?
```

❗ エージェントの設定を中止する場合は、「q」または「quit」を入力し、Enterキーを入力します。

STEP1 Register WhaTap server information

BASH

```
WhaTap Server Ip?  
WhaTap Server Port (default 6600)?
```

WhaTapのサーバー情報を入力します。管理 > エージェントインストールメニューのDBXエージェントの設定セクションからWhaTapのサーバーIPアドレスとポート番号が確認できます。

WhaTapサーバーポートの既定値は6600です。何も入力せずにEnterキーを入力すると、既定値に設定されます。

STEP2 Register WhaTap server access key

BASH

```
WhaTap Project Access Key?
```

プロジェクトアクセスキーを入力します。管理 > エージェントインストールメニューのアクセスキーの確認セクションでコピーボタンを選択して貼り付けます。

正確なアクセスキーを入力すると、そのプロジェクトのコードが出力されます。

STEP3 Register DB server information (ip/port check)

BASH

```
Monitoring DBMS type?  
Monitoring DBMS Server Ip?  
Monitoring DBMS Server Port?
```

モニタリングするDBサーバーの情報を入力します。

- **Monitoring DBMS type**：画面に表示されているデータベースの種類を選択します。
- **Monitoring DBMS Server IP**：データベースサーバーのIPアドレスを入力します。
- **Monitoring DBMS Server Port**：データベースサーバーのポート番号を入力します。

❗ データベースサーバーのIPアドレスとポート番号を入力してから、接続状態を検証します。間違った情報を入力した場合は、IPアドレスとポート番号を再度入力します。

STEP4 Download jdbc driver

DBXエージェントのパスにDBサーバーのOSとバージョンに合わせてJDBCドライバーを自動的にダウンロードします。

STEP5 Register a monitoring account

BASH

```
Monitoring Database Name?  
Monitoring User ID?  
Monitoring User \"{user_id}\"'s Password?  
Connecting DBMS Server with SSL (default false)?
```

DB接続のために暗号化されたUIDを生成する手順です。 [アカウント作成](#) セクションで作成したアカウント情報を入力します。

- **Monitoring Database Name** : モニタリングするデータベースの名前を入力します。
- **Monitoring User ID / Monitoring User {user_id}'s Password** : アカウントIDとパスワードを入力します。パスワードは暗号化し、保存されます。
- **Connecting DBMS Server with SSL** : SSLプロトコルを利用してデータベースサーバーに接続する場合、 `true` を入力します。既定値は `false` です。

エージェントを実行する

`setup.sh` ファイルを利用してエージェントのインストールプロセスを完了したら、データベース情報と `whatap.conf` ファイルに設定した内容が出力されます。

```

***** DATABASE INFO *****
jdbc:postgresql://192.168.1.203:5432/postgres
DatabaseMajorVersion = 10
DatabaseMinorVersion = 4
DatabaseProductName = PostgreSQL
DatabaseProductVersion = 10.4
DefaultTransactionIsolation = 2
DriverMajorVersion = 42
DriverMinorVersion = 1
*****

/Users/hjhang/ios1/dbx_setup/./whatap.conf saved

***** whatap.conf *****
whatap.server.host=15.165.146.117
whatap.server.port=6600
license=+4Br+21p20K1a-x32011st093dne-x3hu4dp1298w23
dbms=postgresql
db_ip=192.168.1.203
db_port=5432
db=postgres
db_user=whatap
db_pwd={{T+K40w5L30K/4w-}}
db_ssl=false
*****

=====The file exists=====

Do you want to start the agent as a daemon? ([y]/n)y

The agent was started as a daemon.
Please check the log [/Users/hjhang/ios1/dbx_setup/logs/whatap.log].

```

エージェントをデーモンのように使用する場合は、最後の項目で「y」を入力します。

次のステップ

- エージェント設定が完了したら、[次の文書](#)を参照の上、インストール後のチェックリストを確認します。
- 追加のエージェント(XOS)をインストールし、DBサーバーのリソースをモニタリングするには、[次の文書](#)を参照します。

DBXエージェントの設定

DBXエージェントに必要な設定は、`whatap.conf`ファイルに作成されます。設定できるオプションについては、次を参照してください。

基本オプション

- `license` `String`

エージェントのインストール時にサーバーから付与されたプロジェクトアクセスキーを設定します。プロジェクトアクセスキーには、エージェントが属するプロジェクトとセキュリティ通信のためのパスワードキーが含まれています。

- `whatap.server.host` `String`

既定値 `127.0.0.1,127.0.0.1`

エージェントが収集したデータを送信するWhaTap収集サーバーのIPアドレスを設定します。収集サーバーの冗長化で2つ以上のIPを所有する場合は、コンマ(,)を区切り文字として使用します。設定したIPアドレスには、収集サーバーproxyデーモンがリスニング状態でサービスされる必要があります。

- `whatap.server.port` `String`

既定値 `6600`

収集サーバーのポート(port)を設定します。ポートは1つのみ設定ができるので、`whatap_server_host` に設定した収集サーバーは同一ポートを使用する必要があります。

- `dbms` `String`

`postgresql`

⚠ 別のオプション値を入力すると、モニタリングのデータ収集が正常に動作しない場合があります。

- `db` `String`

接続するデータベース名です。

- `db_ip` `String`

モニタリングするデータベースサーバーのIPアドレスを設定します。

- **db_port** String

データベースが通信に使用するポート番号を設定します。

- **aws_iam_auth** Boolean

既定値 false

AWS RDS・Aurora(MySQL、MariaDB、PostgreSQL)環境で、DBパスワードの代わりにAWS IAM認証トークンで接続するオプションです。true に設定するとIAM認証トークンで接続し、既定値の false の場合は既存のパスワード方式で接続します。

① aws_arn オプションの値によっては、whatap.conf にAWSアクセスキー(access key)とシークレットキー(secret key)を直接入力することもできます。

データ収集

- **db_param_enabled** Boolean

既定値 true

DBパラメータの収集有無を設定します。

- **conn_fail_count** Int

既定値 15

15回連続コネクションエラーが発生した時に通知を送信します。希望する回数を入力して、通知の発生基準を調整できます。

- **long_run_session_sec** Second

既定値 10 (単位：秒)

設定した時間より長く実行されているセッションを、long runセッションで処理します。

- **tables** Boolean

既定値 true

テーブルサイズ情報を収集します。統計/レポート > DBサイズメニューでそのデータベースを選択すると、テーブル

サイズ情報が確認できます。

- **tables_hour** Int

既定値 5

テーブルサイズ情報を保存するために、情報を収集する時刻を設定します。既定値は 5 であり、午前5時に収集を始めます。

- **tables_min_size** Int

既定値 10

テーブル情報を収集する場合、設定した値より小さいサイズのテーブル情報は収集しません。テーブルデータを収集する最小単位は、 `pg_total_relation_size` 既定値では10MBです。

- **replication_name** String

レプリケーション名を設定すると、同じ `replication_name` を持つDBの間にMaster/Slave関係があるものとみなします。

レプリケーションセットを定義するオプションとしてマスターノードとスレーブノードの両方に同じ値の `replication_name` オプションを設定すると**インスタンスモニタリング**でツリー構造で表示されます。また、このオプション値を基準に**Flexボード**でレプリケーション状態をモニタリングするウィジェットを提供します。

- **skip_schema** String

既定値：pg_catalog、sys、information_schema、pg_toast

オプション値に設定したスキーマのテーブルは収集しません。

- **skip_db** String

既定値：edb、postgres

オプション値に設定したデータベースのテーブルは収集しません。

- **slave_ip** string

IPアドレスの確認は、 `select inet_server_addr();` とします。マスターノードから `pg_stat_replication.client_addr` を検索したときに表示されるIPアドレスのいずれか1つでなければなりません。
`replication_name` オプション値が同じでもSlave DBである場合は、 `slave_ip` を指定する必要があります。

① RDSの場合、IPアドレスの代わりに `pg_replication_slots.slot_name` でMaster/Slave関係が設定されるため、 `slave_ip` オプションを設定する必要はありません。

- **statements_min_row** Int

既定値 10000 (row)

PG SQL統計メニューで使用するデータ収集の基準です。pg_stat_statements を照会してrows値が statements_min_row より大きい場合に収集します。

❗ PG SQL統計メニューについての詳細は、[次の文書](#)を参照してください。

- **skip_whatap_session** Int

既定値 0

1 の場合、WhaTapがデータ収集で実行するクエリのセッションをactive session listに表示しません。(既定値：0)

- **xview_min_sec** Int

既定値 0

xview収集時、クエリの実行時間が設定された時間よりも長くなる場合にのみ収集します。デフォルトとしては、実行時間が0秒以上のすべてのクエリを収集します。

Topオブジェクトの使用設定

次はTopオブジェクトに関する情報を照会するためのエージェント設定です。Topオブジェクト詳細については、[次の文書](#)を参照してください。

- **pg_object** Boolean

既定値 true

Topオブジェクト情報の収集有無を設定します。情報を収集するには true に変更してください。

- **pg_object_hour** Int

既定値 5

Topオブジェクト情報の収集時間を設定します。既定値は 5 で、午前5時に収集を始めます。

- **table_bloat_ratio** Percentage

既定値 50

テーブルbloating ratio値が設定値以上の場合は、情報を収集します。既定値は50%です。

- **table_bloat_bytes** Byte

既定値 10485760

テーブルbloatingbytes値が設定値以上の場合、情報を収集します。既定値は10MBです。

- **index_bloat_ratio** Percentage

既定値 50

インデックスbloatingratio値が設定値以上の場合、情報を収集します。既定値は50%です。

- **index_bloat_bytes** Byte

既定値 10485760

インデックスbloating bytes値が設定値以上の場合、情報を収集します。既定値は10MBです。

- **autovacuum_list_limit** Int

既定値 50

日付の古いage(relfrozenxid)順に設定値程のテーブル情報を収集します。既定値は50件です。

ユーザーメトリック収集 (SQLMON)

SQLMON は、ユーザー定義の SQL 結果を定期的に収集し、メトリクスとして保存します。数値を返し、負荷が低いクエリに適しています。

- **sqlmon** Int

既定値: 1

- 1 : dbx と sqlmon を同時に実行
- 2 : sqlmon のみ実行

対応バージョン: 2.3.5 以上

設定

1. dbx エージェント配下に `sqlmon` ディレクトリを作成し、収集したいクエリを `<filename>.sql` ファイルとして作成します。

```
dbx/
```

```
└─ sqlmon/
  └─ active_sessions.sql
  └─ lock_count.sql
```

2. 収集するクエリを `<filename>.sql` ファイルとして保存します。

`.sql` ファイルの1行目に実行オプションを、2行目以降に SQL 文を記述します。

オプション形式

周期(分);動作時間;タグ名
SQL文

項目	説明	例
周期	実行周期（分単位）、最小値 1	1 → 1分ごとに実行
動作時間	HHMM-HHMM 形式、カンマ（,）で複数の時間帯を指定可能。0 の場合は常時実行	0900-1100 → 09:00～11:00のみ実行
タグ名	メトリクスに保存するタグ名。省略時はファイル名がタグ名として使用されます	whatap

例：1分周期、09:00～11:00動作、タグ名 whatap

```
1;0900-1100;whatap
SELECT count(*) FROM active_sessions
```

例：1分周期、常時動作、タグ名はファイル名を使用

```
1;0
SELECT count(*) FROM active_sessions
```

- ❗ 動作時間は日付をまたいで設定できます。（例：2300-0100 → 23:00～翌日01:00）
- タグ名を省略する場合、動作時間の後にセミコロン（;）を付けないでください。
- SQL 文の末尾にセミコロン（;）を付けないでください。

3. `whatap.conf` で SQLMON を有効化します。
4. エージェントを再起動します。
5. [🔗 サイトマップ > メトリクス照会](#) メニューで、`sqlmon` カテゴリを確認します。

```
# whatap.conf
sqlmon=1
```

スロークエリー収集

スロークエリーを収集するには、まずデータベース(DB)を設定した上で、ユーザの環境(オンプレミスまたはAWS RDS/Aurora)によってWhaTapの設定方法が異なります。

❗ スロークエリー機能はMySQLとPostgreSQLに限って対応します。

❗ データベース設定を確認する

WhaTap設定前、PostgreSQLデータベース(DB)に対してスロークエリー関連のパラメータを設定する必要があります。スロークエリーログ機能が活性化されていることを確認し、WhaTapの設定を進めます。

```
log_min_duration_statement=3000 # 3秒
```

オンプレミス環境

独自のサーバーで運用する場合、WhaTapのDBX/XOS設定方式を使用します。DBXエージェントのインストール以外に、DBサーバーにXOSを必ずインストールする必要があります。XOSエージェントの設定方法は[次の文書](#)を参考にしてください。

1. DBX`whatap.conf`ファイルに'slow_query_log=on'オプションを追加します。
 - `slow_query_log` **Boolean**
スロークエリーロギング機能を活性化(`on`)するか、非活性化(`off`)します。PostgreSQLの既定値は `off` です。

2. XOSがインストールされている場合は、ファイル名を自動認識します。ファイル名が自動的に認識されない場合は、XOS `whatap.conf` ファイルに `slow_query=` に設定します。

AWS RDS/Aurora環境

AWS RDS/Aurora環境はXOSエージェントをインストールできないので、クラウド設定により、スロー クエリを収集できます。

管理 > クラウド設定メニューで提供されるガイドに従って実行すると、ユーザーのAWS環境で自動的にモニタリングリソースを生成し、スロークエリーデータを収集します。別途の追加設定は必要ありません。詳しい内容は、[次の文書](#)を参照してください。

❗ PostgreSQLはDBXエージェント設定方式に対応していません。クラウド設定メニューを活用すると、CloudWatch指標とともにスロークエリーデータを収集できます。

AES256暗号化の適用

WhaTap DBXエージェントは、収集されたデータを暗号化してサーバーに送信します。データの重要度や設定によって変更できます。基本的にXOR演算とAESアルゴリズムによる暗号化を使用し、平文を128ビット単位に分割して暗号化、復号を行います。ユーザー設定により、256ビットまで拡張できます。

WhaTap DBXエージェントをインストールしたパスで、`whatap.conf` ファイルを次のオプションに追加します。

`whatap.conf`

```
1 cypher_level=256
```

設定を追加してから、エージェントを再起動してください。

エージェント名の設定

自動設定する

エージェント設定([whatap.conf](#))で名前を設定しないと、エージェント名はIP アドレスとポート番号が結合されて自動指定されます。例えば、IPアドレスが**10.11.12.13**であり、ポート番号が**3000**であれば、エージェント名は**DBX-12-13-3000**に設定されます。

エージェント名を決定する主な変数については、以下を参照してください。

設定	説明
<code>{type}</code>	DBX値を使います。
<code>{ip0}</code>	IPv4アドレスのうち、第1オクテットを使用します。(例：10.11.12.13のうち 10)
<code>{ip1}</code>	IPv4アドレスのうち、第2オクテットを使用します。(例：10. 11 .12.13のうち 11)
<code>{ip2}</code>	IPv4アドレスのうち、第3オクテットを使用します。(例：10.11. 12 .13のうち 12)
<code>{ip3}</code>	IPv4アドレスのうち、第4オクテットを使用します。(例：10.11.12. 13 のうち 13)
<code>{port}</code>	データベースサーバのポート

❗ RDSのようにIPアドレスが数字でなければDBX- `{ip0}` - `{hash}` - `{port}` 形式で名前が設定されます。ここで `{hash}` 値は、IPアドレスを4桁の16進数に変換した値です。

手動設定する

エージェント名を直接設定したい場合は、[whatap.conf](#)ファイルで `whatap.name` または `object_name` オプションに目的の名前を入力します。ユーザーが入力した名前がエージェント名として設定されます。

`whatap.conf`

```
whatap.name=agent-1234  
# or  
object_name=agent-1234
```

XOSエージェント設定

データベースサーバーのリソースをさらにモニタリングする場合は、データベースサーバーに別のXOSエージェントを実行してデータを収集できます。

- ❗ • x86アーキテクチャで動作するOS環境にのみ適用されます。
- XOSエージェントのインストールについての詳細は、[次の文書](#)を参照してください。

whatap.confファイルの設定

XOSエージェントを実行する前に、[whatap.conf](#)ファイルで、以下のオプションを設定します。

- **xos** **Int**

既定値 0

XOSエージェントを使用するかどうかを設定します。0 に設定するとXOSからのデータは収集されません。1 に設定するとすべてのXOSからのデータを収集します。IPアドレスを設定する場合、そのIPアドレスが一致するXOSからのデータのみ収集します。

- **xos_port** **String**

既定値 3002

XOSエージェントと通信するポートを設定します。XOSエージェントはUDPで通信します。ファイアウォールでUDPポートがブロックされている場合は、該当ポートを開放してください。

- **xos_cpu_iowait** **Int**

既定値 1

XOSエージェントから収集したCPU使用率にiowaitを追加できます。1 に設定するとiowaitが追加され、0 に設定するとiowaitは追加されません。

- **xos_mem_buffcache** **Int**

既定値 0

XOSエージェントから収集したメモリ使用率にbuffcacheを追加できます。1 に設定するとbuffcacheが追加され、

0 に設定するとbuffcacheは追加されません。

xos.confファイルの設定

xosディレクトリをデータベースサーバーにコピーし、xos.confファイルにエージェントサーバー情報を入力して追加オプションを設定してください。

- **dbx_ip** String

DBXエージェントをインストールしたサーバーのIPアドレスを入力します。

- **dbx_port** String

既定値 3002

DBXエージェントと通信するポートを設定します。

DBXエージェントは、UDPで通信します。ファイアウォールでUDPポートがブロックされている場合は、該当ポートを開放してください。

- **cpu_limit** Int

既定値 0

limitの設定値より大きい値を持つプロセス情報を収集します。1/100%単位で1.23%の場合、123として設定します。

- **mem_limit** Int

既定値 10240

limitの設定値より大きい値を持つプロセス情報を収集します。単位は「KB」で既定値である10240は、10MBになります。(1024 KB = 1 MB)

- **disk_usage_interval** Second

既定値 60

インスタンス別のディスク使用率と使用可能サイズ、全体サイズ情報を表示します。値が'0'の場合、ディスク使用率は収集しません。

- **port_check** String

接続状態をチェックするサーバーとポートを指定します。IP:Portの形式で設定し、コンマ(,)で区切って複数登録が

可能です。接続ができないサーバーとポートの一覧は、`db_port_check` カテゴリに保存されます。

xos.conf

```
port_check=192.168.122.113:3333,192.168.122.113:3334
```

接続ができなかったサーバーとポートの一覧を確認する場合、**分析>メトリクス照会メニュー**で `db_port_check` カテゴリを選択します。

-	Time	Oid	Tags	Fields	
			oname	error_count 	port_list
1	2025-01-17 18:39:41	1064156037	DBX-5-112-1521	2	192.168.122.113:3333,192.168.122.113:3334
2	2025-01-17 18:39:41	1064156037	DBX-5-112-1521	2	192.168.122.113:3333,192.168.122.113:3334
3	2025-01-17 18:39:51	1064156037	DBX-5-112-1521	2	192.168.122.113:3333,192.168.122.113:3334
4	2025-01-17 18:39:51	1064156037	DBX-5-112-1521	2	192.168.122.113:3333,192.168.122.113:3334

• process_check String

プロセスの実行状態をモニタリングできます。オプション値でプロセス名を入力します。コンマ(,)を区切り文字として使用し、複数登録することができます。

xos.conf

```
process_check=pmon,smon,tnslsnr,melong,haha
```

指定された文字列を含むプロセス名が検知されない場合、`db_process_check` カテゴリにその文字列の一覧を保存します。例えば、**melong**、**haha**の2つのプロセスがない場合、**分析>メトリクス照会メニュー**で `db_process_check` カテゴリを選択した後、確認できます。

-	Time	Oid	Tags	Fields	
			oname	error_count 	process_name
1	2025-01-17 18:35:06	1064156037	DBX-5-112-1521	2	melong,haha
2	2025-01-17 18:35:06	1064156037	DBX-5-112-1521	2	melong,haha
3	2025-01-17 18:35:12	1064156037	DBX-5-112-1521	2	melong,haha
4	2025-01-17 18:35:12	1064156037	DBX-5-112-1521	2	melong,haha

- ① `process_check` オプションを使用して、データベースで実行されているプロセスが終了した場合はアラート通知を送信できます。詳細は[次の文書](#)を参照してください。

- **process_check_interval** Second

既定値 `6`

`process_check` オプションに設定されたプロセスの実行状況を確認する間隔を設定します。(単位：秒)

- **iplist_interval** Second

既定値 `0` (off)

DBサーバーのIP一覧を収集する間隔を設定します。(単位：秒、3以上で設定)

- ① [whatap.conf](#)のファイルに `xos_dbip=true` オプションを設定すると `db_ip_list` のカテゴリに転送します。[分析 > メトリクス照会](#)メニューで `db_ip_list` カテゴリを選択した後、確認することができます。

- **slow_query** String

データベースのスロークエリを収集します。モニタリング対象DBにXOSがインストールされているものの、ファイル名を自動的に認識できない場合は、XOS [whatap.conf](#)ファイルに `slow_query=` と設定します。

- **file** String

[ログメニュー](#)([ログTail](#)、[ログトレンド](#)、[ログ検索](#)、[ログ設定](#))を使用するには、`file` オプションにモニタリングするログファイル名のフルパスを入力します。

`xos.conf`

```
file=/opt/postgresql/logs/postgresql.log
```

- ✓ モニタリングするログファイルを追加するには `file`、`file1`、`file2` といった方法でオプションを追加します。

`xos.conf`

```
file={log_file_path1}
```



```
file1={log_file_path2}
file2={log_file_path3}
file3={log_file_path4}
```

ユーザーメトリクス収集 (SHMON)

SHMONはユーザー定義のシェルスクリプトの実行結果を定期的に収集し、メトリクスとして保存します。

- **shmon** `Int`

既定値: `1`

対応バージョン: dbx 2.55.01以降 & xos 1.2.0f以降

設定

1. エージェントのインストールパスに `shmon` ディレクトリを作成します。
2. 収集するスクリプトを `.sh` ファイルとして保存します。
3. `xos.conf` でSHMONを有効にします。
4. エージェントを再起動します。
5. [🔗 サイトマップ](#) > [メトリクス照会](#) メニューで、`shmon` カテゴリを確認します。

```
# xos.conf
shmon=1
```



- `file` オプションは、DBXエージェントの1.6.7バージョン以降に対応しています。
- [ログメニュー](#)の詳細は[次の文書](#)を参照してください。

XOSエージェントの実行

データベースサーバーで次のバッチファイルを実行してください。

```
$ ./start.sh
```

```
# または
```

```
$ ./startd.sh #バックグラウンドで実行する場合
```

クラウドエージェント

この文書は、AWSとNCP環境でWhaTapクラウドエージェントを設定するためのガイドです。主要設定項目、フィールドの説明、権限構成、実行コマンドまで含めて、運用環境にすぐに適用できるように構成されています。

エージェント設定

AWS Aurora and RDSとNCP Cloud DB for PostgreSQLの設定方法です。

AWS Aurora and RDSの設定

config.yaml

```
input:
  csp: "aws" # クラウドサービスのプロバイダーを入力します。
  namespace: "rds"
  region: "ap-northeast-1" # インスタンスが位置するリージョンを設定します。
  instances: # 設定したインスタンスは常に収集します。クラスター内のインスタンスをここに指定すると、オートスケールに関係なく収集されます。
    - name: "mysql-rds" # インスタンス名
      slow_query: true # スロークエリーページを使用するには、trueに設定します。該当機能はMySQL及びPostgreSQLインスタンスでのみ対応されています。
  clusters:
    autoscale:
      enabled: false # オートスケールが有効な場合、スケーリングされたインスタンスを収集対象に追加または削除します。
      interval: 60 # 指定したクラスターのオートスケール確認間隔です。（単位: 秒）
    names:
      - "database-cluster-name"
  metrics: # 収集するメトリクスを入力します。
    - "CPUUtilization"
    - "FreeStorageSpace"
    - "FreeableMemory"
    - "ReadLatency"
    - "WriteLatency"
    - "ReadIOPS"
    - "WriteIOPS"
```

```

- "NetworkReceiveThroughput"
- "NetworkTransmitThroughput"
- "FreeLocalStorage"
logs:
  enabled: false # ログ収集の有無を設定します。true/falseで設定できます。
  groups: # 収集するAWSロググループを入力します。
    - "/aws/rds/cluster/database-cluster-name/error"
    - "/aws/lambda/MyLambda"

output: # 収集したメトリクスデータを受信するWhaTap情報を入力します。
license: "abcdefg-higjkgjk-zxcvnbmbmc"
host: "127.0.0.1"

```

フィールド	説明
csp	クラウドサービスの提供者として aws (Amazon Web Services)を意味
namespace	収集対象サービスのネームスペース - 固定値: Aurora and RDSの設定時、 rds を入力
region	AWSリージョンコード - 例 ap-northeast-1 (東京リージョン)
instances	常に収集するインスタンスの一覧で、オートスケーリング有無に関係なく、指定されたインスタンスは無条件に収集する
instances[].name	RDSインスタンス名 - 例 mysql-rds
instances[].slow_query	該当インスタンスのスロークエリーを収集するかどうか、MySQL及びPostgreSQLインスタンスでのみ対応 - true: 収集 - false: 非収集
clusters	RDSクラスター関連設定
clusters.autoscale.enabled	クラスターオートスケーリング収集有無 - true: 自動的にインスタンスを追加/削除して収集

フィールド	説明
	- <code>false</code> : <code>instances</code> に直接入力したインスタンスのみ収集対象に含める
<code>clusters.autoscale.interval</code>	オートスケーल有無を感知する周期(単位:秒)
<code>clusters.names</code>	オートスケーल有無をチェックするクラスター名の一覧
<code>metrics</code>	収集するメトリック項目のリスト - 例 <code>CPUUtilization</code> 、 <code>FreeStorageSpace</code> など - 参考 Amazon RDSに対するAmazon CloudWatch指標の公式文書へのショートカット
<code>output.license</code>	収集されたデータを転送するWhaTapのライセンスキー
<code>output.host</code>	収集データを転送するWhaTapサーバーのホストアドレス(IP)

クロスアカウントモニタリング（マルチAWSアカウント）



単一のクラウドエージェントで複数のAWSアカウントのRDS/DocumentDBをモニタリングできます。マルチアカウント（環境別分離、パートナーアカウントなど）を使用する場合、アカウントごとにエージェントをインストールする必要はなく、1つのエージェントからSTS AssumeRoleで他のアカウントのリソースを照会します。

項目	内容
対応対象	AWS RDS（Aurora、スタンドアロン）、DocumentDB
認証方式	EC2 Instance Profile + STS AssumeRole

設定方式の選択

クラスター設定には2つの方式があり、同時に使用することはできません。

- `names` : 単一アカウント専用（エージェントがインストールされたEC2のIAM Roleで自アカウントのみ照会）
- `account_configs` : クロスアカウント対応（STS AssumeRoleを通じて複数アカウントを照会）

**account_configs** 設定

config.yaml

```

input:
  csp: "aws"
  namespace: "rds"
  region: "ap-northeast-2"
  clusters:
    autoscale:
      enabled: true
      interval: 300
  account_configs:
    # ローカルアカウント (assume_role_arnを省略すると自アカウント)
    - clusters: ["local-aurora-cluster"]
      instances: ["standalone-mysql-1"]

    # 別のアカウント A
    - assume_role_arn: "arn:aws:iam::111111111111:role/DBXCRole"
      clusters: ["partner-cluster-1", "partner-cluster-2"]

    # 別のアカウント B
    - assume_role_arn: "arn:aws:iam::222222222222:role/QARole"
      clusters: ["qa-cluster"]
      instances: ["qa-standalone-pg"]
  metrics:
    - "CPUUtilization"
    - "FreeStorageSpace"
    - "FreeableMemory"
    - "ReadLatency"
    - "WriteLatency"

output:
  license: "your-license-key"
  host: "127.0.0.1"

```

表 | account_configs フィールド



フィールド	必須	説明
<code>assume_role_arn</code>	X	STS AssumeRole ARN。省略するとローカルアカウントを使用
<code>clusters</code>	O	Aurora/DocumentDBクラスター名。所属インスタンスを自動検出
<code>instances</code>	X	クラスターに属さないスタンドアロンRDSインスタンス名

ⓘ `clusters` と `instances` の違い

	<code>clusters</code>	<code>instances</code>
対象	Auroraクラスター、DocumentDBクラスター	クラスターなしで単独運用されるRDS
動作	クラスター名で所属インスタンスを自動ディスカバリー	インスタンス名を直接指定
オートスケール	インスタンスの追加/削除時に自動反映	該当なし
例	Aurora MySQLクラスター	RDS MySQL単一インスタンス

設定例1. 単一アカウント + スタンドアロンインスタンス

```
clusters:
  autoscale:
    enabled: true
    interval: 300
  account_configs:
    - clusters: ["prod-aurora"]
      instances: ["legacy-mysql-standalone"]
```



設定例2. 3つのアカウント（自アカウント + パートナー + QA）

```
clusters:
  autoscale:
    enabled: true
    interval: 300
  account_configs:
    - clusters: ["prod-cluster", "staging-cluster"]

    - assume_role_arn: "arn:aws:iam::111111111111:role/DBXCRole"
      clusters: ["partner-prod", "partner-dev"]

    - assume_role_arn: "arn:aws:iam::222222222222:role/QADBXCRole"
      clusters: ["qa-cluster"]
      instances: ["qa-standalone-pg"]
```

- ❗ names と account_configs は同時に使用できません。エージェント起動時にバリデーションエラーが発生します。
- ARN形式を厳密に検証します。arn:aws:iam::<12桁の数字>:role/<名前> — コロン2つ(::) を省略しないよう注意してください。
- 各 account_config には最低1つのclusterが必要です。
- autoscale.enabled: true でないとクラスターディスカバリーが動作しません。

NCP Cloud DB for PostgreSQL設定

config.yaml

```
input:
  csp: "ncp" # クラウドサービスプロバイダーを入力します。
  namespace: "vpc_postgresql"
  region: "KR" # KR | SGN | JPN | USWN | DEN
  accessKey: "YOUR_ACCESS_KEY"
  secretKey: "YOUR_SECRET_KEY"
```

```

instances: # Cloud DB for PostgreSQL (VPC) リストでモニタリング対象DBサーバー名を明示します。
- name: "instance-name"
  domain: "pg-xxxx.vpc-pub-cdb-kr.ntruss.com" # 接続可能なprivateまたはpublicドメインを入力します。
metrics: # 収集するメトリックを入力します。
- "cpu_user"
- "mem_used"
- "disk_usage"
- "nic_total"
- "cpu_load_1"
- "tup_inserted"

output: # 収集したメトリックデータを受信するWhaTap情報を入力します。
license: "4544ee0few5-a880410bb205-2987709519af"
host: "127.0.0.1"

```

フィールドパス	説明
csp	クラウドサービスのプロバイダーとしてNaver Cloud Platform(ncp)使用を表す
namespace	モニタリング対象の区切り記号またはグループ名で使用する - 例 vpc_postgresql
region	クラウドリージョン(region)コード - 対応値: KR、SGN、JPN、USWN、DEN
accessKey	Naver CloudのAPI呼び出しのためのアクセスキー - 参考 Access Key / Secret Key発行方法
secretKey	Naver CloudのAPI呼び出しのためのシークレットキー
instances	モニタリング対象であるPostgreSQLインスタンスリスト - 各インスタンスは name と domain フィールドを含む
instances[].name	Cloud DB for PostgreSQL (VPC)インスタンス名 - NAVER CLOUD PLATFORMコンソールのDB Server > DB Server名項目で確認可能 - 例 whatap-001-70ld
instances[].domain	当該インスタンスのアクセス可能なドメインアドレス

フィールドパス	説明
	- NAVER CLOUD PLATFORMコンソールのDB Server > PrivateまたはPublicドメイン項目で確認可能 - DBXエージェントが実際に接続中の PrivateまたはPublicドメインアドレスを入力する必要がある - 例 pg-33rr40.vpc-pub-cdb-kr.ntruss.com
metrics	収集するメトリック項目のリスト - 参考 Cloud DB for PostgreSQL (VPC)指標公式文書
output	収集したメトリックデータを受信するWhaTap情報
output.license	収集されたメトリックデータを転送するWhaTapのライセンスキー
output.host	収集データを転送するWhaTapサーバーのホストアドレス(IP)

Access Key / Secret Key の発行方法

1. [NAVER CLOUD PLATFORM コンソール](#)にログインします。
2. 上部メニューから My Page > アカウント管理 に移動します。
3. メニューで 認証キー管理 をクリックします。
4. [キー生成] ボタンをクリックして、新しい Access Key と Secret Key を生成します。

権限設定

エージェントが正常に動作するには、サービスごとに必要な権限を設定する必要があります。

AWS Aurora and RDS権限

クラウドエージェントは、次の機能を実行するためにAWS APIを呼び出します。

- AWS Aurora and RDSインスタンスとクラスター情報を照会して監視対象を確認
- CloudWatchメトリック収集

IAMインラインポリシー

JSONポリシーをIAMユーザーまたは役割にインラインポリシーで追加します。EC2でエージェントを実行する場合、当該EC2インスタンスに接続されたIAMの役割に当該ポリシーを付与します。

- このポリシーはAuroraとRDSサービスの両方に適用されます。
- Resource項目は、すべてのAuroraとRDSリソースを対象に指定されており、セキュリティポリシーによって必要な場合、リソースARNを制限できます。
- エージェントは照会(Read-only) APIのみ使用します。データベースに影響を与える権限は含まれません。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RDSPermissions",
      "Effect": "Allow",
      "Action": [
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "rds:DescribeDBClusterParameters",
        "rds:DescribeOptionGroups"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CloudWatchMetricsPermissions",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:GetMetricData",
        "cloudwatch:GetMetricStatistics",
        "cloudwatch:ListMetrics"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CloudWatchLogsPermissions",
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogGroups",
```

```

    "logs:DescribeLogStreams",
    "logs:FilterLogEvents",
    "logs:GetLogEvents"
  ],
  "Resource": [
    "arn:aws:logs:*:*:log-group:/aws/rds/*"
  ]
}
]
}

```

クロスアカウントIAM権限

- ① account_configs を使用してクロスアカウントモニタリングを設定するには、追加のIAM権限が必要です。

エージェントEC2 Instance Profile

エージェントが実行されるEC2に、対象アカウントのRoleをAssumeできる権限を付与します。

```

{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": [
        "arn:aws:iam::111111111111:role/DBXCRole",
        "arn:aws:iam::222222222222:role/QARole"
      ]
    }
  ]
}

```

対象アカウントのCross-Account Role

対象アカウントでエージェントアカウントからのアクセスを許可するRoleを作成します。

- **Trust Relationship:** エージェントアカウントを許可



```
{
  "Principal": {
    "AWS": "arn:aws:iam::<エージェントアカウントID>:role/DBXC-EC2-Instance-Role"
  },
  "Action": "sts:AssumeRole"
}
```

- **Permission Policy: RDS/CloudWatch照会権限**

```
{
  "Effect": "Allow",
  "Action": [
    "rds:DescribeDBInstances",
    "rds:DescribeDBClusters",
    "cloudwatch:GetMetricData",
    "logs:FilterLogEvents"
  ],
  "Resource": "*"
}
```

NCP Cloud DB for PostgreSQL権限

クラウドエージェントは、NCP Cloud DB for PostgreSQLインスタンスのメトリックと設定情報を収集するために、NCP Cloud Insight APIを呼び出します。 そのためにはAccess KeyとSecret Keyが必要です。

Access Key / Secret Key発行方法

1. [NAVER CLOUD PLATFORMコンソール](#)にログインします。
2. 上段メニューでMy Page> アカウント管理に移動します。
3. メニューで認証キー管理を選択します。
4. ****[キー生成]****ボタンをクリックして新しいAccess KeyとSecret Keyを生成します。

料金案内

- ❗ クラウドモニタリングエージェントは無料ですが、クラウドサービスの使用によっては別途料金が発生することがありますので、確認が必要です。

AWS料金

クラウドモニタリングエージェントは別途のSaaS料金なしで提供され、エージェントのインストールだけでモニタリング機能を使用できます。ただ、AWS CloudWatchなどクラウドサービスのAPIを通じてメトリックを収集するため、CloudWatch使用量によって別途料金がかかる場合があります。

本エージェントは、基本的に1分周期でGetMetricData APIを呼び出ししてメトリックを収集し、呼び出し頻度と収集対象によって料金が増加することがあるので注意が必要です。

詳しい料金ポリシーは[AWS公式文書のAmazon CloudWatch料金](#)ページを参照してください。

NCP料金

クラウドモニタリングエージェントは別途のSaaS料金なしで提供され、エージェントのインストールだけでモニタリング機能を使用できます。クラウドモニタリングエージェントは、5分周期でCloud Insight APIを呼び出し、1分単位のメトリックデータを収集します。

NAVERクラウドプラットフォーム(NCP)のCloud Insightサービスは現在無料で提供されていますが、今後のポリシーによっては有料化される場合があります。Cloud Insightサービスが有料化される場合、API呼び出し回数によって料金がかかる場合がありますので、使用量にご注意ください。

詳しい料金ポリシーは[NAVERクラウドプラットフォームの公式料金案内](#)ページを参照してください。

- ❗
- names と account_configs は同時に使用できません。エージェント起動時にバリデーションエラーが発生します。
 - ARN形式を厳密に検証します。arn:aws:iam::<12桁の数字>:role/<名前> — コロン2つ(::)を省略しないよう注意してください。



- 各 `account_config` には最低1つのclusterが必要です。
- `autoscale.enabled: true` でないとクラスターディスカバリーが動作しません。

インストールと使用

DBX for Cloud、DBXCは2つの主要コンポーネントで構成されています。

- **dbxc-agent**: クラウド指標を監視するバックグラウンドサービス（エージェント）
- **dbxc-ctl**: エージェントを制御するCLIツール（開始、停止、状態確認、ログ参照）

設定ファイル構造

DBXCはYAML形式の設定ファイルを通じて実行構成を定義します。

サポート形式

DBXCは単一/マルチプロジェクトエージェントをサポートします。

- YAML形式 (.yaml, .yml)
- 単一ファイル: 1つのエージェントを実行
- ディレクトリ: 複数のエージェントを実行

設定ファイル自動探索順序

1. `c` オプションで指定されたファイル/ディレクトリ
2. `config.yaml`, `config.yml` (デフォルトファイル)
3. カレントディレクトリ内のすべての `.yaml`, `.yml` ファイル

設定ファイル例

ディレクトリ tree 例

```
> tree
.
├── configs
```

```

|   |—— project-1.yml
|   |—— project-2.yml
|—— dbxc-agent
|—— dbxc-ctl
|—— dbxc.log
|—— dbxc.pid

> ls -al
total 68456
drwxr-xr-x 8 kyw staff 256 7 18 20:23 .
drwxr-xr-x 9 kyw staff 288 7 18 20:10 ..
drwxr-xr-x 4 kyw staff 128 7 18 20:23 configs
-rwxr-xr-x 1 kyw staff 20174450 7 18 20:09 dbxc-agent
-rwxr-xr-x 1 kyw staff 8565570 7 18 20:09 dbxc-ctl
-rw----- 1 kyw staff 5574908 7 18 20:30 dbxc.log
-rw-r--r-- 1 kyw staff 5 7 18 20:23 dbxc.pid

```

yml 例

```

input:
  csp: "aws" # クラウドサービスのプロバイダーを入力します。
  namespace: "rds"
  region: "ap-northeast-1" # インスタンスが位置するリージョンを設定します。
  instances: # 指定したインスタンスは常に収集されます。クラスター内のインスタンスをここに指定すると、オートスケールに関
    係なく収集されます。
    - name: "mysql-rds" # インスタンス名
      slow_query: true # スロークエリページを利用する場合は true に設定してください。
  clusters:
    autoscale:
      enabled: false # オートスケールが有効な場合、スケーリングされたインスタンスを収集対象に追加または削除します。
      interval: 60 # 指定したクラスターのオートスケール確認間隔です。（単位: 秒）
    names:
      - "database-cluster-name"
  metrics: # 収集するメトリクスを入力します。
    - "CPUUtilization"
    - "FreeStorageSpace"
    - "FreeableMemory"
    - "ReadLatency"
    - "WriteLatency"
    - "ReadIOPS"

```

```

- "WriteIOPS"
- "NetworkReceiveThroughput"
- "NetworkTransmitThroughput"
- "FreeLocalStorage"
logs:
  enabled: false # ログ収集の有無を設定します。true/falseで設定できます。
  groups: # 収集するAWSロググループを入力します。
    - "/aws/rds/cluster/database-cluster-name/error"
    - "/aws/lambda/MyLambda"

output: # 収集したメトリクスを送信する WhaTap 情報を入力します。
license: "abcdefg-higjkgjk-zxcvnbmbmc"
host: "127.0.0.1"

```

エージェント実行および制御コマンド

CLIはエージェントの開始から状態確認、ログ参照まで様々なコマンドを提供します。

開始/終了

```

# エージェント開始
dbxc-ctl start -c project.yaml # 特定の設定ファイル
dbxc-ctl start --config configs # 複数プロジェクト（ディレクトリ）
dbxc-ctl start                # デフォルト設定ファイルを自動探索

# エージェント停止
dbxc-ctl stop

# エージェント再起動
dbxc-ctl restart -c project.yaml

```

状態確認

```

# 簡単な状態確認
dbxc-ctl status
dbxc-ctl describe

```

ログ参照

```
# リアルタイムログ参照
dbxc-ctl logs -f

# 特定プロジェクトのログのみ参照
dbxc-ctl logs -c project.yaml

# ログレベルでフィルタリング
dbxc-ctl logs --level error

# 直近100件のログ参照
dbxc-ctl logs --tail 100
```

エージェント起動失敗時

```
# ログ確認
dbxc-ctl logs --level error
```

エージェント状態確認

```
# 全体状態確認
dbxc-ctl describe

# 特定エージェントのログ確認
dbxc-ctl logs -c specific-config.yaml -f
```

✔ 一般的なトラブルシューティング

- **ポート競合:** デフォルトポート53200が使用中か確認
- **設定ファイルエラー:** YAML文法を確認
- **権限問題:** ファイル読み取り権限を確認
- **PIDファイルロック:** 強制終了後にPIDファイルを削除

詳細技術情報

DBXCの詳細なコマンド構成および内部システム動作情報です。

コマンド構造

dbxc-ctlは以下のコマンドをサポートします。

- **start**: DBXCエージェント開始
- **stop**: DBXCエージェント停止
- **restart**: DBXCエージェント再起動
- **logs**: ログ参照
- **status または describe**: エージェント状態確認
- **version**: バージョン情報確認
- **help**: ヘルプ表示

HTTP API エンドポイント

エージェントはHTTPエンドポイントを提供します。

- `http://localhost:53200/health` : ヘルスチェック
- `http://localhost:53200/status` : 状態情報
- `http://localhost:53200/logs` : ログ参照

ログシステム

- ファイル名: dbxc.log
- 最大サイズ: 1ファイルあたり10MB
- 最大バックアップファイル数: 7個 (dbxc.log.1, dbxc.log.2, ... dbxc.log.7)
- 保管期間: 14日
- 圧縮: 古いログファイルを自動圧縮 (.gz形式)

- タイムゾーン: ローカル時間

ログファイルが10MBに達すると自動的にローテーションされ、最大7個のバックアップファイルが保持されます。14日以上経過したログは自動的に削除され、ディスク容量節約のため古いログは圧縮されます。

ログ収集設定

データベースクラウドエージェントは、AWS CloudWatch Logs API を使用して RDS および Aurora インスタンスを含む様々なログを定期的に収集できます。

本ドキュメントでは、API がコストにどのような影響を与えるかを理解し、それに合わせて適切な収集設定を行う方法を説明します。

CloudWatch Logs API の利用とコストへの影響

DBXC Agent は CloudWatch Logs の **FilterLogEvents API** を利用して、すでに保存されたログを取得します。

このとき発生するコストは、大きく分けて2つの要素によって決まります。

1つ目は **取得データ量** です。API を呼び出すと、指定した時間範囲と制限値（`regular_fetch_limit`）に基づいて CloudWatch Logs からログを読み込みます。読み込むデータ量が多いほど、AWS 内部処理およびデータ転送に伴うコストが増加します。

2つ目は **API 呼び出し回数** です。interval が短かったり収集対象のロググループが多い場合、定期的に多数の API 呼び出しが発生します。呼び出し回数自体にも課金が適用されるため、短い間隔で頻繁に呼び出すと、データ量が少なくても呼び出しコストが累積される可能性があります。

結果として、FilterLogEvents を用いたログ収集では **取得データ量** がコストにより大きな影響を与えますが、**短い間隔と多くの対象グループ** を組み合わせて利用すると API 呼び出しコストも無視できなくなります。

そのため、効率的な運用のためには `regular_fetch_limit` で不要に大量のログを一度に取得しないようにし、環境に合わせて interval を調整して呼び出し回数を管理することが重要です。

❗ 公式参考ドキュメント

- AWS API Reference: [FilterLogEvents](#)
- AWS Pricing: [CloudWatch Logs 料金](#)

ログ収集の動作方式

DBXC Agent は FilterLogEvents API を定期的に呼び出し、指定されたロググループから新しいログを取得します。

各呼び出し時点では、最後の収集時刻から現在時刻までのログのみを取得し、各ロググループごとに収集状態を個別に管理します。

最後の収集時刻はローカル状態ファイルに保存されるため、エージェントを再起動しても前回の位置から継続して収集できます。

運用環境別推奨設定

すべての設定では `logs.groups` 項目に実際に収集するロググループ一覧を指定する必要があり、ログ収集には `logs:FilterLogEvents` と `logs:DescribeLogGroups` の権限が必要です。

一般プロダクション環境（デフォルト）

```
logs:
  groups:
    - "/aws/rds/instance/prod-mysql/error"
    - "/aws/rds/instance/prod-mysql/general"
  interval: 60
  regular_fetch_limit: 1000
```

これはエージェントの既定値であり、設定ファイルに明示しなくても同じように適用されます。この構成はサービス開始時に不要に大量のログを一度に取得する状況を防ぎ、その後は各サイクルで最大1,000件のログを安定的に収集します。リアルタイム性、安定性、コスト効率のバランスが良く、ほとんどの運用環境に適しています。

リアルタイム重視環境

```
logs:
  groups:
    - "/aws/rds/cluster/prod-aurora/audit"
    - "/aws/rds/cluster/prod-aurora/error"
  interval: 30
  regular_fetch_limit: 200
```

30秒ごとにログを収集して遅延を最小化し、一度に取得する量を減らすことでメモリ使用量を抑えます。ただし呼び出し回数が増えるためAPI呼び出しコストが上昇する可能性があり、リアルタイム性が重要なサービスにのみ適用するのが望ましいです。

コスト最適化環境

```
logs:
  groups:
    - "/aws/rds/instance/mysql/error"
  interval: 300
  regular_fetch_limit: 500
```

5分ごとにログを収集することで API 呼び出し回数を大幅に削減し、コストを節約できます。ただし最大5分のログ反映遅延が発生するため、リアルタイムモニタリングが必須の環境には適していません。

設定チューニング方法

API コストが予想以上に高い場合は interval を延ばすか regular_fetch_limit の値を下げ、ログ欠落の懸念がある場合は interval を短くします。メモリ使用量が高い場合は regular_fetch_limit の値を減らすか、収集対象のロググループ数を減らすと良いでしょう。

IAM 権限設定

```
{
  "Effect": "Allow",
  "Action": [
    "logs:FilterLogEvents",
    "logs:DescribeLogGroups"
  ],
  "Resource": "arn:aws:logs:*:*:log-group:/aws/rds/*"
}
```

サーバーレスエージェント設定

ホーム画面 > プロジェクト選択 > 管理 > サーバーレスエージェント設定

❗ サーバーレスエージェントは、専用サーバーやプログラムのインストールを必要とせず、サーバーレス実行環境（Lambda、Functions など）を利用してクラウドメトリクスのみを収集するための軽量な収集ツールです。

AWS CloudFormation や Azure Resource Manager Template を通じて簡単にデプロイでき、管理負担なく迅速に設定できます。

このエージェントは [クラウドエージェント設定](#) と同じ機能を提供しますが、単一インスタンス環境でメトリクスのみを簡単に収集したい場合に適しています。

ただし、**ログ収集はサポートしていません**。複数インスタンスを運用する場合や CloudWatch Logs などのログ収集が必要な場合は、クラウドエージェント設定の使用を推奨します。

クラウドサービスが提供するモニタリングメトリクスを、データベースプロジェクトのダッシュボードで追加で収集し、モニタリングできるように設定する機能を提供します。サーバーレスエージェント設定メニューを通じて、ユーザーはクラウドデータベースサーバーのリソース状況をリアルタイムでモニタリングして、運用効率を最大化できます。

クラウド設定により、CloudWatch指標とスロークエリーデータを収集できます。スロークエリー収集はMySQL、PostgreSQLに限って対応します。

- 設定過程が直観的かつ簡単なので、非開発者でも素早く設定できます。
- 設定を完了した後、視覚的なフィードバックを通じて設定エラーをすぐに確認して修正できます。
- CSPコンソールで直接役割を生成して設定するため、セキュリティグループやポリシーを細かく管理できます。

❗ 設定後の使用量によって、WhaTap料金とは別にクラウドで追加費用が発生する場合があります。インストール前に、使用中のクラウドサービスの価格ポリシーを確認します。

AWS CloudWatch設定

AWSクラウド環境で実行中のデータベースの状態とリソース使用状況をモニタリングできます。

メトリクスを追加するサービスを選択する

1. メトリクスを追加するサービスを選択します。

 AWS CloudWatch AWS CloudFormation テンプレートをインストールして Amazon CloudWatch メトリクスを収集します。	 Azure Monitor Azure Active Directory に App Registration を追加して、Azure Monitor メトリクスを収集します。
--	--

最初のセクションで**AWS CloudWatch**を選択します。 **AWS CloudFormation**テンプレートをインストールして Amazon CloudWatchメトリクスを収集します。

パラメータを入力する

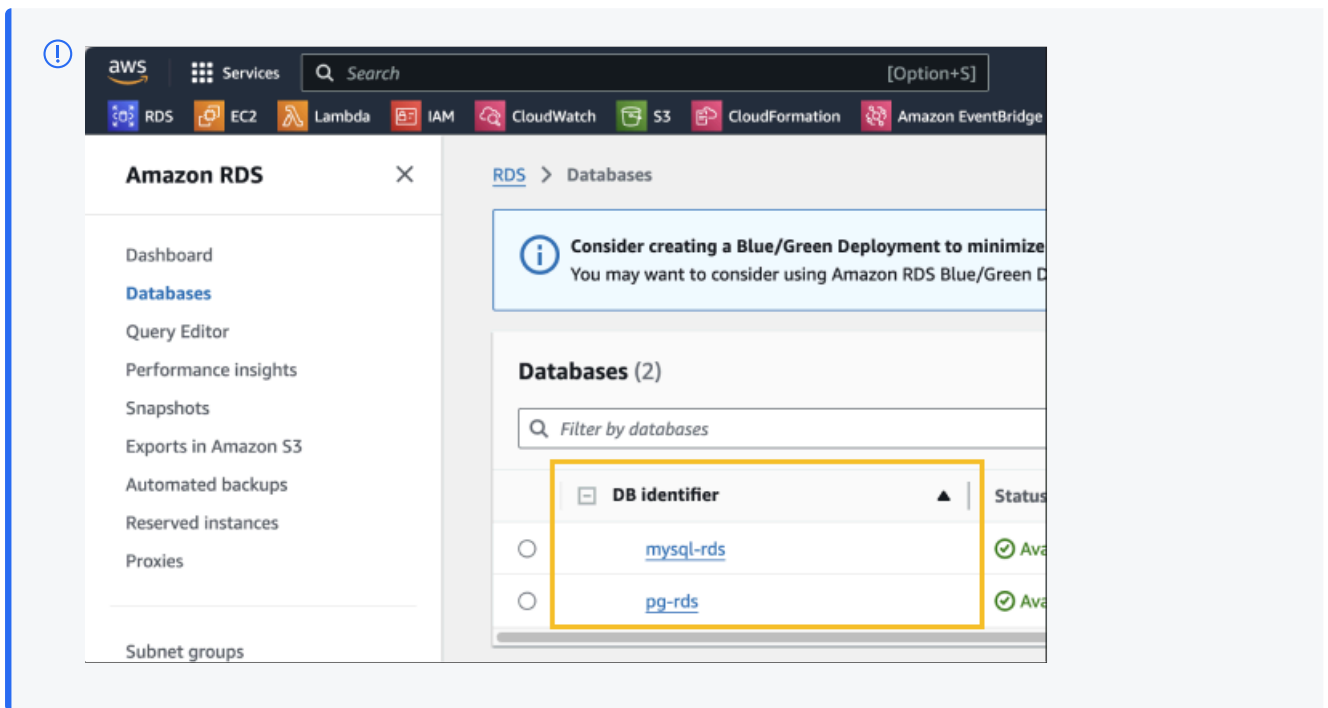
AWS Cloud Formation設定を進める前に、必須項目を入力します。

2. パラメータを入力します。

DB identifier	DB identifier
Instance	Database instance ▼
Interval	60 ▼
Region ⓘ	Tokyo (ap-northeast-1) ▼

- **DB Identifier:** モニタリングするAWSデータベースインスタンスの識別子を入力します。

① Amazon RDSサービスで生成したDatabaseインスタンスの**DB identifier**値を入力します。



- **Instance:** AWSクラウド環境で実行中のデータベースから収集したメトリクスをモニタリングするインスタンス(エージェント)を選択します。
- **Interval:** メトリクス収集間隔を選択します。(60秒/300秒)
- **Region:** モニタリングしたいAWSデータベースに適用されたリージョンを選択します。別のリージョンを入力すると、指標の照会ができません。

① 入力フィールドに文字列を入力すると、一致する結果をすばやく確認して選択できます。

Region ⓘ

Tokyo (ap-northeast-1)

AWS CloudFormationを設定する

WhaTapで提供するAWS CloudFormationテンプレートをインストールします。このテンプレートを使用して、CloudWatchメトリクスをWhaTapで収集できます。**AWS CloudFormation 設定ボタン**を選択すると、インストール

を開始し、設定プロセスを実行できます。

3. AWS CloudFormationを設定します。

WhaTap Monitoringが提供するAWS CloudFormationテンプレートをインストールしてクラウドメトリクスを収集します。以下のガイドリンクを参照して、メトリクス収集用のAWS CloudFormationのインストールと設定を進めます。

[AWS CloudFormation 設定 >](#)

1. **AWS CloudFormation 設定**ボタンを選択すると、**AWS CloudFormation**設定画面に移動します。
2. **Stack name**と**Parameters**セクションで自動的に満たされた項目の値を確認します。
3. 画面の一番下に位置した**Create stack**ボタンを選択します。

- ❗ • AWS CloudFormationについての詳細は[次のリンク](#)を参照してください。
- Amazon RDSに対するAmazon CloudWatchメトリクス情報は、[次のリンク](#)を参照してください。

必要権限を設定する

CloudFormationスタック(stack)を生成するには、次のAWS権限を設定する必要があります。

AWS権限の案内(JSON)



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack",
        "cloudformation>DeleteStack",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackResources",
        "cloudformation:GetTemplateSummary",
        "cloudformation:DescribeStackEvents",
        "cloudformation:ListStacks",
```



```
"cloudformation:ListStackResources"
],
"Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "lambda:CreateFunction",
    "lambda:UpdateFunctionCode",
    "lambda:UpdateFunctionConfiguration",
    "lambda:InvokeFunction",
    "lambda:DeleteFunction",
    "lambda:GetFunction",
    "lambda:GetFunctionConfiguration",
    "lambda:ListFunctions"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogGroup",
    "logs:CreateLogStream",
    "logs:PutLogEvents",
    "logs:DescribeLogGroups",
    "logs:DescribeLogStreams",
    "logs:GetLogEvents",
    "logs:FilterLogEvents"
  ],
  "Resource": "*"
}
{
  "Effect": "Allow",
  "Action": [
    "events:PutRule",
    "events:PutTargets",
    "events:RemoveTargets",
    "events:DeleteRule",
    "events:DescribeRule",
```




```

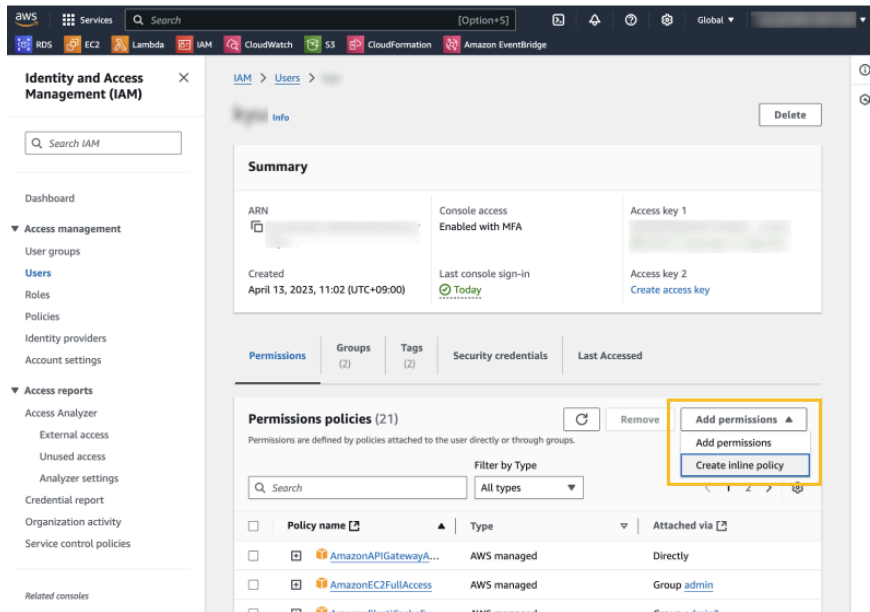
    "events:EnableRule",
    "events:DisableRule"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "iam:CreateRole",
    "iam:AttachRolePolicy",
    "iam:PutRolePolicy",
    "iam:GetRole",
    "iam>DeleteRole",
    "iam:PassRole",
    "iam:ListRoles"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "cloudwatch:GetMetricData"
  ],
  "Resource": "*"
},
{
  "Effect": "Allow",
  "Action": [
    "s3:GetObject"
  ],
  "Resource": "arn:aws:s3:::repo.whatap.io/agent/db/WhaTapRDSMonitoring.zip"
},
{
  "Effect": "Allow",
  "Action": [
    "scheduler:GetSchedule",
    "scheduler:ListSchedules",
    "scheduler:CreateSchedule"
  ],

```

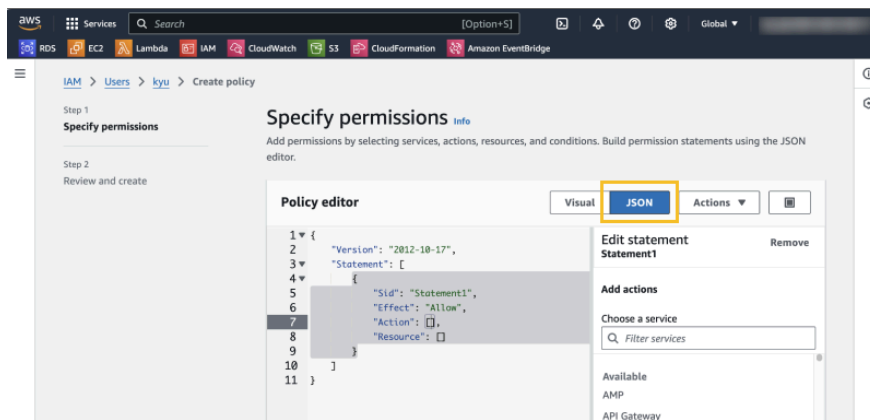


```
"Resource": "*"
}
]
}
```

1. AWS権限案内のJSON内容をコピーします。
2. IAMでCloudFormationスタック(stack)を生成するUserを選択します。
3. Create inline policyを選択します。



4. JSON選択後、コピーした権限を貼り付けてpolicyを登録します。

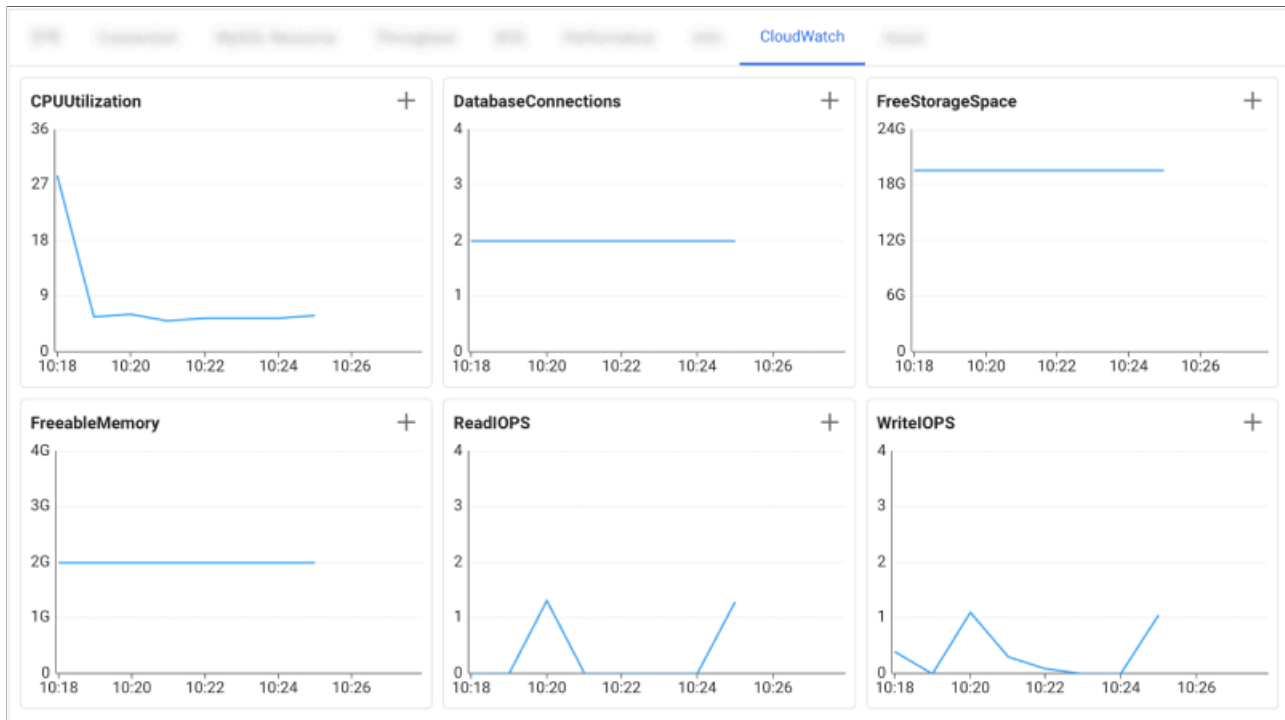


収集メトリクスを確認する

クラウドデータベース接続を完了すると、画面に追加されたクラウドサービスセクションが生成されます。クラウドデータベースに接続されたインスタンス(エージェント) および収集されるメトリクスの一覧を確認できます。

追加されたクラウドサービス		
過去5分以内にクラウド指標が収集された場合に表示されます。		
インストールされたエ...	クラウドサービス	収集指標
...	aws CloudWatch	CPUUtilization, DatabaseConnections, FreeStorageSpace, FreeableMemory, ReadIOPS, WriteIOPS
...	aws CloudWatch	CPUUtilization, DatabaseConnections, FreeStorageSpace, FreeableMemory, ReadIOPS, WriteIOPS

AWS CloudWatchから収集したメトリクスはCloudWatchカテゴリに含まれます。



次のメニューパスで確認できます。

- **ダッシュボード>インスタンスモニタリング:** ウィジェットの⚙️ボタンを選択してメトリクスを選択できます。
- **ダッシュボード>マルチインスタンス:** ユーザー定義ウィジェットを追加してメトリクスを選択できます。

- ❗
- **インスタンスモニタリングメニュー**でウィジェットのメトリクスを変更する方法は、[次の文書](#)を参照してください。
 - **マルチインスタンスメニュー**でユーザー定義ウィジェットを追加する方法は、[次の文書](#)を参照してください。
 - **分析>メトリクス照会メニュー**でカテゴリ項目をdb_aws_rdsに選択します。関連したタグとフィールド情報を確認できます。

Azure Monitor設定

Azureクラウド環境で実行中のデータベースの状態と資源使用状況をモニタリングできます。

メトリクスを追加するサービスを選択する

1. メトリクスを追加するサービスを選択します。

 AWS CloudWatch AWS CloudFormation テンプレートをインストールして Amazon CloudWatch メトリクスを収集します。	 Azure Monitor  Azure Active Directory に App Registration を追加して、Azure Monitor メトリクスを収集します。
--	--

最初のセクションで**Azure Monitor**を選択します。 **Azure Resource Manager**テンプレートをインストールして、Azure Monitorが提供するメトリクスを収集します。

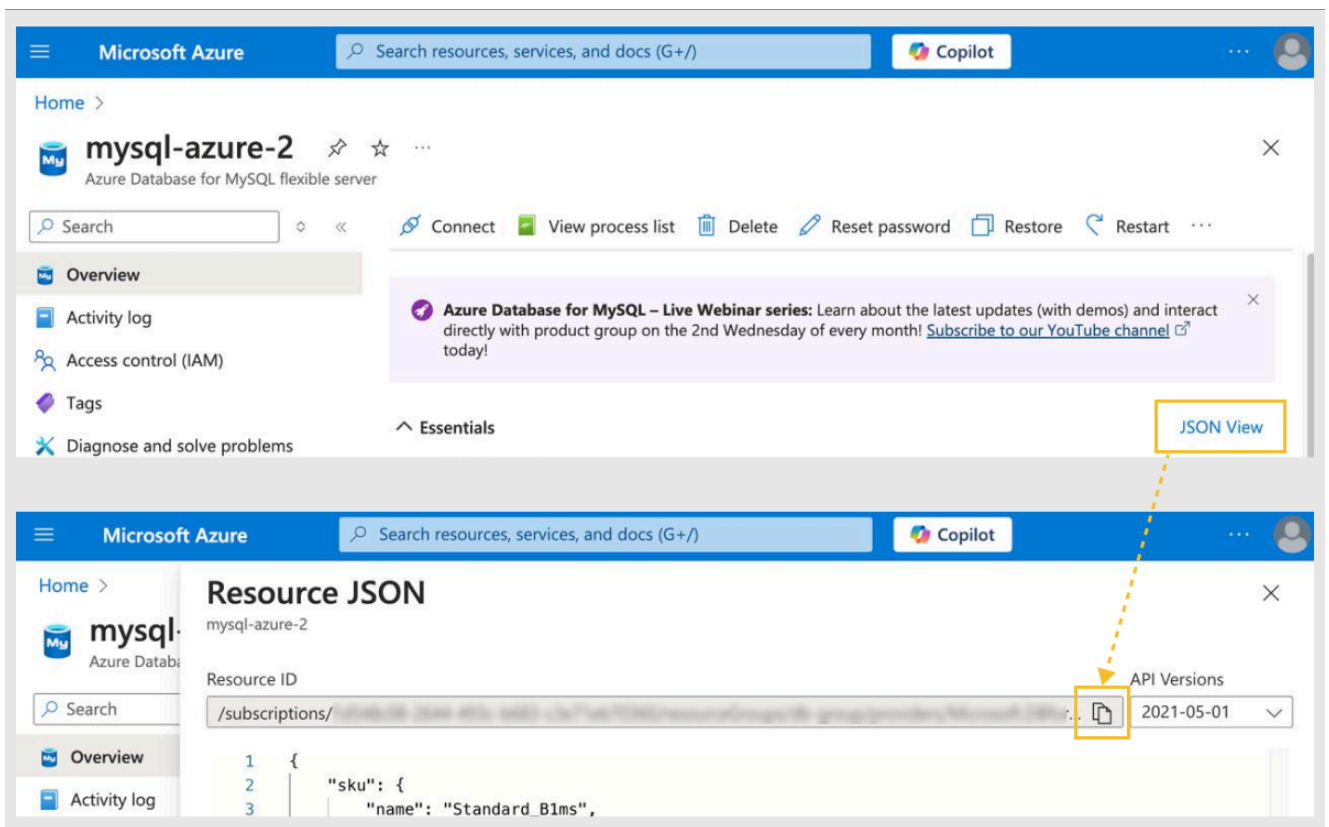
パラメータを入力する

Azure Resource Manager設定を進める前に、必須項目を入力します。

2. パラメータを入力します。

Instance	Azure Monitor
Resource ID ⓘ	Resource ID resourceIDは、 監視対象のデータベースコンソールで確認して入力する 必要があります。
JSONの生成	

- **Instance:** Azureクラウド環境で実行中のデータベースから収集したメトリクスをモニタリングするインスタンス(エージェント)を選択します。
- **Resource ID:** モニタリング対象データベースのリソースIDを入力します。 この情報は、モニタリング対象データベースのコンソールで確認できます。 次の画像を参照してください。



パラメータを入力した後、**JSONの生成**ボタンを選択します。Azure Resource Manager設定に必要なJSONを生成します。生成されたJSONコンテンツをコピーします。

Azure Resource Managerを設定する

WhaTapで提供するAzure Resource Managerテンプレートをインストールします。このテンプレートを通じて、Azure MonitorメトリクスをWhaTapサーバーに収集するFunction Appをインストールできます。**Azure Resource Manager 設定**ボタンを選択すると、インストールを開始し、設定プロセスを実行できます。

3. Azure Resource Managerを設定します。

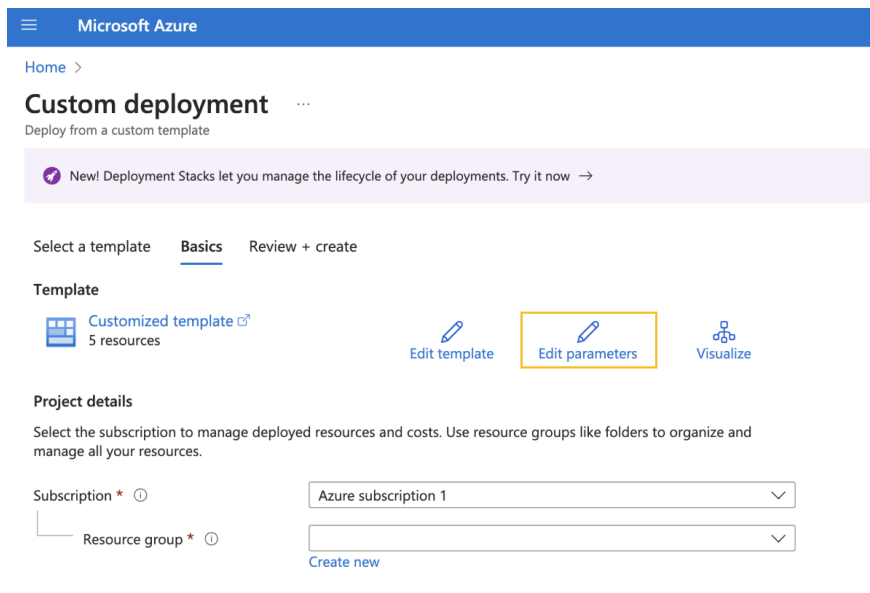
WhaTapが提供するAzure Resource Manager (ARM) テンプレートを使用して、Azure MonitorメトリクスをWhaTapサーバーとして収集するFunction Appをインストールできます。以下のガイドを参照して、ARMテンプレートを使用してメトリクスを収集するための設定を行い、Functions Appをインストールしてください。

[Azure Resource Manager 設定 >](#)

1. **Azure Resource Manager 設定**ボタンを選択すると、Microsoft Azureのテンプレート生成画面に移動しま

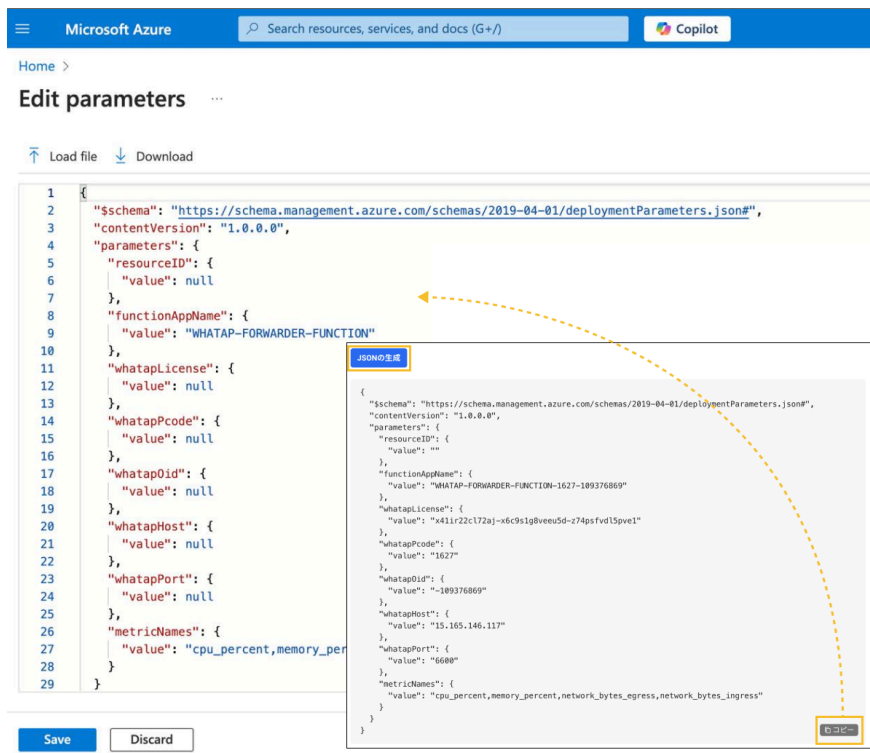
す。

2. Edit Parametersを選択します。



The screenshot shows the Microsoft Azure portal interface for a custom deployment. The top navigation bar is blue with the Microsoft Azure logo. Below it, the breadcrumb 'Home >' is visible. The main heading is 'Custom deployment' with a subtitle 'Deploy from a custom template'. A purple banner below the heading says 'New! Deployment Stacks let you manage the lifecycle of your deployments. Try it now →'. The 'Select a template' section has three tabs: 'Select a template', 'Basics' (which is active), and 'Review + create'. Under the 'Basics' tab, there is a 'Template' section with a 'Customized template' icon and '5 resources'. Below this are three buttons: 'Edit template', 'Edit parameters' (which is highlighted with a yellow box), and 'Visualize'. The 'Project details' section follows, with instructions to 'Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.' It contains two dropdown menus: 'Subscription *' (set to 'Azure subscription 1') and 'Resource group *' (empty). A 'Create new' link is located below the 'Resource group' dropdown.

3. 先のJSONの生成ボタンで生成したJSON形式のパラメータを貼り付けます。



4. 画面下段のSaveボタンを選択すると、パラメータが自動的に入力されます。
5. **Resource group**を選択します。**Resource group**は、モニタリング対象データベースを持つグループを選択する必要があります。

Azure

6. **Review + Create**ボタンを選択します。
7. 次の画面で**Create**ボタンを選択します。

配布が行われ、すべてのプロセスが完了します。

- ❗ • **Azure Resource Manager**についての詳細は、[次のリンク](#)を参照してください。
- Azure Monitorでサポートされるメトリクス情報は、[次のリンク](#)を参照してください。

必要権限を設定する

権限案内

WhaTapのFunction Appがモニタリング機能を正しく活用するには、当該リソースグループ(Resource group)やサブスクリプション(Subscription)に**Monitoring Reader**役割を割り当てる必要があります。しかし、この役割を割り当てる行為自体は、権限が必要な作業です。つまり、**Microsoft.Authorization/roleAssignments/write**権限が必要であり、この権限は一般的なユーザーロールには含まれません。

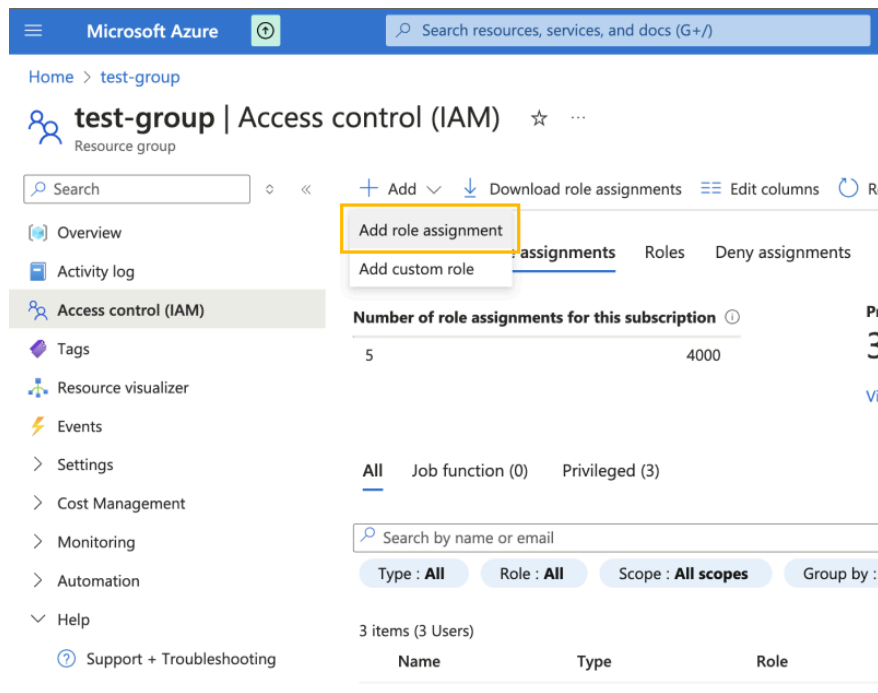
従って、**Monitoring Reader**役割を割り当てるためには、これらの高度な権限を含む**Role Based Access Control Administrator**または**User Access Administrator**役割が必要です。

- ⓘ • 当該リソースグループ(Resource group)やサブスクリプション(Subscription)に**Monitoring Reader**役割がすでに割り当てられている場合は、このコースは省略しても構いません。
- Azure役割の割り当てについての詳細は、[次のリンク](#)を参照してください。

権限を割り当てる

最小限の範囲のためにサブスクリプション(Subscription)より下位階層のリソースグループ(Resource group)を基準にご案内します。

1. リソースを含むグループダッシュボードで**Access control (IAM) > Add > Add role assignment**順番に選択します。



2. Add role assignment画面でPrivileged administrator rolesタブを選択します。

Microsoft Azure

Search resources, services, and docs (G+/)

Home > test-group | Access control (IAM) >

Add role assignment

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Job function roles **Privileged administrator roles**

Grant privileged administrator access, such as the ability to assign roles to other users.

⚠ Can a job function role with less access be used instead?

Search by role name, description, permission, or ID

Type: All Category: All

Name ↑↓	Description ↑↓	Type ↑↓	Category ↑↓
Owner	Grants full access to manage all resources, including ...	BuiltInRole	General
Contributor	Grants full access to manage all resources, but does ...	BuiltInRole	General
Access Review Operat...	Lets you grant Access Review System app permissio...	BuiltInRole	None
Role Based Access Co...	Manage access to Azure resources by assigning role...	BuiltInRole	None
User Access Administr...	Lets you manage user access to Azure resources.	BuiltInRole	General

Showing 1 - 5 of 5 results.

Review + assign Previous Next

3. **Role base Access Controle Administator** または**User Access Administrator**項目を選択します。
4. 画面下段の**Next**ボタンを選択します。
5. **Assign access to**項目で**User, group, or service principal**を選択し、**Select Members**を選択し、ユーザー (User)を追加します。

Microsoft Azure

Search resources, services, and docs (G+)

Home > test-group | Access control (IAM) >

Add role assignment

Role **Members** Conditions Review + assign

Selected role User Access Administrator

Assign access to ☒ User, group, or service principal
☐ Managed identity

Members + Select members

Name	Object ID	Type
Thomas.Hofmann	a1b2c3d4-5678-9010-1111-222222222222	User

Description Optional

Review + assign Previous Next

- 画面下段のNextボタンを選択します。
- Recommended項目を選択してReview + assignボタンをクリックします。

Microsoft Azure

Home > test-group | Access control (IAM) >

Add role assignment

Role Members **Conditions** Review + assign

Selected role
User Access Administrator

What user can do

- ☐ Allow user to only assign selected roles to selected principals (fewer privileges)
- ☒ Allow user to assign all roles except privileged administrator roles Owner, UAA, RBAC (Recommended)
- ☐ Allow user to assign all roles (highly privileged)

Condition

Preconfigured (All roles except privilege built-in roles)

Exclude roles:
Owner, User Access Administrator, Role Based Access Control Administrator

Review + assign Previous Next

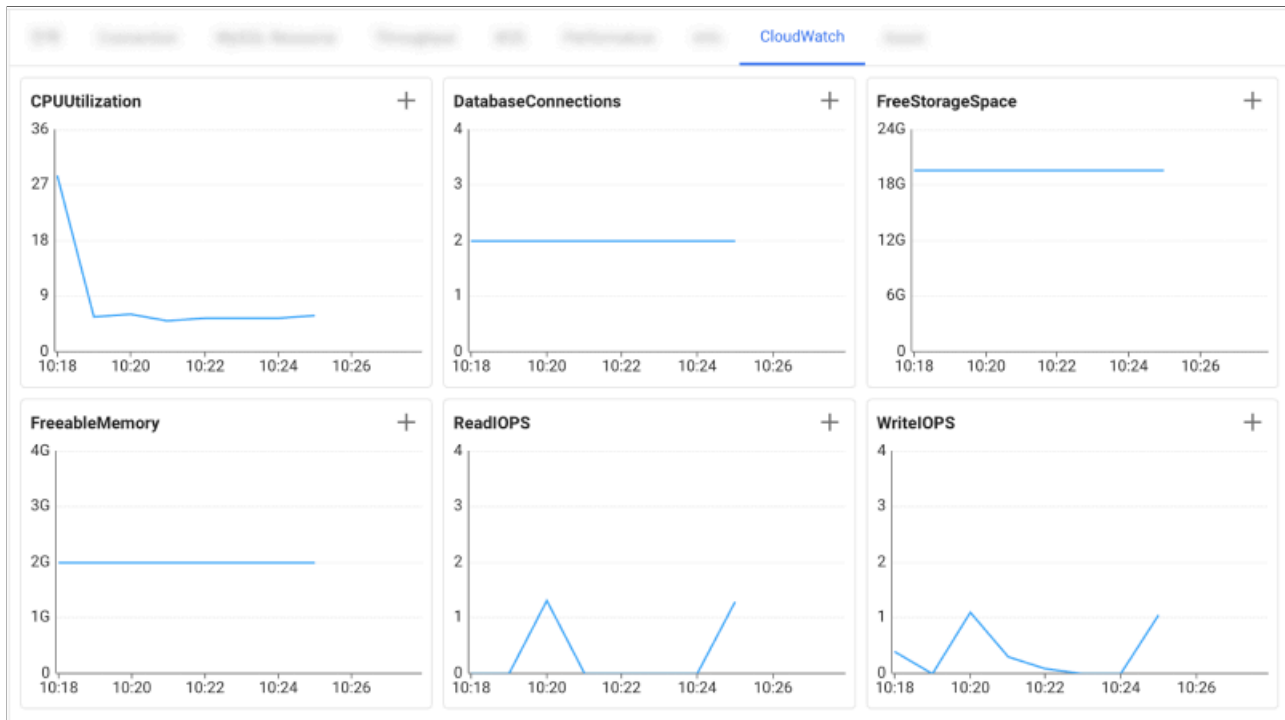
❗ fewer privileges項目を選択するには、Monitoring Reader役割のみ許可します。

収集メトリクスを確認する

クラウドデータベース接続を完了すると、画面に追加されたクラウドサービスセクションが生成されます。クラウドデータベースに接続されたインスタンス(エージェント) および収集されるメトリクスの一覧を確認できます。

追加されたクラウドサービス		
過去5分以内にクラウド指標が収集された場合に 표시됩니다。		
インストールされたエ...	クラウドサービス	収集指標
My-DB-Instance	Monitor	cpu_percent, memory_percent, network_bytes_egress, network_bytes_ingress
My-DB-Instance	Monitor	cpu_percent, memory_percent, network_bytes_egress, network_bytes_ingress

Azure Monitorから収集したメトリクスはAzureカテゴリに含まれます。



次のメニューパスで確認できます。

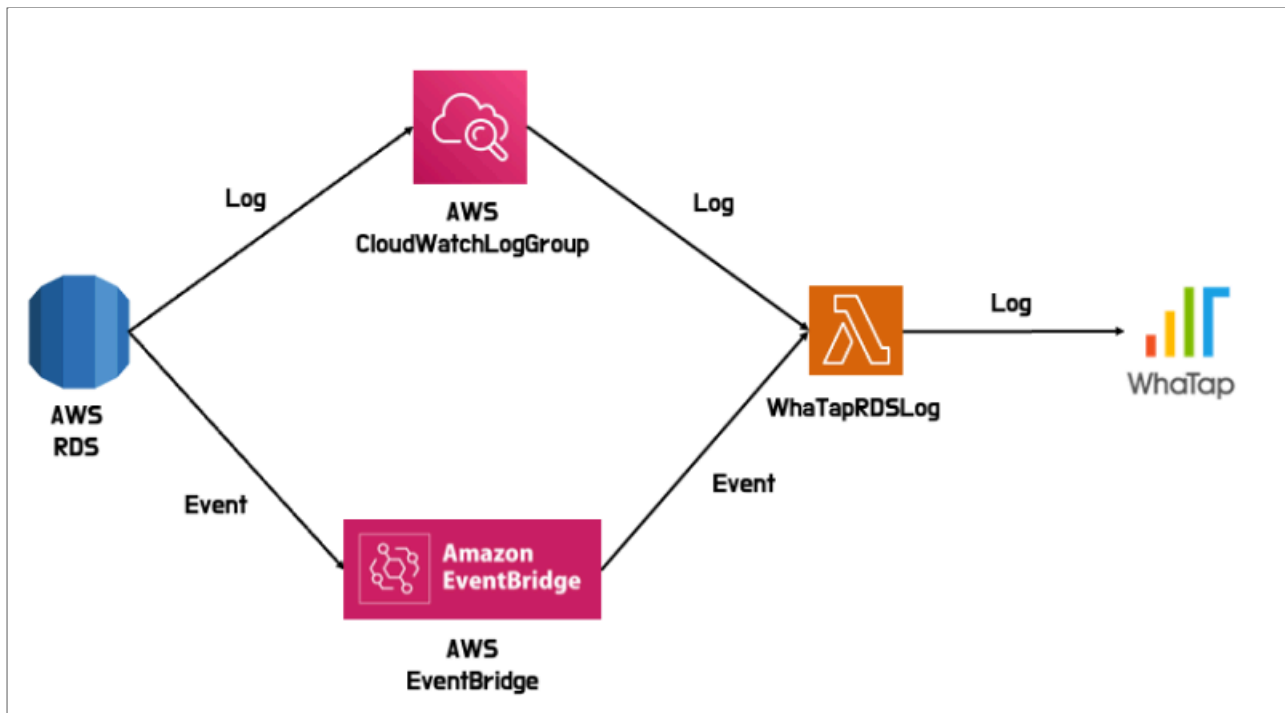
- **ダッシュボード>インスタンスモニタリング:** ウィジェットの⚙️ボタンを選択してメトリクスを選択できます。
- **ダッシュボード>マルチインスタンス:** ユーザー定義ウィジェットを追加してメトリクスを選択できます。

- ❗
- **インスタンスモニタリング**メニューでウィジェットのメトリクスを変更する方法は、[次の文書](#)を参照してください。
 - **マルチインスタンス**メニューでユーザー定義ウィジェットを追加する方法は、[次の文書](#)を参照してください。
 - **分析>メトリクス照会**メニューでカテゴリ項目をdb_azure_databaseに選択します。関連したタグとフィールド情報を確認できます。

サーバーレスログ収集設定

AWS RDS Logに適用する

AWS RDS Logモニタリングのための構成



AWSではRDSのモニタリング向けに様々な情報を提供しますが、これらの情報を確認するにはAWSコンソールから直接アクセスする必要があります。しかし、WhaTapのAWS RDS LogはデータベースモニタリングのログメニューからRDSのログとイベントを提供します。

WhaTapのAWS RDS LogはAWS CloudFormationを通じてインストールできます。AWS CloudFormationについての詳細は[次のリンク](#)を参照してください。

権限確認

AWS CloudFormationを通じてWhaTapのAWS RDS Logをインストールするには、まず必要なIAM権限を確認してくだ

さい。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:UpdateStack",
        "cloudformation>DeleteStack",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackEvents",
        "cloudformation:DescribeStackResource"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:PutBucketVersioning",
        "s3:PutBucketPublicAccessBlock"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "lambda:CreateFunction",
        "lambda:UpdateFunctionCode",
        "lambda:UpdateFunctionConfiguration",
        "lambda>DeleteFunction",
        "lambda:AddPermission"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
```



```

    "iam:GetRole",
    "iam:AttachRolePolicy"
  ],
  "Resource": "*"
}
]
}

```

インストールと設定

1. Whatapが提供する**AWS CloudFormation**テンプレートを通じてインストールできます。次のリンクへ移動してから、**パラメータセクション**で以下の項目を入力します。

[AWS CloudFormation Install](#)

- **Host:** Whatap収集サーバーアドレス。Whatapプロジェクトの**管理 > エージェントインストールメニュー**で確認できます。例) 13.124.11.223/13.209.172.35
 - **MemorySize:** WhatapRDSLogのメモリサイズ。128MB ~ 10,240MB間の値を入力します。
 - **Pcode:** Whatapプロジェクトのプロジェクトコード。**管理 > プロジェクト管理メニュー**で**プロジェクトコード**項目を確認してください。
 - **ProjectAccessKey:** Whatapプロジェクトのアクセスキー。**管理 > プロジェクト管理メニュー**で**プロジェクトアクセスキー**項目を確認してください。
 - **TimeOut:** インストールされるWhatapRDSLogが実行時に最大に維持できる時間です。
 - **UseReservedConcurrency:** Whatap RDSの予約された同時性の使用有無です。
 - **ReservedConcurrency:** Whatap RDSが予約された同時性を使用したときに、同時に実行できる最大インスタンスの数です。
2. ログモニタリングの詳細設定のために、次のリンクへ移動してください。

[AWS CloudFormation Setting](#)

3. **パラメータセクション**で以下の項目を順番に入力します。
 - **AwsRdsNames:** イベントを受信したいRDS Cluster、Instanceの名前一覧
 - 例) database-1、seoul-pro-db-01、seoul-pro-db-01-writer …

- EventRuleを生成しないためには、 `none` を入力します。
- **RdsLogGroupName(1~9):** ログを受信したいRDS CloudWatchLogGroup Name
 - 例) `/aws/rds/seoul-pro-db-01`
 - サブスクリプションフィルターを生成しない場合は、 `none` を入力します。

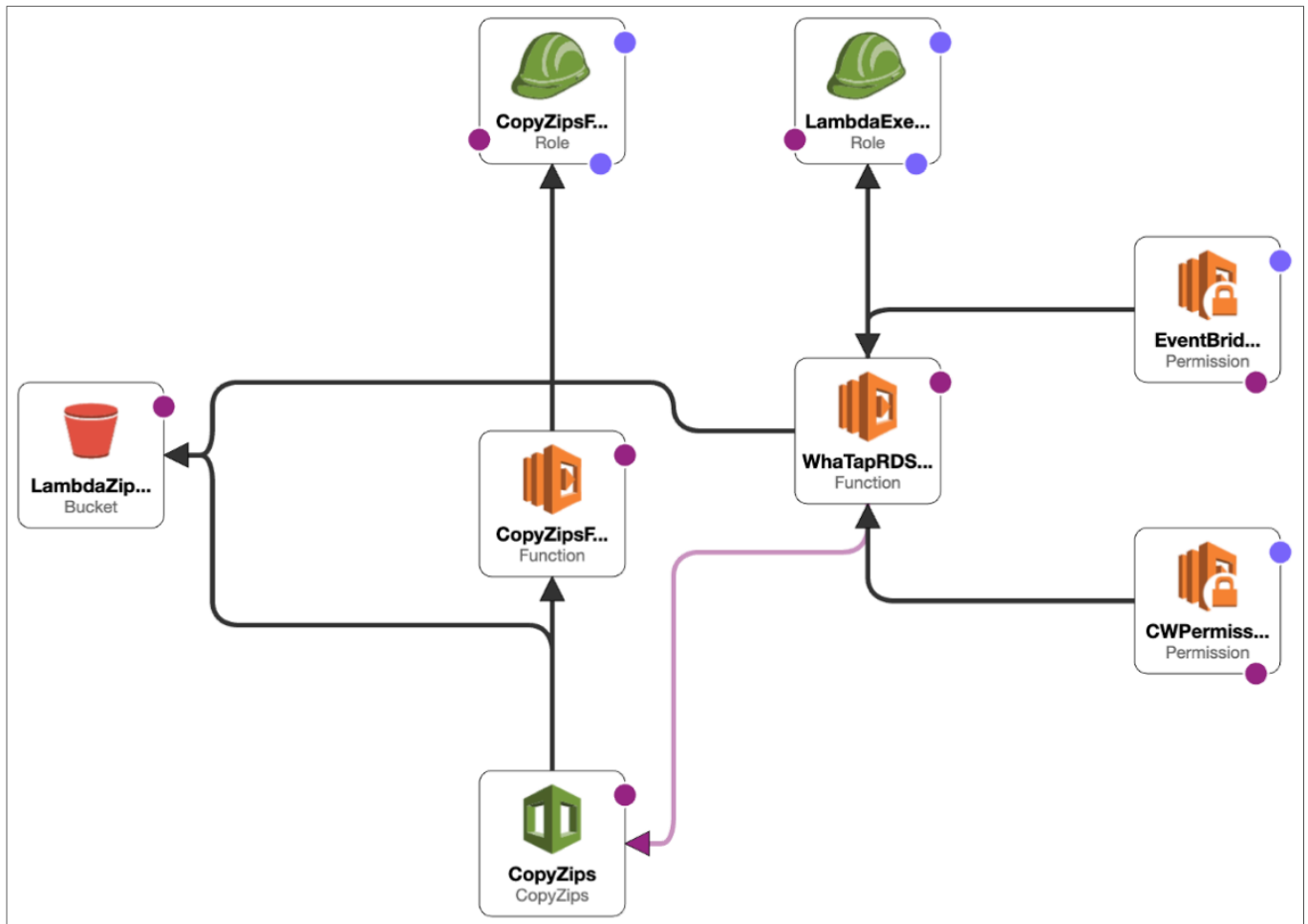
❗ CloudWatchLogGroupは一度に最大9つまで接続できます。

- **WhaTapRDSLogArn:** 前の段階で生成されたWhaTapRDSLogArn

Outputs (1)		
Search outputs		
Key	Value	Description
WhaTapRDSLogArn	arn:aws:lambda:ap-northeast-2:911937781722:function:whataprdseventforwarder-WhaTapRDSLog-nLFyYmp1Od5s	WhaTapRDSLog's Arn

生成リソースを確認する

AWS CloudFormationからインストールされているリソースを確認します。



- S3

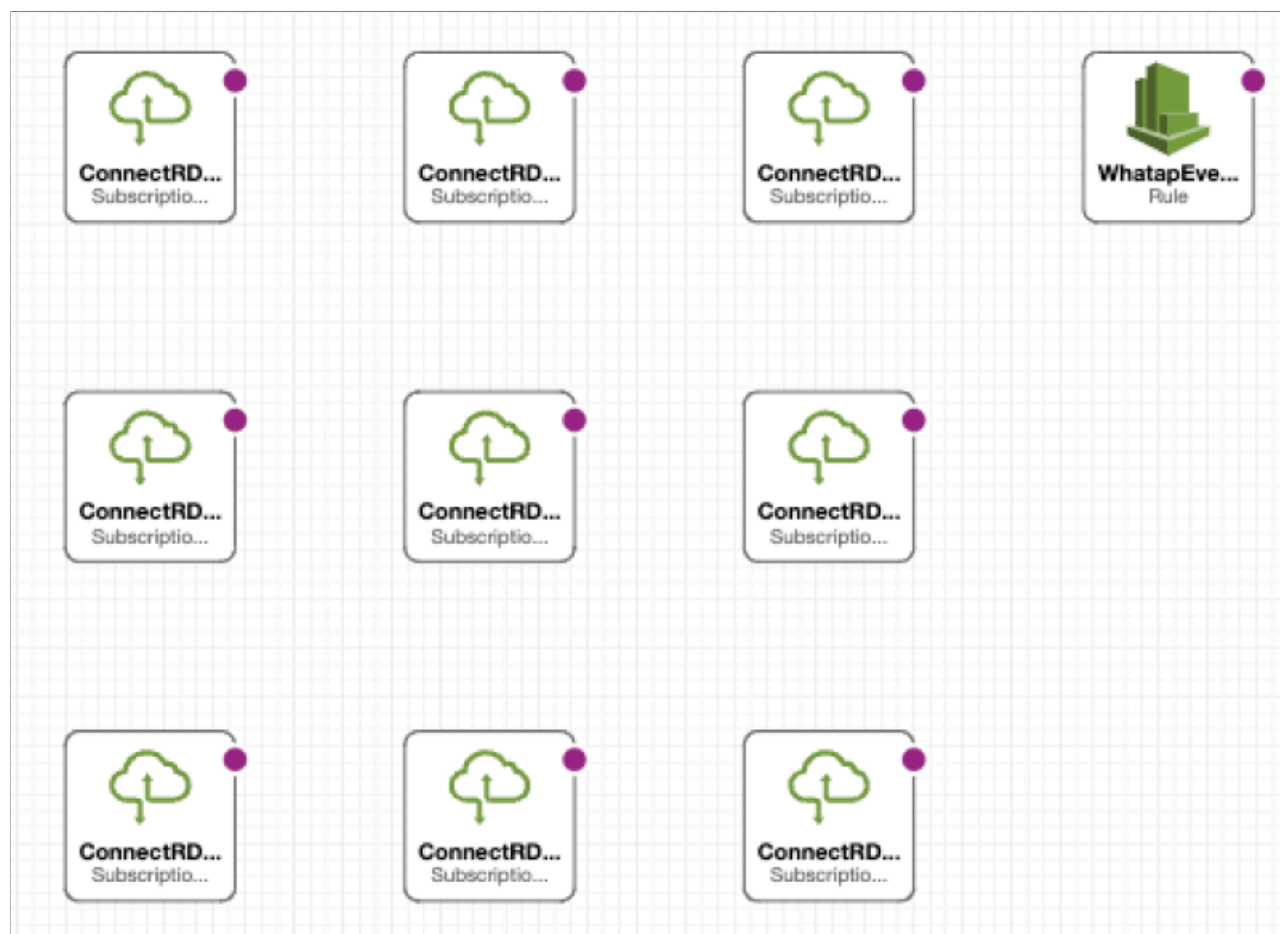
- Bucket : WhaTap**AWS RDS Log**の実行ファイルを保存するAWS S3 Bucket (CopyZips)

- Lambda

- Function
 - CopyZipsFunction:WhaTap AWS S3 Bucketからユーザーの環境に実行ファイルを移すAWS Lambda Function
 - WhaTapRDSLog : AWS RDSのモニタリングのためのAWS Lambda Function
- Permission
 - CWPermission: RDS CloudWatchLogGroupにLogが生成される時、WhaTapRDSLogを実行できるようにするPermission

- EventBridgePermission: RDS Eventが発生した場合、EventBridgeでWhaTapRDSLogを実行できるようにするPermission
- Role
 - CopyZipsFuntionがWhaTapのS3からファイルをインポートし、CopyZipsに入れるために必要なRole
 - WhaTapRDSLogがCloudWatchLogGroupにLogを残すために必要なRole

フィルターとルールを確認する



• WS CloudWatchLogGroup

SubscriptionFilter(1~9): RDSの CloudWatch LogGroupに Logが入ってくるとWhatapRDSLogを実行させるフィルター

- **AWS EventBridge**

Rule: AWSイベントで、RDSと関連したイベントが発生するとWhatapRDSLogを実行させるルール

管理

エージェントのアップデート

最新バージョンにアップデートする

ホーム画面 > プロジェクト選択 > 管理 > エージェントアップデート

エージェントアップデート

エージェントアップデートのご案内

「アップデート」ボタンをクリックすると、最新のエージェントファイルがエージェントのインストール場所にダウンロードされ、自動的に再起動されます。アップデートには1分がかかります。完了すると、該当のエージェントのバージョンが最新バージョンに変更されます。

① Windows環境にインストールされたエージェントは、自己更新機能を提供していません。 [詳細](#)

② 更新が正しく行われない場合は、[WhaTap Docs](#) を参考してください。

エージェントリスト

最新エージェントのバージョン 2.3.3 [すべてのエージェントを最新バージョンに更新](#) [旧バージョン](#)

no.	id	ip	name	agent path	version	update
1	704779695	192.168.1.101	WhaTap-001	C:\Program Files\WhaTap\agent.exe	2.2.8 20241212	アップデート
2	1652207373	192.168.1.102	WhaTap-002	C:\Program Files\WhaTap\agent.exe	2.2.8 20241212	アップデート
3	755606486	192.168.1.103	WhaTap-003	C:\Program Files\WhaTap\agent.exe	2.1.4 20241101	アップデート

エージェントをアップデートすると、最新バージョンのエージェントファイルをエージェントインストールパス ([agent path](#))に自動的にダウンロードして再起動します。エージェントのアップデートには約1分かかり、アップデート過程でダウンロードとアップデートの成功メッセージが画面上に順番に表示されます。アップデートを完了すると、そのエージェントのバージョンが自動的に最新バージョンに変更されます。

- アップデートできるエージェントは、**アップデート**ボタンが活性化されます。
- エージェントを個別にアップデートするには、各エージェント項目の右側にある**アップデート**ボタンを選択します。
- すべてのエージェントを一度に更新するには、**すべてのエージェントを最新バージョンに更新**ボタンを選択します。

✔ エージェントのアップデート中に別のメニューに移動する場合は、進行状況を確認することはできず、1つのエージェントをアップデートしながら別のエージェントを同時にアップデートすることはできません。

- ❗ Windows環境にインストールされたエージェントは、自動更新機能をサポートしていません。手動で最新のエージェントに再インストールするか、最新の `whatap.agent.dbx-*.jar` ファイルを既存のエージェントパスに入れて、サービスを再登録します。

```
# 1. Delete Windows service
install_WindowsService.bat delete WhatapDBXAgent # existing service name

# 2. Create Windows service
install_WindowsService.bat create WhatapDBXAgent # existing service name
```

アップデートに失敗した場合

エージェントが再起動しない場合

アップデート中にダウンロード成功メッセージが表示されても、エージェントのバージョンが最新に自動的にアップデートされない場合は、手動でエージェントを再起動する必要があります。エージェントのインストールパス([agent path](#))で`stop.sh`、`startd.sh`バッチファイルを順番に実行します。

エージェント手動ダウンロード

ダウンロードができない環境の場合は、次の手順でエージェントのインストールを進めてください。

1. 管理 > エージェントインストールメニューに移動します。
2. インストールの案内セクションの2番項目で**Download**ボタンを選択し、最新バージョンのエージェントをダウンロードします。
3. ダウンロードしたファイルを解凍します。
4. `whatap.agent.dbx-X.Y.Z.jar` ファイルだけをエージェントインストールパス([agent path](#))にアップロードします。
5. `stop.sh`、`startd.sh`バッチファイルを順番に実行して、エージェントを再起動します。

前のバージョンに戻す

1. 画面右上の旧バージョンボタンを選択します。

2. ダウンロード可能なエージェントの一覧が表示されます。
3. 希望するバージョンを選択します。
4. 選択したバージョンに戻すエージェント項目で **ダウンロード** ボタンを選択します。
5. 選択したエージェントのインストールパスでエージェントファイルをダウンロードします。
6. エージェントのインストールパスで上位バージョンの `whatap.agent.dbx-X.Y.Z.jar` ファイルを削除します。
7. `stop.sh`、`startd.sh` バッチファイルを順番に実行して、エージェントを再起動します。

❗ バッチファイルがエージェントのインストールパスで最も高いバージョンのエージェントファイルを実行するように設定されているため、上位バージョンのJARファイルを削除しないと、以前のバージョンのエージェントを実行できません。

❗ PostgreSQLモニタリングを利用するには、DBXエージェントの最小バージョンは1.6.7です。

エージェント削除

エージェントがインストールされたサーバーから削除する

• [Linux](#) Windows

サーバー上で起動されるJavaプロセスから、`whatap.agent.dbx` 文字列を含むプロセスを終了します。エージェントをインストールしたパスから`stop.sh`を実行します。その後、エージェントファイルとエージェントに登録したUser権限を削除します。

サーバー上で起動するJavaプロセスからエージェントファイル名のjarファイルで起動中のプロセスを見つけて停止します。もし、batファイルをForegroundで実行中の場合は、そのbat実行スクリプトを終了します。その後、エージェントファイルとエージェントに登録したUser権限を削除します。

サービス画面から削除する

WhaTapモニタリングサービスからエージェントを削除するには、**エージェントリスト**メニューに移動します。エージェントを削除する前に、エージェントをインストールしたサーバーでエージェントプロセスを終了する必要があります。

す。

1. プロジェクトリストからエージェントを削除するプロジェクトを選択します。
2. 左側のメニューから**管理** > **エージェントリスト**を選択します。
3. エージェントリストで削除するエージェントで右側の  ボタンをクリックします。

- ⓘ • 非活性化状態のエージェントのみ削除できます。

ステータス	名前 (初期)	ID
 活性	 	
 非活性	 	
 活性	 	

- エージェントのバージョンによっては、削除機能をサポートしていない場合があります。最新バージョンのエージェントに更新します。

ダッシュボード

WhaTapエージェントのインストールが完了しましたら、[WhaTapモニタリングサービス](#)にログインしてください。初期画面はプロジェクトリストです。プロジェクトリストで作成したプロジェクトを確認できます。プロジェクトリストで作成したプロジェクトを選択すると、[インスタンスリスト](#)が表示されます。

WhaTapデータベースモニタリングサービスでダッシュボードは、プロジェクトの全体のステータスを一目で確認できる機能です。モニタリングされているリソース全体の使用量を確認でき、リアルタイムダッシュボードでモニタリング対象のリソースをフィルタリングできます。

- 主なパフォーマンス指標とステータスを表示するように構成されています。パフォーマンス指標は5秒間隔で更新されます。
- 現在使用中のアクティブセッション(Queryを含む)、ロック情報および、プロセス情報を確認することができます。Public CloudとPrivate Cloudの両方に対応するために、クエリベースのモニタリングを使用して構成しました。
- 1つのインスタンスで複数のDBをフィルタリングできます。
- 収集したデータに基づいて、特定の1日あたりの全体的なトレンドを分析できる機能を提供します。特定の時間に進行中だったセッションを特定し、低速なQueryを分析することができます。

インスタンスリスト

エージェントをインストールしたデータベースサーバーの一覧を確認できます。主なパフォーマンス指標と状態(Status)を表示できるように構成されています。パフォーマンス指標は5秒間隔で更新され、一覧のカラムは設定を使用して目的のカラムに配置できます。詳細については、[次の文書](#)を参照してください。

インスタンスモニタリング

- データベースサーバーの主要メトリクスとアクティブセッションに関する詳細情報をリアルタイムでモニタリングし、問題になる可能性のあるアラートを確認できます。関心のある重要メトリクスを選択して、全体的なシステム状況をモニタリングできます。
- リアルタイムモニタリングは、5秒間隔でパフォーマンス指標をモニタリングします。実行中の**アクティブセッション**、**ロックツリー**、**プロセス情報**を確認できます。

詳細については、[次の文書](#)を参照してください。

インスタンスモニタリング

インスタンスモニタリングメニューでは、1つのインスタンスだけをモニタリングすることに対して、**マルチインスタンス**は、複数のインスタンスのメトリクスを1つの画面で確認して比較できます。いくつかの重要なインスタンス指標をグルーピングした数値で、インスタンスの状況を確認できます。グラフチャートを使用すると、インスタンスの指標値の推移とどのインスタンスの指標値が高いかを簡単に確認できます。最も長く続いているアクティブセッション情報を提供します。詳細については、[次の文書](#)を参照してください。

スロークエリ

SQL別の応答時間分布を提供します。応答分布の遅延SQLの状態を直観的に確認し、対象を特定してトラブルの原因を特定することができます。詳細については、[次の文書](#)を参照してください。

インスタンスリスト

ホーム画面 > プロジェクト選択 > ダッシュボード > インスタンスリスト

エージェントをインストールしたデータベースサーバーの一覧を表示し、各インスタンスの主要なパフォーマンス指標と状態が把握できます。パフォーマンス指標は、5秒間隔で更新されるし、カラム選択ボタンで表示する指標を選択できます。

基本画面ガイド



- ① 領域は、プロジェクトに設定されているインスタンスの状態を表示します。合計にはテーブルに表示中のインスタンスの数、危険、警告、正常にはステータスカラムの各ステータスの数が表示されます。
 - 正常: 定常状態のエージェントの数です。
 - 警告: アラート通知 > 通知設定メニューで設定したイベントのしきい値が警告レベルを超えたエージェントの数です。
 - 危険: アラート通知 > 通知設定メニューで設定したイベントのしきい値が危険レベルを超えたエージェントの数です。

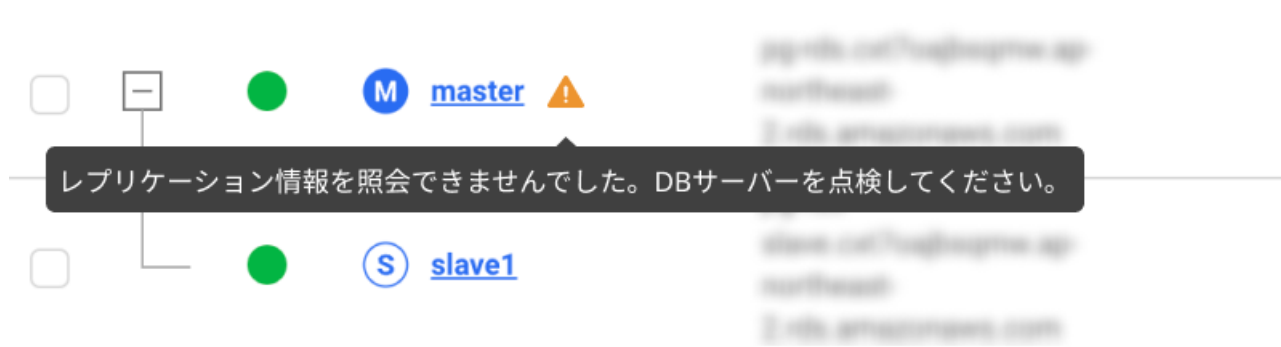
ⓘ アラート通知メニューの詳細については、[次の文書](#)を参照してください。

- ステータスカラムで正常ではない状態のアイコンをクリックするとイベント画面が表示されます。そのインスタンスで発生したイベントの履歴を確認できます。
- 各インスタンスを選択すると、インスタンスモニタリングメニューが表示されます。選択したインスタンスの詳細をリアルタイムでモニタリングできます。

① インスタンスモニタリングメニューの詳細については、[次の文書](#)を参考にしてください。

- 主要なパフォーマンス指標をグラフチャート形式で提供します。最も高い数値をチャートに表示します。チャートにカーソルを合わせると、その場所の数値が表示されます。パフォーマンス指標の詳細については、[次の文書](#)を参照してください。
- インスタンスリストでサーバー名カラムに表示された **M**、**S**、**C** は、それぞれMaster、Slave、Clusterを意味します。Replicationで構成したDBの場合に表示します。

Replicationインスタンスが異常な状態の場合 **!** アイコンが表示されます。



- 画面右上の **!** ボタンをクリックすると、リアルタイムの通知が確認できます。通知設定の詳細については、[次の文書](#)を参照してください。

① 画面上部にある固定領域の詳細については、[次の文書](#)を参照してください。

- ✓ インスタンスリストのサーバー名を変更するには、エージェント設定(whatap.conf)で `object_name` オプションを利用します。詳細については、[次の文書](#)を参照してください。

インスタンス性能指標の比較

2つ以上のインスタンスを選択して、性能指標を比較できます。性能指標を比較するインスタンスを一覧から選択すると、画面右上のマルチビューボタンが活性化されます。マルチビューボタンを選択すると、マルチインスタンスメニューに移動します。主要な性能指標のデータを比較したグラフチャートとアクティブセッション、ロックツリーに関する

る詳細情報を確認できます。

❗ マルチインスタンスメニューの詳細については、[次の文書](#)を参考にしてください。

カラム表示または非表示

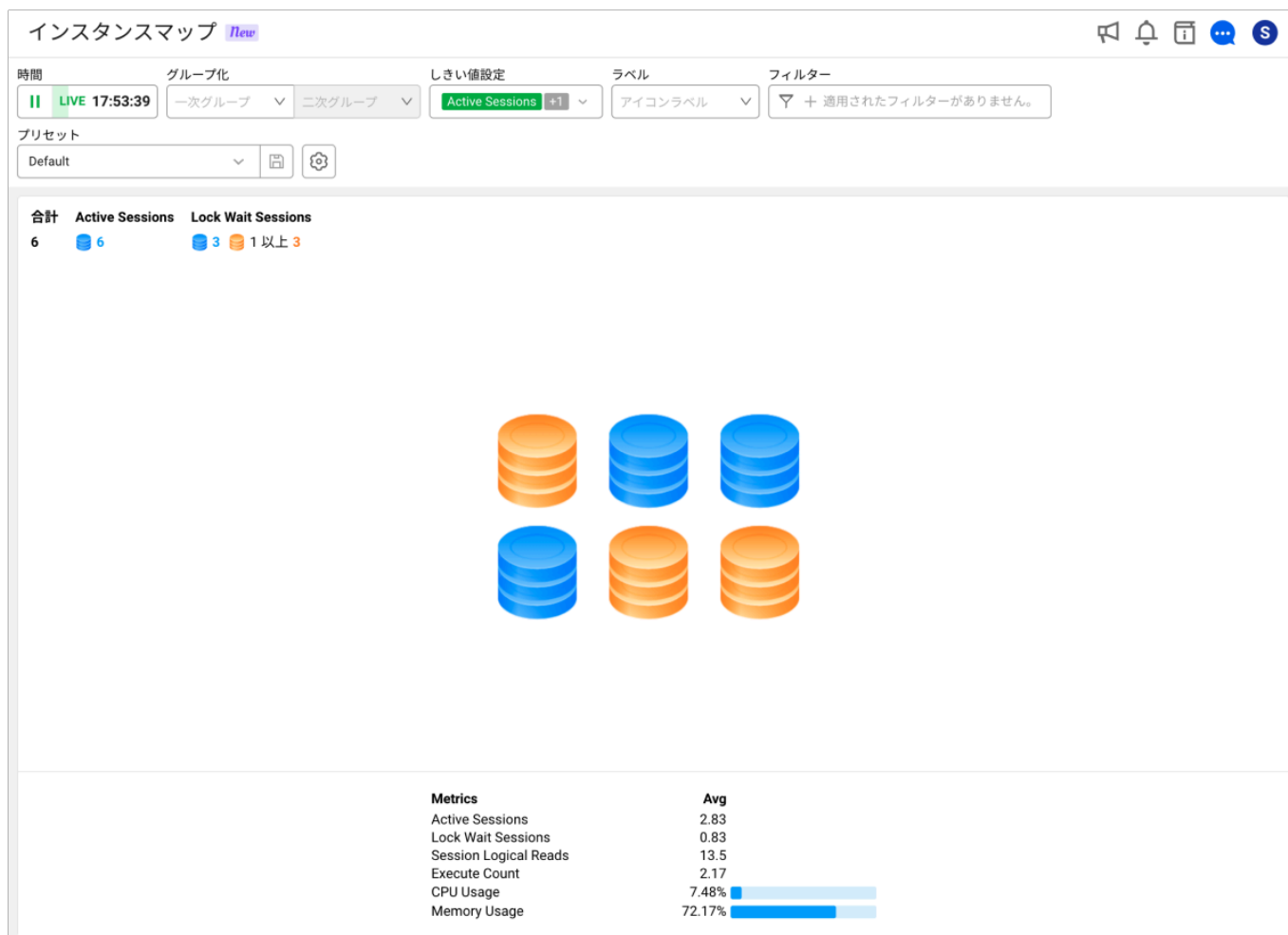
インスタンスリストのパフォーマンス指標カラムを表示または非表示できます。画面右上の**カラム選択**ボタンをクリックします。目的の項目を選択、または解除します。カラムを表示または非表示できます。

インスタンスマップ

ホーム画面>プロジェクト選択>**ダッシュボード**>インスタスマップ

インスタンスマップは、データベースサーバーをモニタリングしているインスタンスの状態を視覚化し、難しく感じられることの多いデータベースモニタリングを簡単に行えるよう支援する機能です。

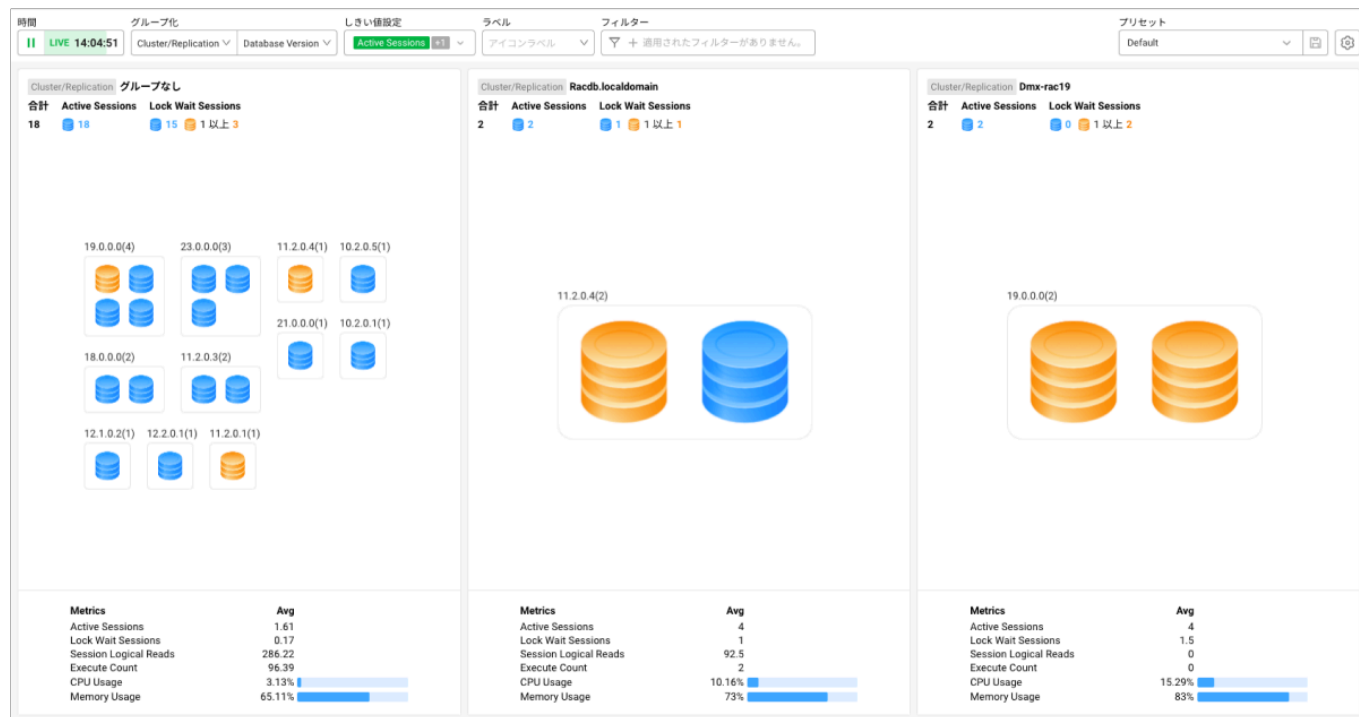
データベースサーバーの主要指標にしきい値（スレッシュホールド）を設定することで、データベースの状態を一目で把握でき、実行中のアクティブセッションもリアルタイムで確認できます。また、インスタンスモニタリング画面やマルチインスタンスモニタリング画面へ遷移して、障害が発生したデータベースの分析を迅速に行えるよう連携されています。



❗ CPU使用率、メモリ使用量など、データベースサーバーのプロセス使用状況はXOSエージェントの実行によりデータ収集が可能です。

基本画面

インスタンスマップは主に **ユーザー設定エリア** とインスタンスの **マップチャートエリア** で構成されています。



ユーザー設定

インスタンスマップ画面上部のユーザー設定エリアでは、グループ化、しきい値設定、ラベル、フィルター、プリセット、 追加設定が可能です。

グループ化

インスタンスをさまざまな基準でグループ化したり、ユーザーのニーズに応じてカスタム基準でまとめてモニタリングできます。グループ化機能は最大2階層までサポートされています。2階層目のグループをクリックすると、そのグル

ープに属するインスタンスのデータをまとめて確認できる詳細画面が表示されます。

しきい値の設定

データベースモニタリングの主要指標に対してしきい値を設定することで、インスタンスの状態を色で表現することができます。複数の指標を設定でき、各指標ごとに警告（Warning）および重大（Critical）の範囲を指定できます。正常（Normal）区間は青色で表示します。設定された範囲に応じて、該当状態の色が適用されます。

1. 画面上の **しきい値設定** オプションをクリックしてください。
2. 任意の指標を選択し、Warning と Critical の範囲を設定してください。
3. メトリクスを追加する場合は、下段の **+ 指標追加** をクリックしてください。
4. 各指標のしきい値設定が完了したら、**保存** ボタンをクリックしてください。

インスタンス状態の表示優先順位

状態表示の優先順位はCritical区間が一番高いです。複数の指標のうち、いずれか1つでも Critical の範囲に該当する場合、そのインスタンスは赤色で表示されます。以下の例では、2つのインスタンスがともに Critical として表示されます。

ラベル

インスタンスアイコンの内部にはテキストラベルが表示されます。ラベルの種類は、データベース情報および主要指標で構成されています。

フィルター

フィルターを指定することで、特定のインスタンスに集中してモニタリングを行うことができます。

- **フィルターキー:** データベースおよびインスタンス情報がフィルターキーとして提供されます。
 - **収集カラム:** インスタンス名、クラスタ/レプリケーション情報、エージェントバージョン、CPUコア数、データベースの種類、インストール位置、バージョン、IP、Port
- **条件:** フィルターの条件は、 $\geq$ 、 $\leq$ 、 $>$ 、 $<$ などの数値型条件と一致、不一致、包含、不包含などの文字列条件で設定できます。

プリセット

ユーザーは、設定したオプションをプリセットに保存し、必要な時読み込みできます。プリセットはプロジェクト単位で保存されます。これにより、プロジェクトメンバーが同じ設定でインスタンスをモニタリングすることができます。デフォルトプリセット(Default) が提供されており、デフォルトプリセットは削除できません。

✔ 基本プリセット

インスタンスマップのデフォルトプリセットでは、`active_sessions` と `lock_wait_sessions` のしきい値が以下のように設定されています。

- `active_sessions`
 - **Warning:** 値が 20 以上
 - **Critical:** 値が 30 以上
- `lock_wait_sessions`
 - **Warning:** 値が 1 以上
 - **Critical:** 値が 3 以上

追加設定

インスタンスマップ右上の ⚙️ アイコンをクリックして、追加設定が可能です。

- **第1グループなしを表示:** 第1グループを設定した際、該当グループが存在しない'グループなし'を表示するかどうかを設定します。
- **グループ下部メトリクスエリアの表示:** 下部のメトリクス情報の要約を表示するかどうかを設定します

インスタンスマップチャート

インスタンスマップチャートは、3つのエリアで構成されています。

上部エリア

インスタンスの全体数、またはグループ化されている場合はグループごとの総数と、選択されたメトリクスの状態別の件数を表示します。

インスタンスの可視化エリア

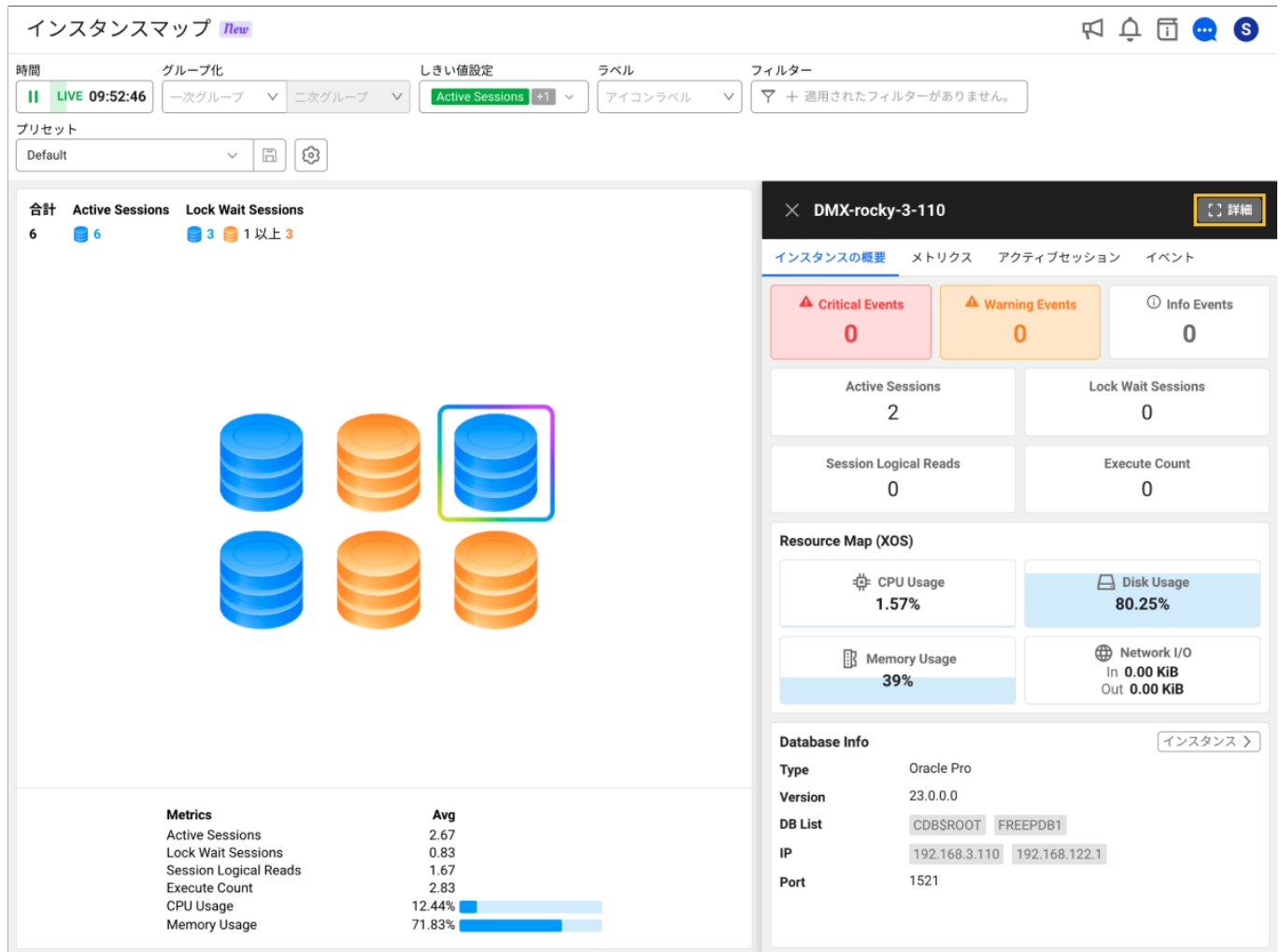
すべてのインスタンスの状態と件数を直感的に把握できるビジュアルダッシュボードを提供します。

下部エリア

主要なメトリクスのインスタンス平均値を表示します。⚙ の **グループ下部メトリクスエリアの表示** オプションを使ってオン/オフを切り替えることができます。

要約の表示と詳細分析

インスタンスマップで、確認したいインスタンスを選択してください。右側に表示されるサマリービュー画面では、該当インスタンスの概要情報を **インスタンス概要**、**メトリクス**、**アクティブセッション**、**イベント**、**ログタブ** で確認できます。サマリービュー画面で **詳細分析** ボタンをクリックすると、詳細分析画面を確認できます。



インスタンス概要

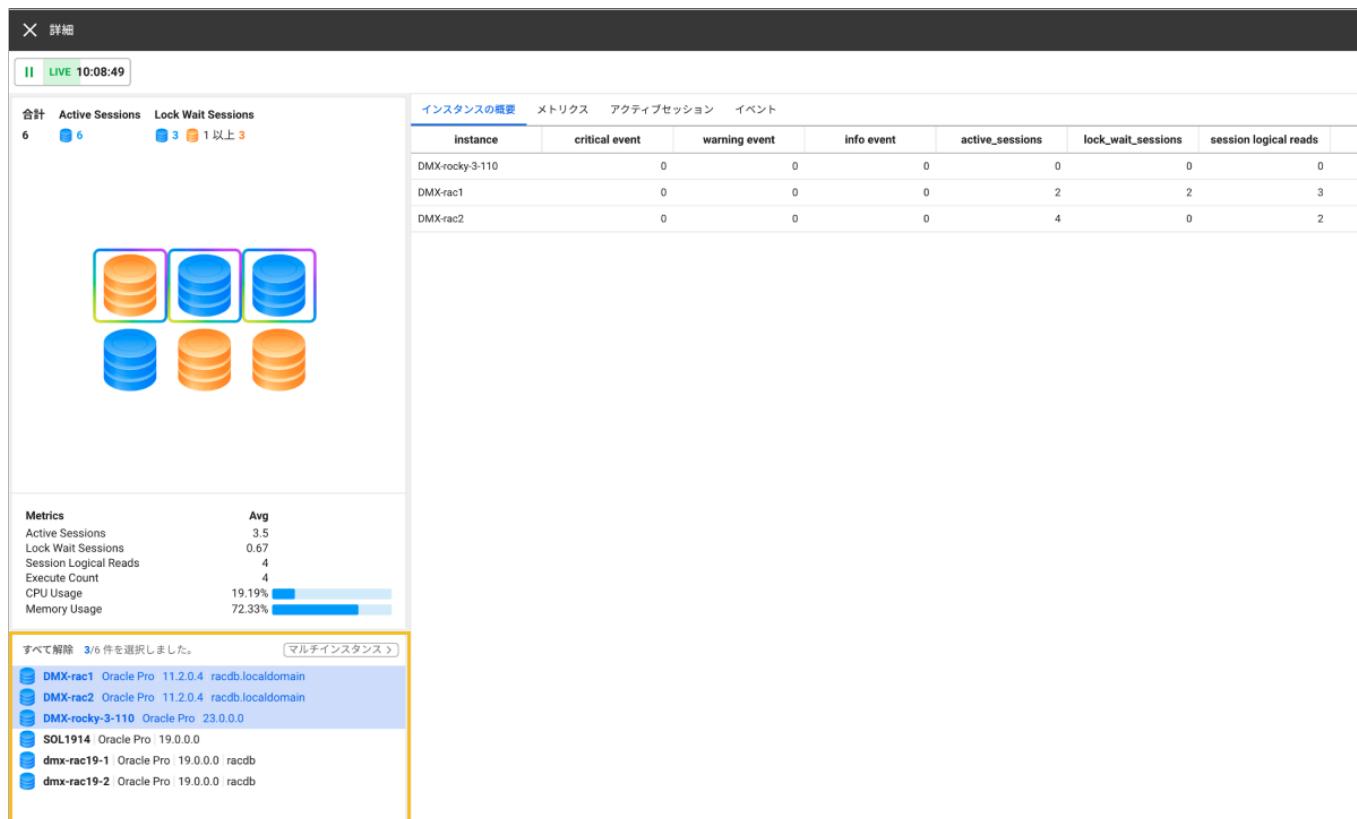
詳細分析ページの **インスタンス概要** タブでは、選択したインスタンスがモニタリングしているデータベースの情報および指標情報、発生したイベントを確認できます。また、インスタンスモニタリングメニューへ移動することもでき、非アクティブなインスタンスの場合は、非アクティブ化された時点のインスタンスモニタリング画面へ移動します。

- **イベントエリア:** ユーザーが設定したアラート基準に基づくイベント発生情報を提供します。各イベントカードをクリックすると、どのイベントが発生したかを確認できます。
- **指標エリア:** 主要なデータベースメトリクスおよびリソース使用状況をリアルタイムで確認できます。リソース使用状況は XOS エージェントを設定することで確認可能です。
 - データベース指標

- Active Sessions: 実行中のクエリセッション数
- Lock Wait Sessions: ロック待機中のセッション数
- Session Logical Reads: メモリから読み取ったデータブロック数 (db block gets + consistent gets)
- Execute Count: 実行された SQL ステートメントの呼び出し回数
- リソース使用状況 (Resource Map (XOS))
 - CPU Usage: CPU 使用率
 - Disk Usage: ディスク使用量
 - Memory Usage: メモリ使用量および使用率
 - Network I/O: ネットワークの送受信量
- データベース情報エリア (Database Info) : データベースの詳細情報を確認できます。 インスタンスモニタリングをクリックすると、選択したインスタンスのモニタリング画面へ移動できます。

インスタンス概要詳細ビュー

詳細ビュー画面の **インスタンス概要** タブでは、複数のインスタンスを選択して、主要メトリクスおよびデータベース情報をテーブル形式でモニタリングできます。 **インスタンス概要** タブの右上にある **詳細ビュー** ボタンをクリックするか、2次グループを設定後、インスタンス視覚化エリアで2次グループ名をクリックすると、詳細ビュー画面に移動します。



- ・詳細ビュー画面では、視覚化エリア内のインスタンスアイコンをクリックするか、画面左下のインスタンスリストをクリックして、インスタンスを選択または選択解除できます。**すべて選択** または **すべて解除** をクリックすることで、全インスタンスの一括選択または解除が可能です。
- ・複数のインスタンスを選択した場合、**マルチインスタンスモニタリング** ボタンをクリックして、選択されたインスタンスを **マルチインスタンスモニタリング** 画面でモニタリングできます。

メトリクス

メトリクスタブでは、選択されたインスタンスのメトリクストレンドを確認できます。時間セレクターを操作することで、リアルタイムまたは過去のメトリクスの変化を確認できます。

アクティブセッション

アクティブセッションタブでは、リアルタイムで実行中のセッションおよびクエリの内容を確認できます。

❗ テーブルリスト内の文字色は、黒 → オレンジ → 赤 の順にセッションの実行速度が遅いことを示します。

イベント

イベント タブでは、直近1時間に発生したイベントを確認できます。発生したイベント項目をクリックすると、イベントが発生した時点のインスタンスモニタリング画面に移動します。

ログ

ログ タブでは、発生したログの確認およびリアルタイムログの確認が可能です。右側の **詳細ビュー** ボタンをクリックすると、より詳細なログを確認できます。

トポロジー

トポロジー画面は、データベースのレプリケーション・クラスター構造をグラフで可視化する機能です。マスターとレプリカ(スレーブ・スタンバイ)の接続関係、各ノードの健全性状態、レプリケーション遅延や切断の有無を1つの画面で把握できます。

❗ サポートプラットフォーム

トポロジー画面は、次のプラットフォームのレプリケーション構造をサポートします。

- MySQL(Replication)
- PostgreSQL(Replication)

トポロジーメニューは上記2つのプラットフォームで表示されます。その他のプラットフォームはサポート予定です。

画面への進入

ホーム画面> プロジェクトを選択した後、左メニューから**トポロジー**を選択します。

画面構成

トポロジー画面は、グラフ領域と補助パネルで構成されます。

- **サマリーカード**：全ノード数、クラスター数、問題のあるノード数など、レプリケーション構造の要約情報を表示します。
- **凡例**：ノード状態の色と役割、エッジ(接続線)の意味を案内します。
- **問題パネル**：レプリケーション遅延や切断など、注意が必要な項目を一覧で表示します。項目を選択すると、該当ノードに画面が移動します。パネルは折りたたみ・展開できます。
- **ミニマップ**：グラフ全体で現在表示している領域の位置を表示します。
- **コントロール**：画面の拡大・縮小(ズーム)と画面フィット機能を提供します。

ノードとエッジの読み方

- **ノード**：データベースインスタンス1つを表します。ノードの色は健全性状態を、ノードに表示される役割表示はマスター・レプリカなどレプリケーション構造での役割を表します。
- **エッジ(接続線)**：マスターからレプリカへつながるレプリケーション関係を表します。レプリケーション遅延の程度は線の色で、レプリケーションが切断された関係は点線で表示します。同期・非同期レプリケーションは矢印で区別します。
- **クラスター**：同じレプリケーショングループに属するノードをボックスでまとめて表示します。ボックスにはグループ名とプラットフォームが一緒に表示されます。

❗ レプリケーショントポロジーのノードの色は、該当ノードで発生したイベント(アラート)を基準に表示されます。

2Dビューと3Dビュー

画面上部のビュー切り替えボタンで2Dと3Dの表示を切り替えることができます。3Dビューでは、ドラッグで視点を回転し、ホイールで拡大・縮小できます。両ビューともダークモードをサポートします。

詳細情報を確認する

ノード、エッジ、クラスターを選択すると、画面右側に詳細情報ウィンドウ(Drawer)が表示されます。選択した対象は強調表示され、関連のない対象は淡く表示されます。各要素にマウスポインターを合わせると、要約情報をツールチップで確認できます。

AI診断

ノードまたはクラスターの詳細情報ウィンドウで**AI診断**を実行すると、選択した時点のレプリケーション状態を基準にAIがレプリケーション構造を分析し、結果を提供します。分析は、進行ステップ、分析内容、診断結果(verdict)の順に表示されます。

ⓘ AI診断は、レプリケーショントポロジーをサポートするプラットフォーム(現在MySQL、PostgreSQL)でノードまたはクラスター単位で使用できます。その他のプラットフォームは今後サポート予定です。

Flexボードウィジェット

統合Flexボードにデータベーストポロジーウィジェットを追加できます。ウィジェットで、複数のプロジェクトのレプリケーショントポロジーを1つの画面にまとめて確認できます。

インスタンスモニタリング

ホーム画面 > プロジェクト選択 > **ダッシュボード** > インスタンスモニタリング

インスタンスモニタリングメニューでは、データベースサーバーのコア指標とアクティブセッションに関する詳細情報をリアルタイムでモニタリングし、問題のあるアラートを特定できます。デフォルトでは、過去10分間の収集データをリアルタイムで表示します。収集データは5秒間隔で更新します。

基本画面ガイド



1 検索時間と検索先を選択します。

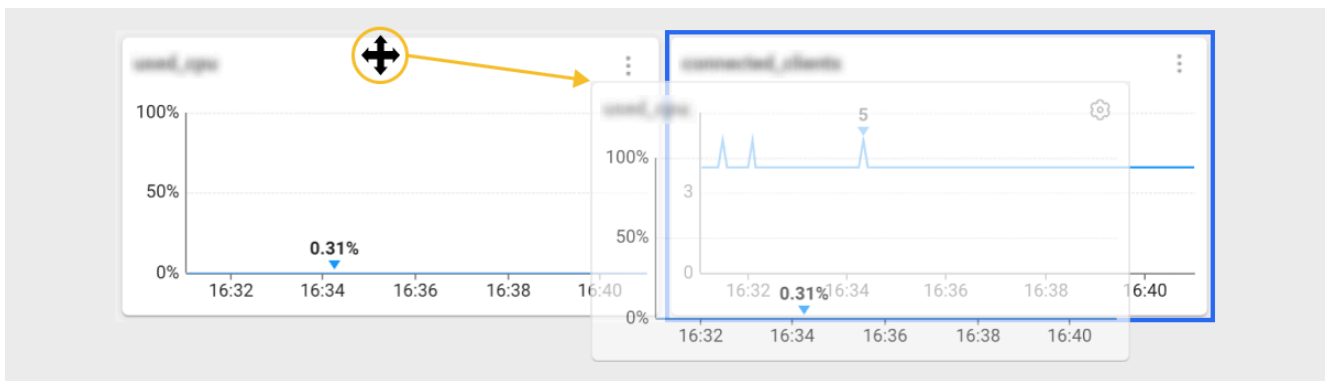
基本的に、直近10分間の収集されたデータをリアルタイムで表示します。時間とインスタンス項目から時間と対象を選択すると、収集データが2 グラフチャートと4 のアクティブセッションテーブルに自動的に反映されます。

- 過去のデータを照会するには、**時間**で緑のボタンをクリックし、希望の照会時間を選択します。過去3週間までのデータを表示できます。
- 任意の日時を照会するには**時間**で **II** ボタンを選択します。日付と時間テキスト領域をクリックして、日付と時間を選択するオプションを表示します。再度、リアルタイムでデータを検索します **▶** ボタンを選択します。
- 他のエージェントのデータを照会するには、**インスタンス**をクリックし、目的のエージェント名を選択します。

2 グラフチャートセクション

3 Default プリセットで保存された16個の指標をグラフチャート形式のウィジェットで表示します。横軸は時間、縦軸は各指標の数値です。各指標の意味を知りたい場合は、名前の横の ⓘ ボタンを選択するか、[次の文書](#)を参照してください。

- 他のウィジェットの位置を変更する場合は、ウィジェットの上部を選択した状態でドラッグします。ただし、ウィジェットのサイズは変更できません。



- 時間の**リアルタイム照会**モードを停止し、ウィジェットでチャートの特定時間をクリックすると、選択した時間に行われたセッション情報を **4 アクティブセッションテーブル**で確認できます。その後、**アクティブセッションテーブル**の上に、5秒単位でデータを照会できるタイムバーが表示されます。

- ⓘ ◦ 選択した時間は、**アクティブセッションテーブル**の右上で確認できます。
- **アクティブセッションテーブル**一覧の文字の色は黒 → オレンジ → 赤の順にセッションの実行速度が遅いことを意味します。

- リアルタイム照会モードを停止した後、ウィジェットの右上の ⓘ ボタンをクリックすると、チャートデータをCSVファイルにダウンロードできます。

4 アクティブセッション / ロックツリー / プロセス情報

リアルタイムで実行中のアクティブセッションおよびロックツリー、プロセス情報を照会できます。ウィジェットのチャートで特定の時間を選択すると、**リアルタイム参照モード**が停止し、選択した時間に実行したセッション情報がテーブルに表示します。テーブルの上には5秒単位でデータを検索できる時間選択ボタンが表示されます。

- **アクティブセッション**：リアルタイムで実行中のセッションとクエリ内容を照会できます。
 - 実行中のセッション項目を選択すると**セッション詳細画面**が表示され、選択したセッションの詳細を確認できます。[次の文書](#)を参考にしてください。
 - テーブルの右側にある**query**カラムの項目を選択すると、**SQL詳細画面**が表示されます。選択したクエリ(Query)のFull TextおよびPlan情報を確認できます。詳細については、[次の文書](#)を参照してください。
- **ロックツリー**

ロックセッションに関する情報やHolderとWaiter情報をリアルタイムで照会できます。テーブルのカラムの詳細については、[次の文書](#)を参照してください。
- **プロセス情報**

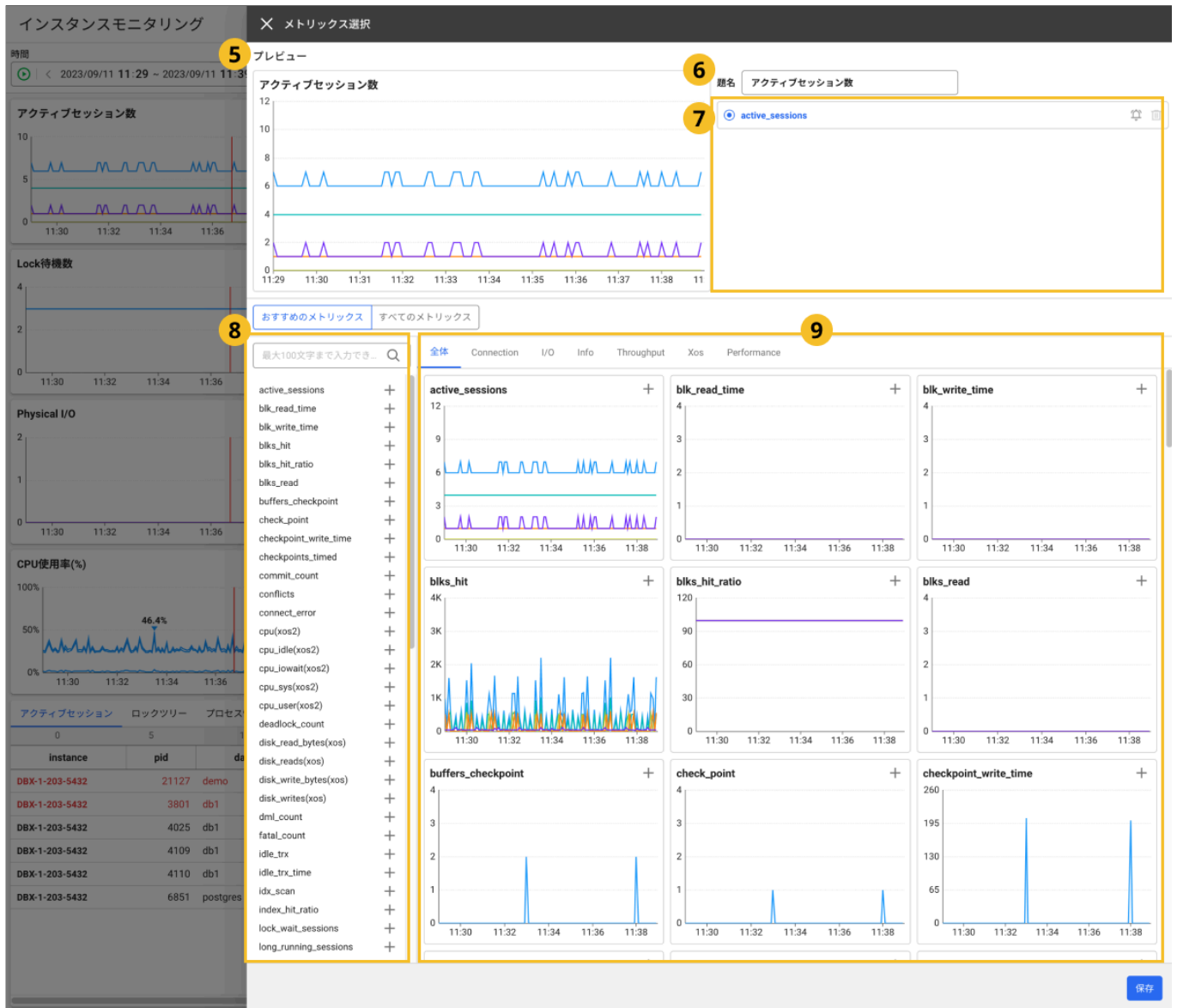
データベースサーバーのリソース使用量をリアルタイムで表示できます。

✔ 連携プロジェクトを追加すると**アクティブセッション**テーブルで、アクティブセッションと連動したアプリケーションアクティブTXの詳細情報をリアルタイムで確認できます。連携プロジェクトの追加と活用についての詳細は、[次の文書](#)を参照してください。

- ⚠
- テーブル一覧の文字の色は黒 → **オレンジ** → **赤**順にセッションの実行速度が遅いことを意味します。
 - **プロセス情報**は、XOSエージェントをインストールしないと確認できません。詳細については、[次の文書](#)を参照してください。

メトリクスを選択する

ウィジェットに指標グラフを追加または置換するには、**⋮** ボタンを選択します。**メトリクス選択画面**が表示されます。設定を完了して**保存**ボタンを選択します。



- 5 プレビュー：選択した指標のグラフのプレビューを確認できます。
- 6 題名：ウィジェット名を入力します。
- 7 現在選択されているウィジェットのチャートで表示されるメトリクス項目が表示されます。最大4つのメトリクスを選択できます。

① メトリクス項目の右側に🔔ボタンを選択すると、そのメトリクスに基づいてアラート通知を設定できるイ

① **ベント設定** *New* メニューに移動できます。

- **⑧ おすすめのメトリクス / すべてのメトリクス**：おすすめメトリクスを選択すると、データベースモニタリングで頻繁に表示される指標のリストが表示されます。一覧に表示されている指標をウィジェットの指標に変更または追加できます。
- **⑨ 領域**で指標の推移をグラフで確認し、ウィジェットの追加または変更することができます。目的のタブを選択して、メトリクスのリストをカテゴリ別に絞り込むことができます。

指標の追加

ウィジェットに2つ以上の指標を追加できます。最大4つまで追加できます。

1. 指標を追加するウィジェットの右上の **⋮** ボタンを選択します。
2. **メトリクス選択** ウィンドウが表示されると、**⑧** メトリクス一覧または **⑨** グラフ一覧で追加する指標の右側にある **+** ボタンを選択します。
3. 指標をさらに追加する場合は、2の手順を繰り返します。
4. ウィジェットを簡単に識別できるように **⑥** 題名を変更します。
5. 画面の右下にある **保存** ボタンを選択します。

指標を追加したウィジェットに追加した指標の数に対してチャートが作成されます。ウィジェット名の横にある **①** ボタンを選択すると、追加した指標の名前と内容を確認できます。また、チャートにカーソルを合わせると、ツールチップから数値を確認できます。

❗ ① ボタンのツールチップ機能は多言語ではサポートされていません。

指標の変更

ウィジェットの指標を別の指標に変更できます。

1. 指標を変更するウィジェットの右上の **⋮** ボタンを選択します。
2. **メトリクス選択** ウィンドウが表示されると、**⑦** 領域で変更する指標を選択します。

3. ⑧ メトリクス一覧または ⑨ グラフ一覧から目的の指標を選択します。
4. ⑦ 領域で選択した指標の名前が変更されたことを確認してください。
5. 画面の右下にある**保存**ボタンを選択します。

指標の削除

ウィジェットに複数の指標を設定されている場合は、不要な指標を削除できます。

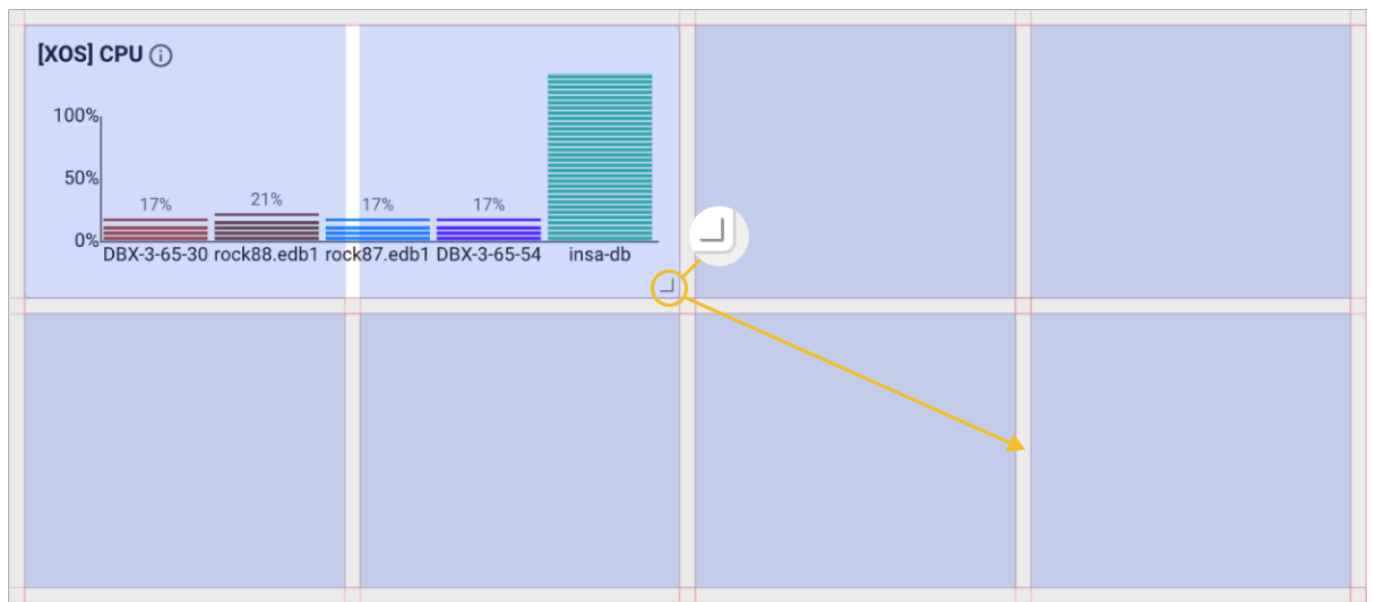
1. 指標を削除するウィジェットの右上の ⋮ ボタンを選択します。
2. **メトリクス選択**ウィンドウが表示されると、⑦ 領域から削除する指標の右側の  ボタンを選択します。
3. 画面の右下にある**保存**ボタンを選択します。

❗ ウィジェットには、少なくとも 1つの指標を選択する必要があります。ウィジェットのすべての指標を削除できません。

ダッシュボードウィジェットの編集

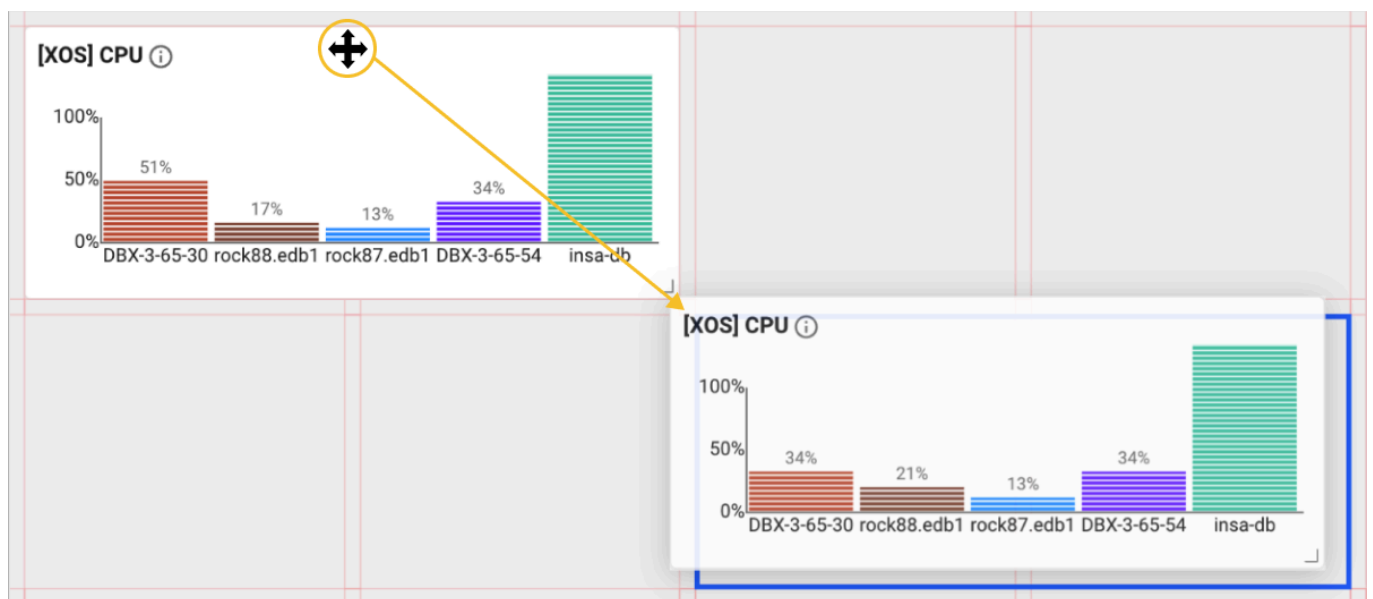
ダッシュボードに配置したウィジェットは、ユーザーが希望するサイズに調整でき、任意の位置に配置できます。るウィジェットは削除したり、再度追加したりすることもできます。

ウィジェットのサイズ変更



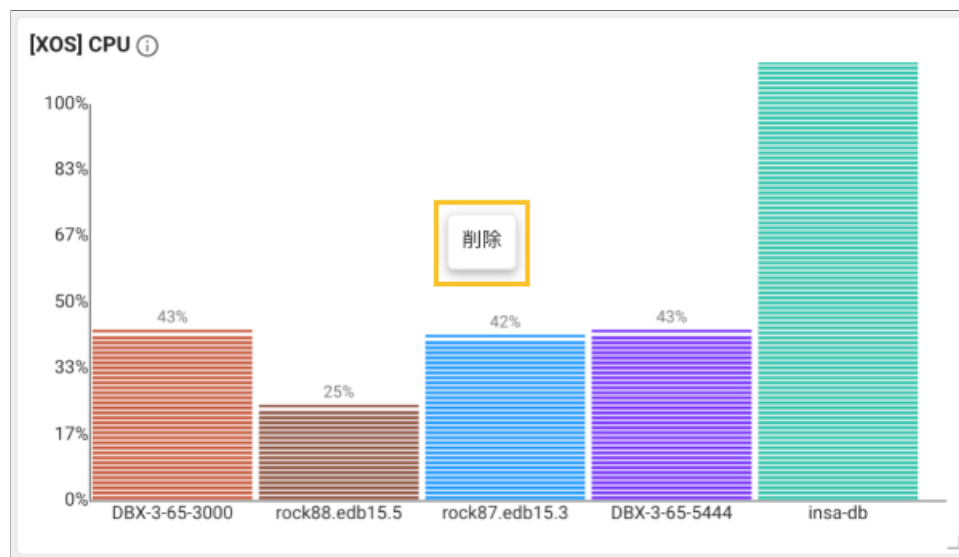
ウィジェットの右下にある「」をマウスでクリックした状態で希望するサイズにドラッグします。幅と高さの比率が均一なグリッドが表示され、グリッド単位でウィジェットのサイズを調整できます。

ウィジェットの移動



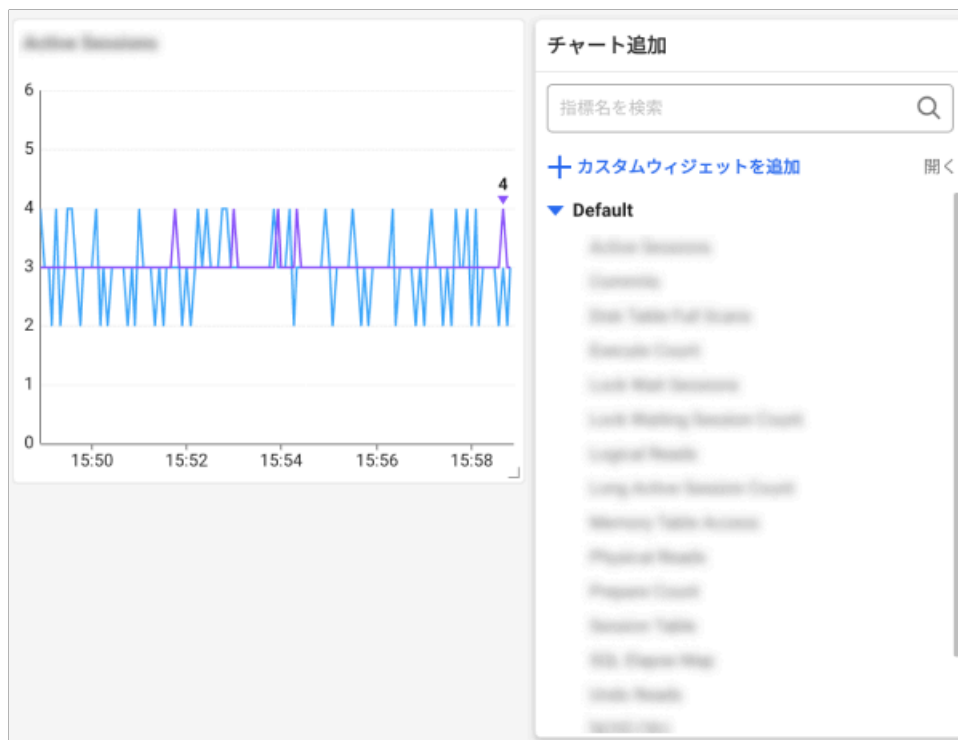
ウィジェットの上部にマウスカーソルを位置すると、カーソルの形が $\oplus$ の形に変更されます。この時、マウスの左ボタンをクリックした状態で希望する位置にドラッグすると、ウィジェットを移動することができます。

ウィジェットを削除する



削除するウィジェットで右クリックします。**削除**ボタンをクリックすると、ウィジェットがダッシュボードから削除されます。

ウィジェットを追加する



ダッシュボードから空白エリアにマウスカーソルを移動し、マウスの右ボタンをクリックします。ポップアップメニューから追加するウィジェットを選択します。希望する場所にウィジェットを配置してサイズを変更します。

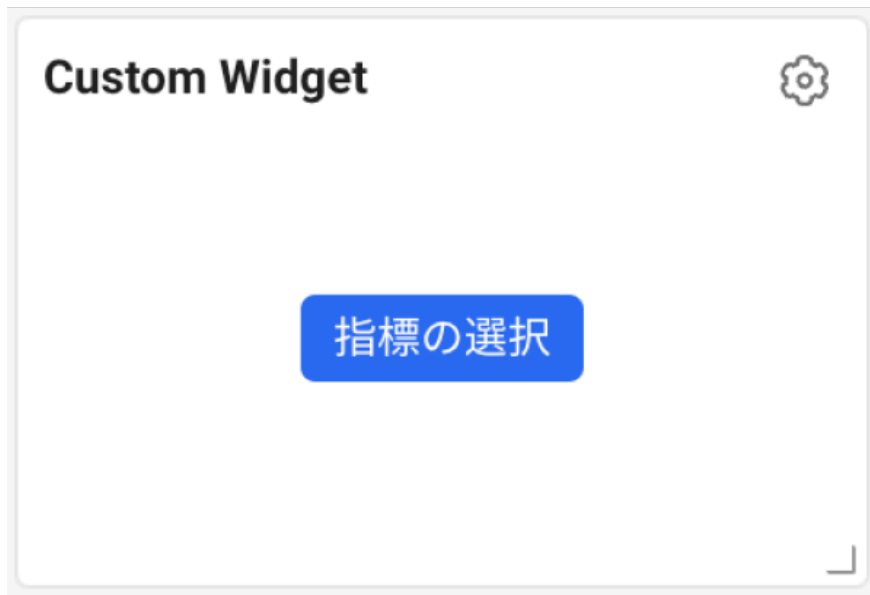
カスタムウィジェット

カスタムウィジェットは、ダッシュボードでデフォルトに提供されるメトリクスの他に、ユーザーがメトリクスを追加できる機能です。デフォルトで提供するメトリクスの他に、ダッシュボードで継続的にモニタリングしたいメトリクスがある場合は、カスタムウィジェット機能を利用できます。

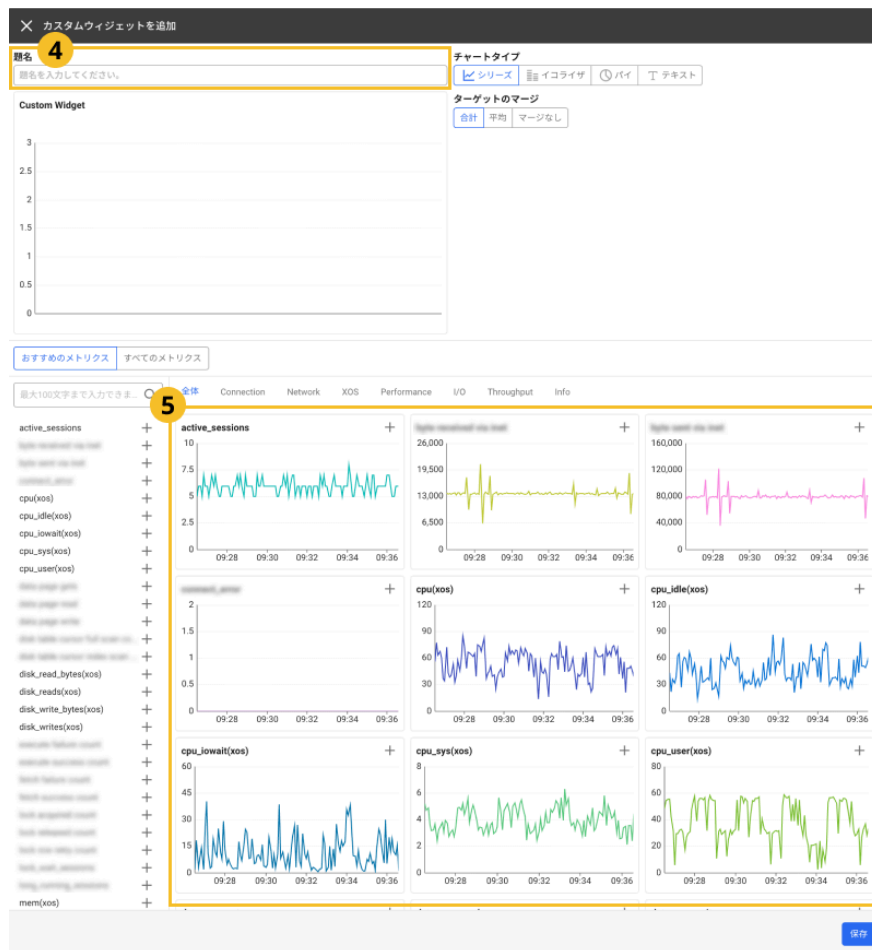
追加する

1. ダッシュボードの空のスペースで、右クリックします。
2. ポップアップメニューで**カスタムウィジェットを追加**を選択します。

3. ダッシュボードに**Custom Widget**が生成されたら、**指標の選択**ボタンを選択します。



4. 画面の右側から **カスタムウィジェット**を追加画面が現れると、**題名項目**にウィジェットの名前を入力します。



5. 画面下のメトリクス一覧から、追加するメトリクスの右上にある＋ボタンを選択します。
6. メトリクスをすべて追加したら**保存**ボタンを選択します。

- ❗ 最大4つのメトリクスを追加できます。メトリクスを追加する時は、＋ボタンを選択します。
- メトリクスを変更する時は、メトリクス一覧で別のメトリクス項目を選択します。＋ボタンを選択すると、そのメトリクスが追加されます。
- ウィジェットのタイトルを入力しないと、Custom Widgetと保存されます。

チャートタイプを選択する

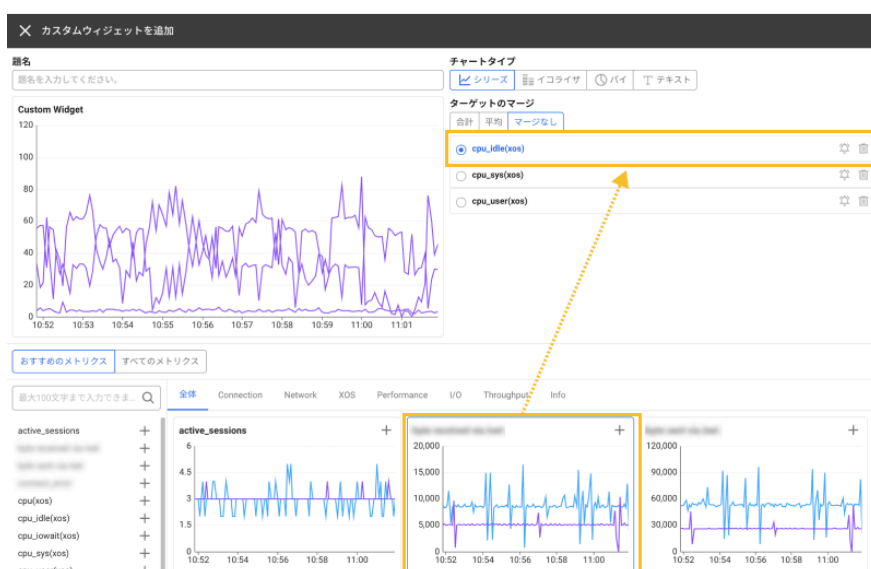
カスタムウィジェットを追加画面のチャートタイプでは、次の形式のチャートを選択できます。

-  **シリーズ**: 時間の経過によるメトリクスの変化を確認できる時系列形式のチャートです。
-  **イコライザ**: 複数のメトリクスの性能を比較できる棒グラフ形式のチャートです。様々なメトリクスを1つの画面に視覚的に表現することで、全体的な性能状態を簡単に把握できます。
-  **パイ**: 複数のメトリクスの全体に対する部門構成比率を確認するのに役立つ円形チャートです。
-  **テキスト**: メトリクスに対する数値をテキスト形式で表示します。

指標の変更

ユーザー定義ウィジェットに追加したメトリクスを、他のメトリクスに変更または追加できます。

1. ダッシュボードに追加したユーザー定義ウィジェットの右上に  ボタンを選択します。
2. 画面の右側で **カスタムウィジェットを追加画面**が表示されたら、変更するメトリクスを選択します。



3. 画面下のメトリクスの一覧で変更するメトリクスを選択します。メトリクスを追加する場合、追加するメトリクスの右上に  ボタンを選択します。
4. メトリクスを全て変更したのであれば **保存** ボタンを選択します。

✓ 追加または変更したメトリクス一覧に表示されたアイコンボタンの機能は、次のとおりです。

-  : 該当メトリクスをカスタムウィジェットから除外できます。
-  : 該当メトリクスに対するアラート通知を設定できます。ボタンを選択すると、**アラート通知 > イベント設定**メニューに移動します。メトリクスイベント設定の詳細については、[次の文書](#)を参考にしてください。

ウィジェットオプション

ウィジェットに表示されているアイコンボタンの機能は次のとおりです。

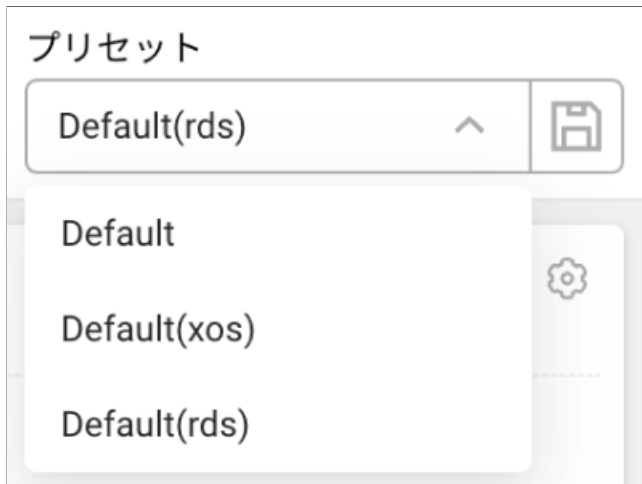
-  : 主要ウィジェットに関する機能と情報が確認できます。(多言語に対応予定)
-  /  : チャートの縦軸範囲を拡大または縮小できます。
-  : 該当ウィジェットのデータを広い画面に開いて表示します。
-  : 該当ウィジェットのデータをエージェントごとに区分して照会できる詳細画面が表示されます。

ⓘ ウィジェットによって使用可能なオプションは異なる場合があります。

プリセット

ユーザーが設定したウィジェットの設定とアクティブセッションテーブルの配置状態をプリセットで保存して読み込むことができます。次のプリセットがデフォルトで提供されており、**Default**のプリセットは変更できません。

- **Default** : 主要DB指標で構成されています。
- **Default(xos)**: 主要なDB指標とXOS指標(CPUおよびメモリ使用率、メモリSWAPの使用率、Disk I/Oなど)で構成されています。
- **Default(rds)**: 主要なDB指標とRDS CloudWatch指標(CPU使用率、メモリ可用性、ストレージ可用性、Networkなど)で構成されています。



プリセット保存

プリセット一覧から保存した項目を選択して、設定を呼び出すことができます。

1. ウィジェットの指標を追加または変更します。
2. 画面の右上に  ボタンを選択します。
3. **プリセット保存**で名前を変更した上で**保存**ボタンを選択します。

- ❗
- **Default**プリセットは、変更できません。
 - データベース環境によって、一部のプリセットはサポートされない場合があります。
 - ユーザー定義プリセットは、変更した指標グラフだけでなく、アクティブセッションのカラム順序、カラム幅、フィルター適用内容も一緒に適用されます。
 - **プロジェクト修正権限**のあるメンバーのみがプリセットを保存または変更できます。メンバー権限の詳細については、[次の文書](#)を参照してください。

プリセットを作成する

1. ダッシュボードから、希望する形にウィジェットを配置してみてください。サイズを変更して、頻繁に確認するウィジェットのみを配置することもできます。
2. 画面の右上に  アイコンをクリックします。

3. 新しいプリセット名を入力します。
4. **保存**ボタンをクリックします。プリセット一覧から新しく保存したプリセットを確認できます。

- ①
- ・新しく作成したプリセットに変更が加えられた場合は、再度プリセットを保存してください。📁 アイコンをクリックし、同じ名前でプリセットを保存します。既存のプリセットに変更を上書きします。
 - ・ダッシュボードの変更を保存せずに別のメニューに移動すると、変更は保存されません。
 - ・プリセットはプロジェクト単位で保存され、他のユーザーと共有できます。

プリセット削除

画面の右上ある**プリセット**を選択すると、プリセット一覧が表示されます。削除するプリセット項目の右側にある **×** ボタンを選択します。

アクティブセッションのセクションを活用

画面の下にある**アクティブセッションセクション**からリアルタイムで実行中のセッション情報を確認できます。ここでは、テーブルの右上にあるボタン機能を説明します。

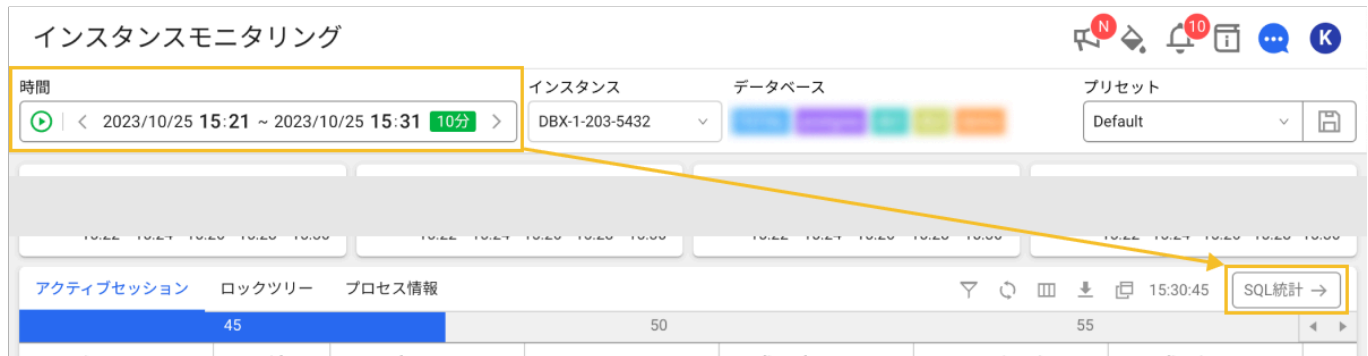
- ・⌵：テーブルのカラムヘッダー項目に基づいて一覧をフィルタリングできます。
- ・🔄：テーブルのリストを更新できます。
- ・☐：テーブル内のカラムヘッダー項目を表示または非表示にすることができます。
- ・⏸：稼働中のセッションを停止できます。

- ① この機能は、**修正権限**のあるメンバーのみ利用できます。メンバー権限に関する詳細な説明は[次の文書](#)を参照してください。

- ・⬇️：テーブルの内容をCSVファイルとしてダウンロードできます。
- ・🖥️：別ウィンドウで開き、より大きな画面でリストを確認できます。

SQL統計を確認する

過去の特定期間の間に発生した主要性能指標の推移とともに、その間のTop SQL一覧が照会できます。



1. 時間から || ボタンを選択して、リアルタイムモードから非リアルタイムモードに切り替えてください。
2. 時間で照会したい時間帯を指定してください。
3. アクティブセッション領域の右上のSQL統計ボタンを選択します。

新しい画面とともにSQL統計メニューへ移動します。インスタンスメニューで設定した時間の間に実行されたTop SQL一覧を確認することができます。SQL統計詳細については、[次の文書](#)を参照してください。

セッション詳細情報を確認する

アクティブセッションタブでセッション項目を選択するとセッション詳細画面が表示されます。選択したセッションについての詳細を確認できます。

The screenshot displays the PostgreSQL instance monitoring interface. At the top, a table lists active sessions. The selected session (pid: 10087) is highlighted, and its details are shown below. The session is in an 'active' state, running a query that extracts data from the 'pg_stat_activity' table. The interface includes a 'Kill Session' button and a 'Session History' link. The bottom section shows the SQL text of the query and various process statistics.

instance	pid	datname	cpuusage	client_addr	username	runtime ↓	state	wait_event	query	query_param	application_n
DBX-1-203-5432	10087	postgres	0.33%	192.168.1.204	whatap	0	active		SELECT /* WhaTap2D#4 */ datname,pid,extrac... 'idle'		

セッション詳細 (pid: 10087)

|| LIVE 08:34:18 [Refresh] [Kill Session] セッション履歴 >

Session Info				Wait Info	
pid	10087	query_hash	348448832	wait_event	
datname	postgres	username	whatap	wait_event_type	
query_start	2024-03-11 08:34:15	application_name	PostgreSQL JDBC Driver		
runtime	0	backend_start	2024-03-11 05:31:50		
state	active	backend_type	client backend		
backend_xmin	2093749	xact_start	2024-03-11 08:34:15		
client_addr	192.168.1.204	state_change	2024-03-11 08:34:15		
client_hostname	192.168.1.204	client_port	52804		
query_param	'idle',26353				

SQL Text Plan

Default Bind [Reset Bind] [Formatting]

```

SELECT
/* WhaTap2D#4 */
datname,
pid,
extract(
epoch
from
current_timestamp - query_start
) as runtime,
extract(

```

Process Info	
cpuetime	14
cpuusage	0.00%
elapsed	10948
vsize	344528
rss	45188
state	active
ioread	0
iowrite	0

- セッションに関する情報は、既定値でリアルタイムで照会されています。タイムセレクターの || ボタンを選択すると、リアルタイム照会を中止し、中止した時間ベースのデータを確認できます。リアルタイム照会を中止した状態でデータを更新するには、🔄 ボタンを選択します。
- Kill Session**：選択したセッションを停止できます。セッション中止画面が表示されたらパスワード(Parameter Key)を入力し、適用ボタンを選択します。

① この機能は、リアルタイム照会モードでのみ利用できます。

- セッション履歴**：選択したセッション履歴が確認できるセッション履歴メニューに移動できます。

① セッション履歴メニューの詳細については、次の文書を参照してください。

- **Session Info**: 選択したセッションの基本情報とステータスを含む詳細を確認できます。
- **Wait Info**: 選択したセッションの待機イベントと待ち時間の情報を提供します。特定の作業を実行するために、データベースシステムで待機する時に発生する待機イベントに関する情報も確認できます。
- **SQL Text / Plan**: 選択したセッションのSQLクエリ文とPlan情報を確認できます。詳細については[次の文書](#)を参照してください。
- **Process Info**: 選択したセッションのプロセス情報を確認できます。

SQL 詳細情報の確認

実行中セッションのSQLクエリ情報を確認するには、**アクティブセッション**セクションの**query**(または**sql_text**) カラム項目を選択します。SQL詳細画面が表示されます。SQLクエリ文とPlan情報を確認することができます。

SQL統計の表示 → ボタンをクリックすると、選択した SQL クエリに関連する統計情報を確認できる **SQL統計** に移動します。

The screenshot shows the SQL query editor and execution plan view. The top panel displays the query text: `START_REPLICATION ...` and `SELECT /* ...`. An arrow points from the query text to the **SQL統計の表示** button in the top bar. The main panel displays the query text on the left and the execution plan on the right. The execution plan shows a series of nested loops and scans, with a table of rows and node costs.

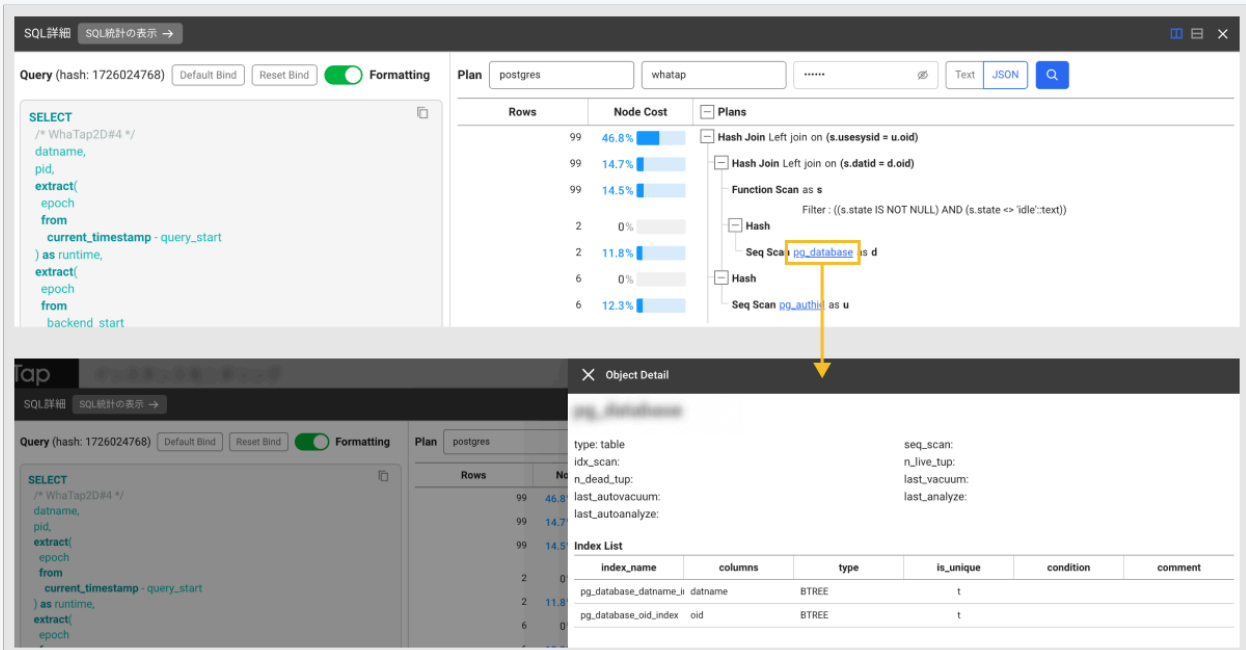
Rows	Node Cost	Plans
4	0%	Append
1	13.9%	Aggregate
606149825272	86.1%	Nested Loop Inner join
579686	0%	Seq Scan on boarding_passes as b
1045652	0%	Materialize
1045652	0%	Seq Scan on ticket_flights as a
3	0%	Nested Loop on ((tf.ticket_no = bp.ticket_no) AND (tf.flight_id = bp.flight_id)) Left join
3	0%	Nested Loop Inner join
3	0%	Nested Loop Inner join
3	0%	Nested Loop Inner join
3	0%	Index Only Scan on ticket_flights as tf on (tf.ticket_no = '0005432661915':bpchar)
1	0%	Index Scan on flights as f on (f.flight_id = tf.flight_id)
1	0%	Index Scan on airports_data as ml on (ml.airport_code = f.departure_airport)
1	0%	Index Scan on airports_data as ml_1 on (ml_1.airport_code = f.arrival_airport)
3	0%	Materialize
3	0%	Bitmap Heap Scan on boarding_passes as bp
3	0%	Bitmap Index Scan on (bp.ticket_no = '0005432661915':bpchar)

- **Default Bind**: クエリ文で変数として設定された部分を既定値としてバインディングできます。

- **Reset Bind**：バインディングされた既定値を解除し、変数を確認できます。
- **Formatting**：SQLクエリ文にインデントおよびフォーマットを適用して可読性を高めます。
- **Plan**：Plan情報を確認するには**DB Name**、**User Name**、**Password**を入力してから、 ボタンを選択します。
 - **Text**：テキスト形式でプラン情報を確認できます。
 - **JSON**：JSON形式でプラン情報を照会すると、ボトルネックが発生した箇所がすばやく確認できます。

❗ **JSON形式でプラン情報を照会する機能は、DBXエージェント1.6.12バージョンから対応しています。**

- ✔ **JSON形式でプラン情報を照会した状態でtableまたはindex項目を選択すると、テーブルのcolumn、index構成が確認できるObject detailウィンドウが表示されます。**



The screenshot shows the DBX Agent interface. The top window is the 'SQL Plan' window, which displays a query plan for a SELECT statement. The query is: `SELECT /* WhaTap2D#4 */ datname, pid, extract(epoch from current_timestamp - query_start) as runtime, extract(epoch from backend_start`. The plan shows a Hash Join Left join on (s.usersysid = u.oid) and a Hash Join Left join on (s.datid = d.oid). The bottom window is the 'Object Detail' window, which shows details for the 'pg_database' table. The table is of type 'table' and has columns 'datname' and 'oid'. The 'Index List' section shows two indexes: 'pg_database_datname_i' and 'pg_database_oid_index'.

この機能は、DBXエージェント1.6.15バージョン以降に対応しています。データベース権限に関する設定は、[次の文書](#)を参考にしてください。

AI チューニングガイド

AI チューニングガイドは、SQL クエリ、プラン、統計情報を分析してパフォーマンス問題を診断し、最適化の提案を

行う機能です。

開発者や DBA がボトルネックの原因を迅速に特定し、効率的な SQL によりパフォーマンスを改善できるよう支援します。

⚠ 使用条件および注意事項

PostgreSQL、MySQL、SQL Server ではプランの取得が必須です。

プランを取得しない場合、AI チューニングガイドアイコンは （無効）として表示され、機能を使用できません。

- AI が生成する結果は自動分析に基づいており、精度が 100% ではないことにご留意ください。

1. 分析および診断する SQL をクリックし、**SQL詳細** 画面に移動します。
2. **SQL詳細** 画面の右下にある AI チューニングガイドアイコン  をクリックして AI 分析を開始します。
3. AI 分析結果を確認します。

結果項目	説明
クエリプランおよび概要	クエリの目的と実行概要を表示します。 実行回数、累積実行時間、データベース全体の負荷比率を分析し、当該 SQL がシステムパフォーマンスに与える影響を評価します。
パフォーマンス分析	分析結果を総合し、パフォーマンススコアと診断結果を提供します。 CPU 使用率、ディスク使用率、キャッシュヒット率、待機時間など、クエリ実行過程における詳細なリソース使用量を分析し、ボトルネックの発生区間を視覚的に表示します。
発見された主な問題	主な問題点を要約して表示します。
最適化推奨事項	検出された問題に基づき、最適化されたクエリを提案します。

デフォルトプランを照会する

デフォルトプランユーザーファイル作成時、エージェントが事前定義されたDB情報を使用してプランを自動的に照会するように設定できます。 **デフォルトプランの照会**項目のチェックボックスを選択した後、次のように予め入力した

DB接続情報を通じてプラン情報を照会します。

• Linux Windows

1. エージェント設定(`whatap.conf`)ファイルに `plan_db` オプションを追加し、DB名を入力します。

`whatap.conf`

```
plan_db=(dbname)
```

2. DB設定後 `planuid.sh` スクリプトを使用してユーザーとパスワードを設定します。

```
./planuid.sh user pwd
```

❗ ユーザー(`user`)とパスワード(`pwd`)に特殊文字が含まれている場合、各特殊文字の前にバックスラッシュ(`\`)を追加します。

例、 `user$example` → `user\$example`

1. エージェント設定(`whatap.conf`)ファイルに `plan_db` オプションを追加し、DB名を入力します。

`whatap.conf`

```
plan_db=(dbname)
```

2. DB設定後 `planuid.bat` スクリプトを使用してユーザーとパスワードを設定します。

```
planuid.bat user pwd
```

連携プロジェクトを確認する

連携プロジェクト管理メニューで別のアプリケーションプロジェクトを追加すると、アクティブセッション項目に  ボタンが追加されます。



🔗 ボタンを選択すると表示される**連携分析**画面を使用すると、ユーザーはアクティブセッションと連動したアプリケーションのアクティブTX情報をリアルタイムで確認できます。トランザクションURLおよびデータベース接続情報、DBタイプ、HTTPメソッド、実行時間などの基本情報とともに、トランザクションID、スレッドID、クライアントIP、CPU使用時間などの具体的なメトリクスを確認できます。また、SQLクエリの実行時間、呼び出されたSQLの数、DB接続時間などもモニタリングでき、システムの性能を総合的に分析できます。

特に、スタックトレースによるコールスタック(Call stack) 情報とSQLクエリを一緒に提供し、問題発生時の詳細な原因を追跡して解決するのに役立ちます。これにより、アプリケーションとデータベース間の相互作用を深く分析し、性能のボトルネック現象を迅速に診断できます。

❗ 連携プロジェクトの追加と活用についての詳細は、[次の文書](#)を参照してください。

進行中のセッションの停止

1. テーブルの右上にある 🔄 ボタンを選択します。
2. テーブル一覧から停止するセッションを選択します。複数選択できます。
3. 停止ボタンを選択します。
4. **セッション中止**画面が表示されると、パスワード(**Parameter Key**)を入力します。パスワードは、エージェントのインストールディレクトリにある [paramkey.txt] で確認できます。
5. **適用**ボタンを選択します。選択したセッションが停止します。

❗ • DBXエージェントのインストールパスの **paramkey.txt** ファイルからパスワードを確認できます。

- ❗ この機能は、**修正権限**のあるメンバーのみ利用できます。メンバー権限の詳細については、[次の文書](#)を参照してください

グラフチャートセクションを拡大する

現在見ている画面が小さくてウィジェットを確認しづらい場合は、より広い画面でグラフチャートウィジェットを確認することができます。画面下段に**アクティブセッション**セクションの▼ボタンを選択します。**アクティブセッション**セクションは縮小され、グラフチャートウィジェットセクションは拡大します。

グラフチャートウィジェットセクションを縮小するには、画面下部に**アクティブセッション**セクションの▲ボタンを選択します。

テーブルカラムの設定

テーブルヘッダーカラムを非表示にしたり、希望する項目を追加できます。カラムの順序を変更することもできます。☐ボタンをクリックします。

- ❗ 設定を完了した後は、**確認**ボタンを選択すると、設定事項がテーブルに反映されます。
- ③ 検索バーにテキストを入力し、カラム項目を検索できます。入力したテキストと一致するカラム項目が表示されます。
- 画像は、製品またはプロジェクト、メニューによって異なる場合があります。

カラムを追加する

① 一覧でテーブルヘッダーカラムに追加する項目を選択します。すべての項目を追加するには、**すべて選択**を選択します。

カラムを削除する

① 一覧から削除するカラム項目のチェックボックスのチェックを解除します。または② 一覧から削除する項目の右側の×ボタンを選択します。

カラムの順序の変更

2 一覧で順序を変更するアイテムをドラッグして、希望する位置に移動できます。

設定事項の初期化

変更事項をすべてキャンセルし、初期化するには、○ 初期化ボタンを選択します。

テーブルデータのフィルタリング



1. テーブルの右上にある▽ボタンを選択します。
2. テーブルのカラムヘッダー項目と条件を選択します。
3. 条件を入力してください。テキストボックスに条件を入力します。
4. 保存ボタンを選択します。

カラム情報ガイド

- [Active session](#) [Lock tree](#) [Process information](#)

項目	説明
db	データベース名
user	SQL実行アカウント

項目	説明
<code>application</code>	クライアントアプリケーション名
<code>client_host</code>	接続されたクライアントのホスト名(<code>client_addr</code> の逆引きDNS参照で報告されます)
<code>client_addr</code>	クライアントのIPアドレス
<code>client_port</code>	クライアントのポート番号(Unixソケットを使用する場合は-1)
<code>query_hash</code>	クエリのハッシュ値(WhaTap内部管理値)
<code>query_start</code>	現在アクティブ状態のクエリが開始された時間(アクティブ状態でない場合、最後にクエリが開始された時間)
<code>state</code>	バックエンドプロセスの現在の状態(active/idleなど)
<code>state_change</code>	stateが変更された最近の時間
<code>xact_start</code>	そのプロセスの現在のトランザクションが開始された時間(アクティブTXがない場合は <code>null</code>)
<code>backend_start</code>	そのバックエンドプロセスが開始された時間
<code>backend_xmin</code>	バックエンドプロセスが処理中の <code>xid</code> の最小値
<code>backend_type</code>	バックエンドプロセスのタイプ、 <code>autovacuum worker</code> 、 <code>logical replication launcher</code> 、 <code>client backend</code> 、 <code>checkpointer</code> など
<code>backend_xid</code>	バックエンドが使用中の <code>xid</code>
<code>wait_event_type</code>	バックエンドプロセスが待機中のイベント <code>type</code>
<code>wait_event</code>	バックエンドプロセスが現在待機中の場合は、待機イベント名

ⓘ WhaTapは、クライアントに関する情報をデフォルトで保存します。

項目	説明
Instance	インスタンス（エージェント）名
pid	セッションID
lock_mode	ホルダーセッションのロックモード - RowExclusiveLock：INSERT、UPDATE、DELETE - AccessExclusiveLock：DDL（ALTER、DROP、TRUNCATE など）によるロック - AccessShareLock：SELECT 時に使用 ほとんどすべてのロックと共存可能 - ShareLock：LOCK TABLE ... IN SHARE MODE
holder_relation	ロックを保有しているセッションがアクセス中のテーブル名またはオブジェクト識別子
waiter_type	ロックがどの対象に対してかかっているかを示します - relation：テーブルにかかっているロック tuple：特定の行（row）にかかっているロック - transactionid：トランザクションにかかっているロック
waiter_relation	ロックを待機しているセッションがアクセス中のテーブル名またはオブジェクト識別子
lock_request	ロックを待機しているセッションのロックモード lock_mode を参照

次の項目は、XOSエージェントをインストールした場合に収集される指標です。

項目	説明	単位
pid	プロセスID	-
cpuusage	プロセスが使用するCPU使用率	%
cputime	プロセスがこれまでに使用した累積CPU時間	秒
elapse	プロセスが実行されてからの経過時間	秒
vsize	Virtual memory size. 全体の仮想メモリ使用量（RAM + Swap + 予約）	KB

項目	説明	単位
rss	Resident Set Size. 現在のプロセスが占有している実際の物理メモリサイズ	KB
pss	Proportional Set Size. 共有メモリを含む実際の物理メモリ使用量	KB
state	プロセスの現在の状態コード（例：R、S、D、T、Z など）	-
ioread	累積読み込みバイト数（read_bytes）	バイト（B）
iowrite	累積書き込みバイト数（write_bytes）	バイト（B）
uid	このプロセスを実行したユーザーID	-
cmd	実行されたコマンドの短縮形式	-
longcmd	実行されたコマンドの完全な形式（フルパスおよび引数を含む）	-

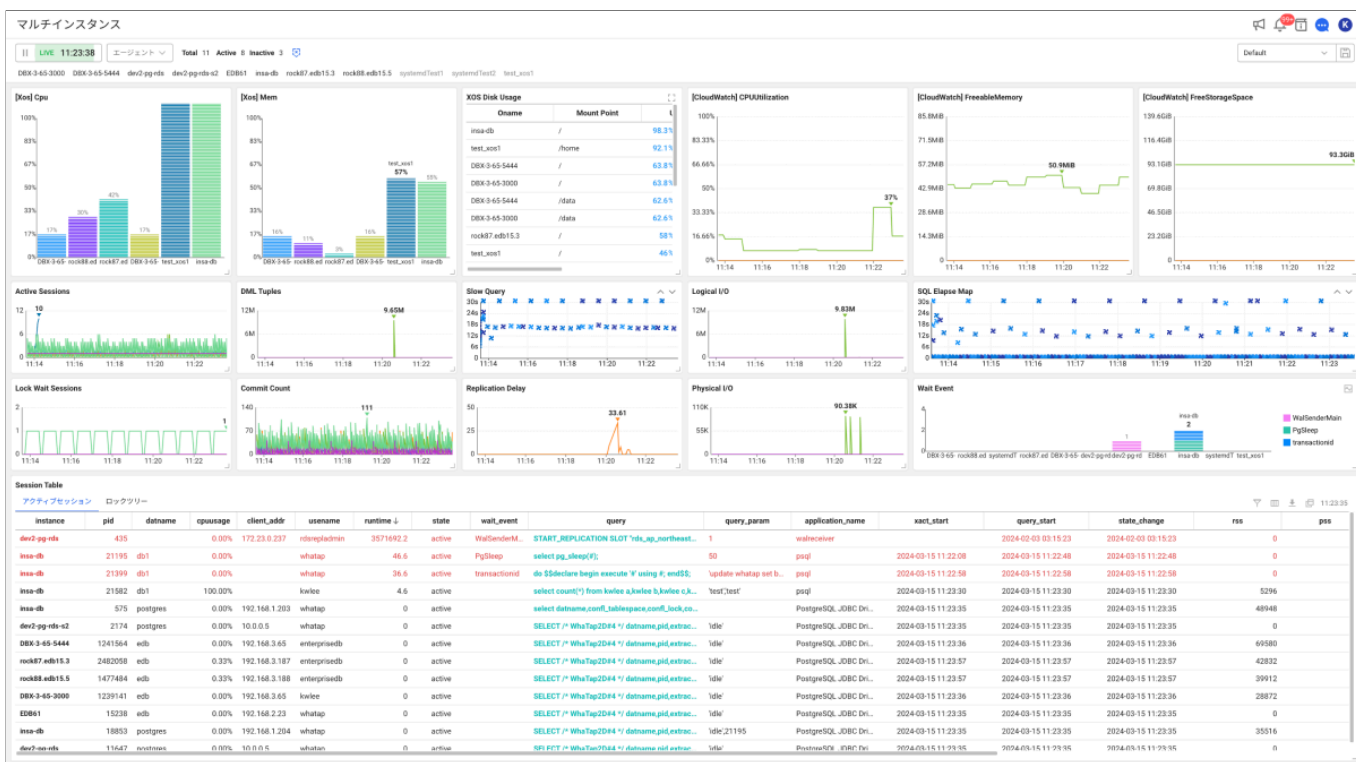
マルチインスタンス

ホーム画面 > プロジェクト選択 > ダッシュボード > マルチインスタンス

インスタンスモニタリングメニューでは、1つのエージェント(インスタンス)だけをモニタリングすると、マルチインスタンスは、複数のエージェント(インスタンス)を同時に監視して管理できる機能を提供します。

各インスタンスで発生する主な性能指標を、一目で比較して分析できます。実行中の各エージェントの数、すべてのアクティブセッションの合計、トランザクション数などのデータを集計して、効果的にモニタリングできます。ユーザーは、グラフチャートを通じて時間によるメトリクス値の変化と各インスタンスの高い性能メトリクスを簡単に確認できます。

また、最も長く続くアクティブセッションに関する情報を確認でき、ユーザーはシステム性能のボトルネック現象を診断し、解決できる根拠資料として活用できます。



✅ **ダッシュボード > インスタンスリストメニューで比較しようとするエージェントを選択しマルチビューボタンを選択するとマルチインスタンスメニューに入ります。**

照会時間設定

ダッシュボードではリアルタイムモニタリング機能を基本的に提供しますが、画面上部のタイムセクターを利用して過去時間のデータを照会できます。タイムセクターでIIボタンを選択して時間を設定します。ユーザーが設定した時間に基づいて、ダッシュボードに配置したウィジェットのデータを更新します。

エージェント確認

エージェント接続状態を確認する



画面の左上のタイムピッカーの右側には、プロジェクトに関連したエージェントの状態を確認できる情報が表示されます。この機能で、モニタリング対象サーバーの動作有無がすぐに確認できます。

- **Total**: プロジェクトに関連付けられているすべてのエージェントの数
- **Active**: 活性化されたエージェントの数
- **Inactive**: 非活性化状態のエージェント数
- : 非活性化されたエージェントを表示または非表示します。

エージェント別モニタリング



基本的に、ダッシュボードにはすべてのエージェントから収集した指標をチャートとして表示しますが、エージェントを選択してデータを表示することもできます。タイムピッカーの下にあるエージェントを1つ以上選択します。選択したエージェントの指標でダッシュボードに配置されたウィジェットのデータを更新します。

- ✔ エージェントを1つ以上選択した状態で再度すべてのエージェントを選択するには、選択を解除するか、[Total](#)をクリックします。

- ⓘ プロジェクトに接続されたエージェントの数が多い場合は、エージェント名を短く設定することをお勧めします。エージェント名の設定に関する詳細は、[次の文書](#)を参照してください。

ウィジェット編集

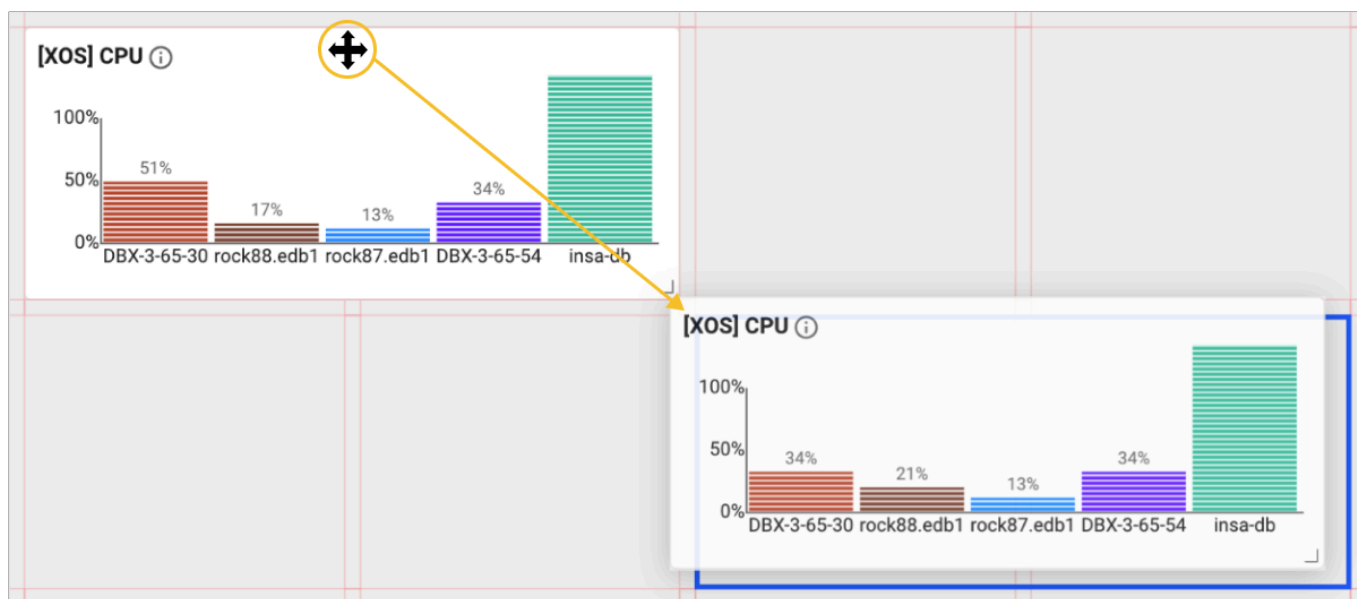
ダッシュボードに配置したウィジェットは、ユーザーが希望するサイズに調整でき、任意の位置に配置できます。ウィジェットは削除したり、再度追加したりすることもできます。

ウィジェットのサイズ変更



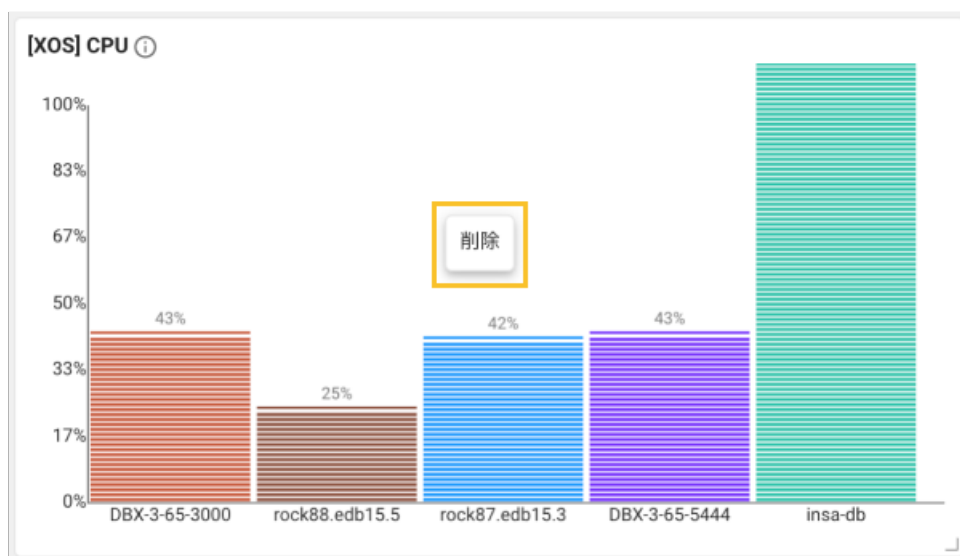
ウィジェットの右下にある「」をマウスでクリックした状態で希望するサイズにドラッグします。幅と高さの比率が均一なグリッドが表示され、グリッド単位でウィジェットのサイズを調整できます。

ウィジェットの移動



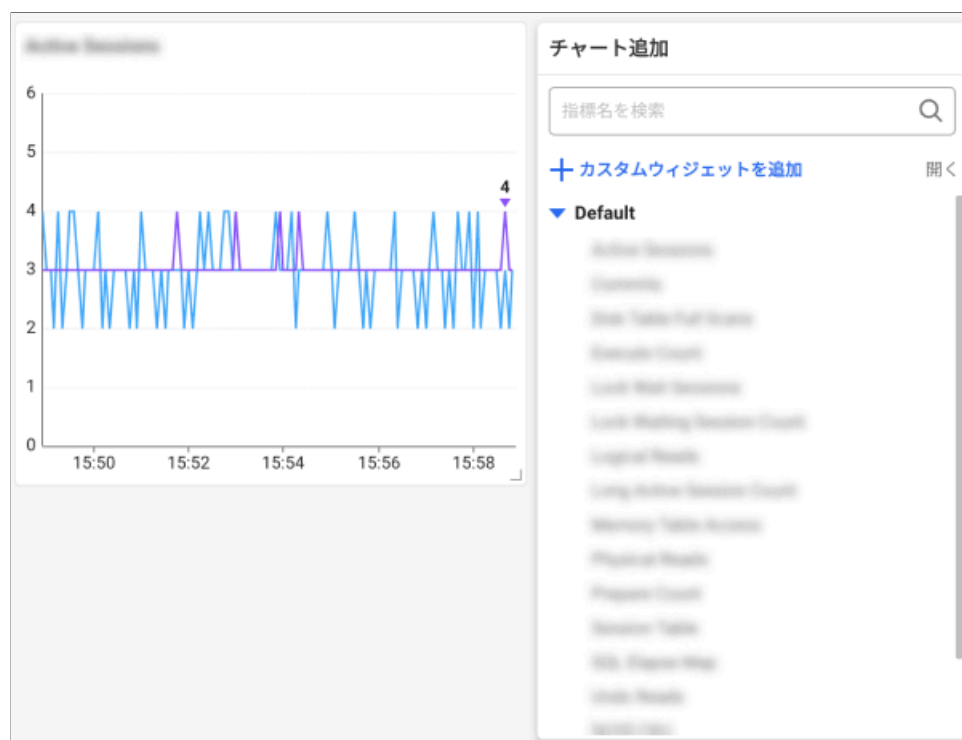
ウィジェットの上部にマウスカーソルを位置すると、カーソルの形が $\oplus$ の形に変更されます。この時、マウスの左ボタンをクリックした状態で希望する位置にドラッグすると、ウィジェットを移動することができます。

ウィジェットを削除する



削除するウィジェットで右クリックします。削除ボタンをクリックすると、ウィジェットがダッシュボードから削除されます。

ウィジェットを追加する



ダッシュボードから空白エリアにマウスカーソルを移動し、マウスの右ボタンをクリックします。ポップアップメニューから追加するウィジェットを選択します。希望する場所にウィジェットを配置してサイズを変更します。

- ① ダッシュボードに配置できるウィジェットの詳細については、[次の文書](#)を参考にしてください。
- ダッシュボードで基本的に提供されるメトリクスの他に、ユーザーが希望するメトリクスを追加できます。ユーザー定義ウィジェットの詳細については、[次の文書](#)を参照してください。

カスタムウィジェット

カスタムウィジェットは、ダッシュボードでデフォルトに提供されるメトリクスの他に、ユーザーがメトリクスを追加

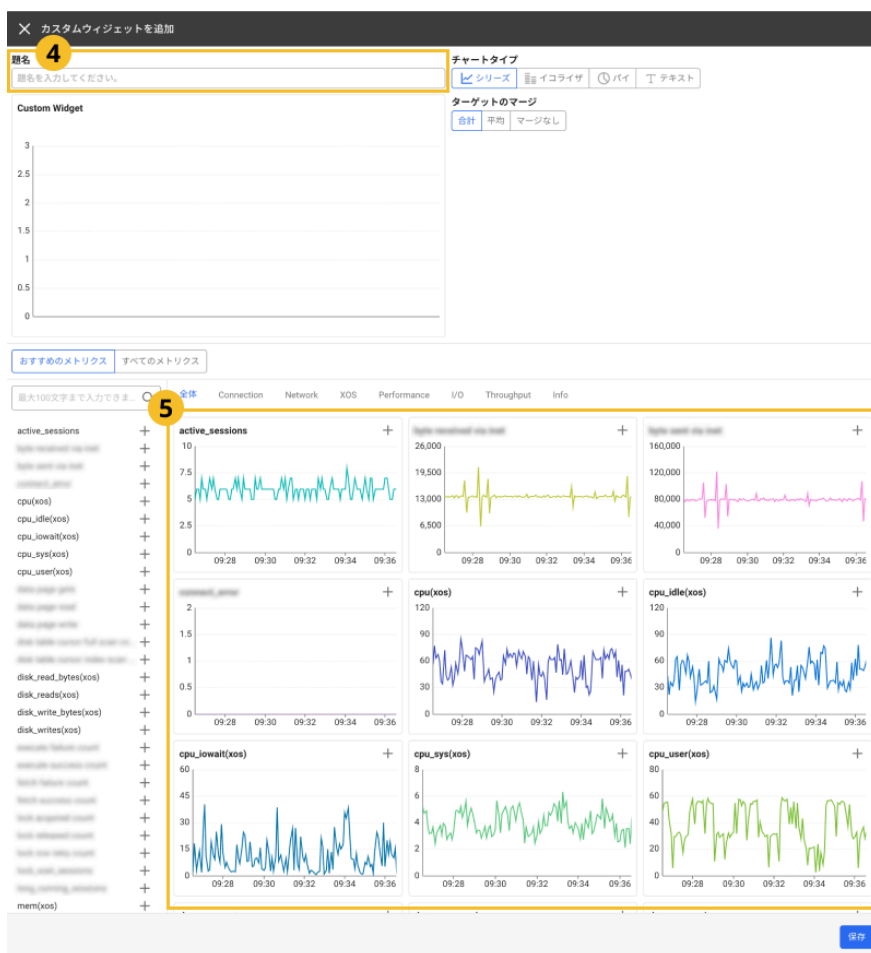
できる機能です。デフォルトで提供するメトリクスの他に、ダッシュボードで継続的にモニタリングしたいメトリクスがある場合は、カスタムウィジェット機能を利用できます。

追加する

1. ダッシュボードの空のスペースで、右クリックします。
2. ポップアップメニューで**カスタムウィジェットを追加**を選択します。
3. ダッシュボードに**Custom Widget**が生成されたら、**指標の選択**ボタンを選択します。



4. 画面の右側から **カスタムウィジェットを追加**画面が現れると、**題名項目**にウィジェットの名前を入力します。



5. 画面下のメトリクス一覧から、追加するメトリクスの右上にある＋ボタンを選択します。

6. メトリクスをすべて追加したら**保存**ボタンを選択します。

- ① 最大4つのメトリクスを追加できます。メトリクスを追加する時は、＋ボタンを選択します。
- メトリクスを変更する時は、メトリクス一覧で別のメトリクス項目を選択します。＋ボタンを選択すると、そのメトリクスが追加されます。
- ウィジェットのタイトルを入力しないと、Custom Widgetと保存されます。

チャートタイプを選択する

カスタムウィジェットを追加画面のチャートタイプでは、次の形式のチャートを選択できます。

-  **シリーズ**: 時間の経過によるメトリクスの変化を確認できる時系列形式のチャートです。
-  **イコライザ**: 複数のメトリクスの性能を比較できる棒グラフ形式のチャートです。様々なメトリクスを1つの画面に視覚的に表現することで、全体的な性能状態を簡単に把握できます。
-  **パイ**: 複数のメトリクスの全体に対する部門構成比率を確認するのに役立つ円形チャートです。
-  **テキスト**: メトリクスに対する数値をテキスト形式で表示します。

データマージ方式を選択する

カスタムウィジェットを追加画面の対象のマージでは、エージェントから収集されたメトリクスを表示する方法を選択できます。

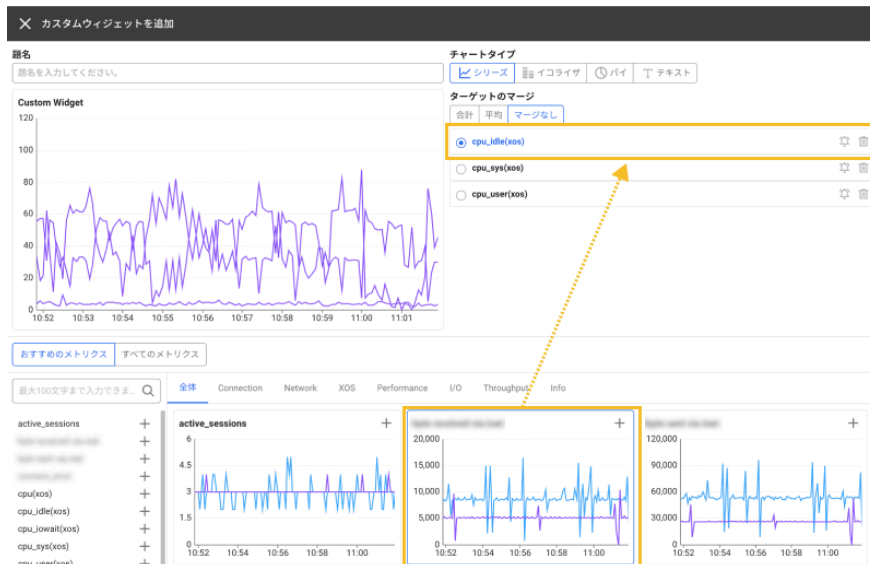
- **合計**: 複数のエージェントから収集されたメトリクス値を合計して、1つの値として表示します。このオプションは、複数のエージェントのデータを統合して全体的な合計を表示する場合に役立ちます。
- **平均**: 複数のエージェントから収集されたメトリクス値を平均して、1つの値として表示します。このオプションは、エージェント全体の平均性能や状態を把握するのに役立ちます。
- **最大値**: 複数のエージェントから収集された指標値の中から最大の値を選択して、一つの値として表示します。このオプションは、エージェントの中で最もパフォーマンスが消耗した状況や最悪の状態を把握するのに役立ちます。
- **マージなし**: 各エージェントのメトリクスを個別に表示します。

❗ チャートのタイプによって選択できるマージ方式は異なる場合があります。

指標の変更

ユーザー定義ウィジェットに追加したメトリクスを、他のメトリクスに変更または追加できます。

1. ダッシュボードに追加したユーザー定義ウィジェットの右上に⚙️ボタンを選択します。
2. 画面の右側で**カスタムウィジェットを追加画面**が表示されたら、変更するメトリクスを選択します。



- 画面下のメトリクスの一覧で変更するメトリクスを選択します。メトリクスを追加する場合、追加するメトリクスの右上に+ ボタンを選択します。
- メトリクスを全て変更したのであれば保存ボタンを選択します。

✔ 追加または変更したメトリクス一覧に表示されたアイコンボタンの機能は、次のとおりです。

- 🗑️: 該当メトリクスをカスタムウィジェットから除外できます。
- 🔔: 該当メトリクスに対するアラート通知を設定できます。ボタンを選択すると、**アラート通知 > イベント設定**メニューに移動します。メトリクスイベント設定の詳細については、[次の文書](#)を参考にしてください。

ウィジェットオプション

ウィジェットに表示されているアイコンボタンの機能は次のとおりです。

- ℹ️: 主要ウィジェットに関する機能と情報が確認できます。(多言語に対応予定)
- ⤴️ / ⤵️: チャートの縦軸範囲を拡大または縮小できます。
- 📄: 該当ウィジェットのデータを広い画面に開いて表示します。
- 🔍: 該当ウィジェットのデータをエージェントごとに区分して照会できる詳細画面が表示されます。

❗ ウィジェットによって使用可能なオプションは異なる場合があります。

プリセット設定

ダッシュボードからユーザーが設定したウィジェットの設定とレイアウト状態を保存して読み込むことができます。ウィジェットのサイズを調整し、希望する場所に配置して新しいプリセットを作成できます。



- **Default** : XOS指標とAWS RDS CloudWatch指標、主なDB指標で構成されたプリセットです。
- **Default(xos)** : XOS指標と主なDB指標で構成されたプリセットです。
- **Default(rds)** : AWS RDS CloudWatchの指標と主なDB指標で構成されたプリセットです。

- ❗
- プリセットに保存できるダッシュボードの設定は次のとおりです。
 - ウィジェットのサイズ、位置
 - ユーザー定義ウィジェットに定義された設定
 - **Session Table**ウィジェットに定義されたテーブルの幅とフィルター、選択したカラム値
 - 基本プリセット(デフォルト)は変更できません。
 - データベース環境によって、一部のプリセットはサポートされない場合があります。
 - **プロジェクト**修正権限のあるメンバーのみがプリセットを保存または変更できます。メンバー権限の詳細については、[次の文書](#)を参照してください。

新しいプリセットを作成する

1. ダッシュボードから、希望する形にウィジェットを配置してみてください。サイズを変更して、頻繁に確認するウィジェットのみを配置することもできます。
2. 画面の右上にボタンを選択します。
3. 新しいプリセット名を入力します。



エージェント選択履歴と一緒に保存するには、**エージェント選択履歴を含む**を選択します。

4. 保存ボタンを選択します。

プリセット一覧から新しく保存したプリセットを確認できます。

- ① 新しく作成したプリセットに変更が加えられた場合は、再度プリセットを保存してください。📁ボタンを選択し、同じ名前でプリセットを保存します。既存のプリセットに変更を上書きします。
- ダッシュボードの変更を保存せずに別のメニューに移動すると、変更は保存されません。
- プリセットはプロジェクト単位で保存され、他のユーザーと共有できます。

プリセットを削除する

使用していないプリセットは、プリセット一覧から削除できます。プリセット一覧から削除したい項目の右側に **×** ボタンをクリックします。

- ① **Default** プリセットは削除できません。

主なウィジェットを調べる

ダッシュボードに配置できるウィジェットの種類は次のとおりです。

XOSウィジェット

- [XOS] CPU

エージェント別のCPU使用率をリアルタイムで確認できます。

- [XOS] Memory

エージェント別のメモリ使用率をリアルタイムで確認できます。

- [XOS] Disk Usage

エージェント別のディスク使用率と使用可能サイズ、全体サイズ情報を表示します。

- : **ディスク使用量の詳細**ポップアップが表示されます。時間帯別のディスク使用量を表示するチャートを確認できます。インスタンスごとに確認でき、**Size**、**Free**、**Used**ボタンを選択すると、全体のサイズ、残りのサイズ、使用中のサイズが確認できます。
- : ウィジェットに表示されたテーブルをより広い画面で確認できます。

✔ データが表示されない場合は、[xos.conf](#)ファイルに次の設定を追加します。

xos.conf

```
# disk usage 0ならばoff、単位: 秒  
disk_usage_interval=60
```

AWS RDS CloudWatchウィジェット

- [RDS] CPUUtilization

RDSデータベースのインスタンス別CPU使用率をリアルタイムで確認できます。

- [RDS] FreeableMemory

RDSデータベースのインスタンス別メモリ使用率をリアルタイムで確認できます。

- [RDS] FreeStorageSpace

RDSデータベースの使用可能なストレージ容量をリアルタイムで確認できます。

❗ AWS CloudWatch指標照会のためのエージェント設定方法は、[次の文書](#)を参考にしてください。

Databaseウィジェット

提供するウィジェットを通じて、データベース性能分析の主要メトリクスとして、CPU、メモリ、ディスクI/Oなど多様なシステムリソースの状態を確認できます。

- **Active Sessions**

時間に応じたアクティブセッションの数値を時系列チャートで視覚的に表示します。データベースで実行されているセッションの数を時間ごとに追跡し、グラフ化したウィジェットです。セッションの使用パターンと負荷を視覚的に把握するのに役立ちます。

- **Lock Wait Sessions**

時間に応じたロック待機セッションの数値を時系列グラフで視覚的に表示します。データベースでロック待機状態にあるセッションの数を時間ごとに追跡し、リアルタイムグラフで表示したウィジェットです。

- **Long Active Session Count**

データベースで実行される活性セッションの長期実行に関する情報を提供します。セッションの実行時間に応じてグループ化され、時間単位で集計された情報をチャートで表示します。長時間実行されるセッションを識別してデータベースの性能問題やリソース漏れを早期に感知したり、セッション実行時間による分類を通じてどのセッションが長期間にわたり活性状態に維持されているかを視覚的に把握できます。

3秒未満は青、3秒以上10秒未満は緑、10秒以上15秒未満はオレンジ、15秒以上は赤で表示します。

- **Long Waiting Session Count**

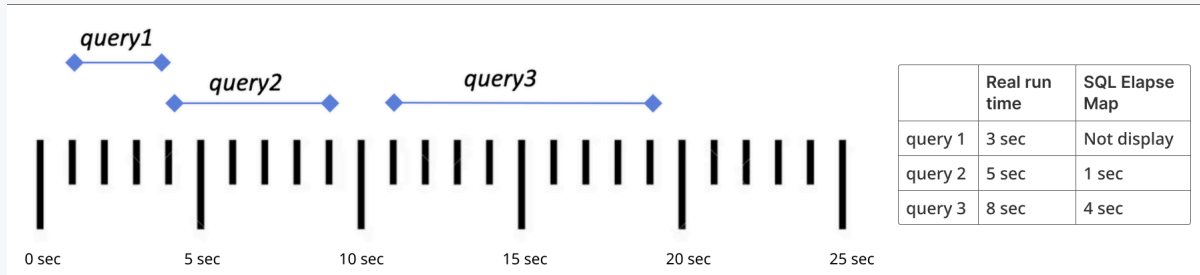
データベースでロックを保持または待機するセッションの長期待機時間に関する情報を提供します。ロック情報と活性セッション情報を組み合わせて、待機中のセッションの長期待機時間を追跡できます。長時間ロックを保持または待機するセッションをモニタリングします。どのセッションが長時間ロックを保持または待機しているかを視覚的に把握できます。

5秒未満は青、5秒以上10秒未満は緑、10秒以上60秒未満はオレンジ、60秒以上は赤で表示します。

- **SQL Elapse Map**

実行中のクエリの経過時間をリアルタイムチャートで表示します。チャートの特定の領域をドラッグするとQuery List一覧を確認できます。へまたはvボタンを選択して、縦軸の範囲を拡大または縮小できます。

- ① クエリの実行時間は、アクティブセッションを収集して計算します。その結果、アクティブセッションの収集サイクルによって誤差範囲が発生する可能性があります。次の図を参照してください。収集サイクルの間に終了したSQL実行時間は計算されません。収集サイクルが短いほど誤差範囲は小さくなりますが、DBの負荷を考慮して一般的に5秒に設定されています。長時間にわたって実行されるクエリをモニタリングする場合に役立ちます。SQL統計メニューのデータは同じです。



- **DML Tuples**

DML(Data Manipulation Language)文法(statement)が実行された回数をリアルタイムチャートで表示します。

- **Slow Query**

時間帯別のスロークエリをリアルタイムチャートで表示します。チャートの特定の領域をドラッグすると**Slow Query List**一覧を確認できます。 ^または v ボタンを選択して、縦軸の範囲を拡大または縮小できます。

- **Logical I/O**

`blks_hit` メトリクス、ディスクブロックがすでにバッファキャッシュで検出され、読み取りを必要としない合計回数です。

- **Physical I/O**

物理的なDiskからデータを読み取った回数です。

- **Uptime**

エージェント(インスタンス)が起動してからどれくらいの時間が経過したかを確認できます。時間は `{dd}d {hh:mm:ss}` 形式で表示します。

- **Commits**

システムまたはセッションで発生したコミット(Commit) の回数です。

- **Replication Delay**

Replication構造の場合、replication DBに反映される遅延時間をリアルタイムチャートで表示します。

- **Wait Event**

各バックエンドプロセスが、現在待機中の待機イベント名とそのイベントの発生頻度をリアルタイムのグラフで表示します。主にApplication、Configuration、Administrative、Networkなどに関連する項目の待機イベントをモニタリングします。各待機イベントの発生頻度を視覚的に確認でき、グラフの上にマウスオーバーすると、ツールチップが表示され、各待機イベントに関する詳細情報を提供します。

- **Vacuum Sessions**

進行中のVacuumプロセス数を確認できます。ウィジェット右上にある  ボタンを選択すると**Vacuum履歴**画面が表示されます。**Vacuum履歴**では行われたVacuumプロセスの詳細情報を確認できます。

Session Tableウィジェット

画面下段の**Session Table**ウィジェットでは、リアルタイムで実行されているアクティブセッションとロックツリー情報を照会できます。

- **アクティブセッション**：リアルタイムで実行中のセッションとクエリ内容を照会できます。

- 実行中のセッション項目を選択すると**セッション詳細**画面が表示され、選択したセッションの詳細を確認できます。[次の文書](#)を参考にしてください。
- テーブルの右側にある**query**カラムの項目を選択すると、**SQL詳細**画面が表示されます。選択したクエリ(Query)のFull TextおよびPlan情報を確認できます。詳細については、[次の文書](#)を参照してください。

- **ロックツリー**

ロックセッションに関する情報やHolderとWaiter情報をリアルタイムで照会できます。テーブルのカラムの詳細については、[次の文書](#)を参照してください。

❗ テーブル一覧の文字の色は黒 → **オレンジ** → **赤**順にセッションの実行速度が遅いことを意味します。

セッション詳細情報を確認する

アクティブセッションタブでセッション項目を選択すると**セッション詳細**画面が表示されます。選択したセッションについての詳細を確認できます。

The screenshot displays the PostgreSQL monitoring interface. At the top, a table lists active sessions. One session is highlighted with a yellow border. Below this, the 'セッション詳細 (pid: 10087)' panel is shown. It contains three sections: Session Info, Wait Info, and Process Info. The Session Info section lists various session parameters like pid, datname, query_start, runtime, state, backend_xmin, client_addr, client_hostname, and query_param. The Wait Info section shows wait_event and wait_event_type. The Process Info section displays resource usage metrics like cputime, cpuusage, elapse, vsize, rss, state, ioread, and iowrite. At the bottom, the SQL Text editor shows a query to extract session data for the selected pid.

instance	pid	datname	cpuusage	client_addr	username	runtime ↓	state	wait_event	query	query_param	application_n
DBX-1-203-5432	10087	postgres	0.33%	192.168.1.204	whatap	0	active		SELECT /* WhaTap2D#4 */ datname,pid,extrac...	'idle'	PostgreSQL 15

セッション詳細 (pid: 10087)

|| LIVE 08:34:18 Kill Session セッション履歴 >

Session Info

pid	10087	query_hash	348448832
datname	postgres	username	whatap
query_start	2024-03-11 08:34:15	application_name	PostgreSQL JDBC Driver
runtime	0	backend_start	2024-03-11 05:31:50
state	active	backend_type	client backend
backend_xmin	2093749	xact_start	2024-03-11 08:34:15
client_addr	192.168.1.204	state_change	2024-03-11 08:34:15
client_hostname	192.168.1.204	client_port	52804
query_param	'idle',26353		

Wait Info

wait_event	
wait_event_type	

Process Info

cputime	14
cpuusage	0.00%
elapse	10948
vsize	344528
rss	45188
state	active
ioread	0
iowrite	0

SQL Text Plan

Default Bind Reset Bind ☒ Formatting

```

SELECT
/* WhaTap2D#4 */
datname,
pid,
extract(
epoch
from
current_timestamp - query_start
) as runtime,
extract(

```

- セッションに関する情報は、既定値でリアルタイムで照会されています。タイムセレクターの || ボタンを選択すると、リアルタイム照会を中止し、中止した時間ベースのデータを確認できます。リアルタイム照会を中止した状態でデータを更新するには、🔄 ボタンを選択します。
- Kill Session**：選択したセッションを停止できます。セッション中止画面が表示されたらパスワード(Parameter Key)を入力し、適用ボタンを選択します。

① この機能は、リアルタイム照会モードでのみ利用できます。

- セッション履歴**：選択したセッション履歴が確認できるセッション履歴メニューに移動できます。

① セッション履歴メニューの詳細については、次の文書を参照してください。

- **Session Info**: 選択したセッションの基本情報とステータスを含む詳細を確認できます。
- **Wait Info**: 選択したセッションの待機イベントと待ち時間の情報を提供します。特定の作業を実行するために、データベースシステムで待機する時に発生する待機イベントに関する情報も確認できます。
- **SQL Text / Plan**: 選択したセッションのSQLクエリ文とPlan情報を確認できます。詳細については[次の文書](#)を参照してください。
- **Process Info**: 選択したセッションのプロセス情報を確認できます。

SQL 詳細情報の確認

実行中セッションのSQLクエリ情報を確認するには、**アクティブセッションセクションのquery(またはsql_text)** カラム項目を選択します。**SQL詳細画面**が表示されます。SQLクエリ文とPlan情報を確認することができます。

SQL統計の表示 → ボタンをクリックすると、選択した SQL クエリに関連する統計情報を確認できる **SQL統計** に移動します。

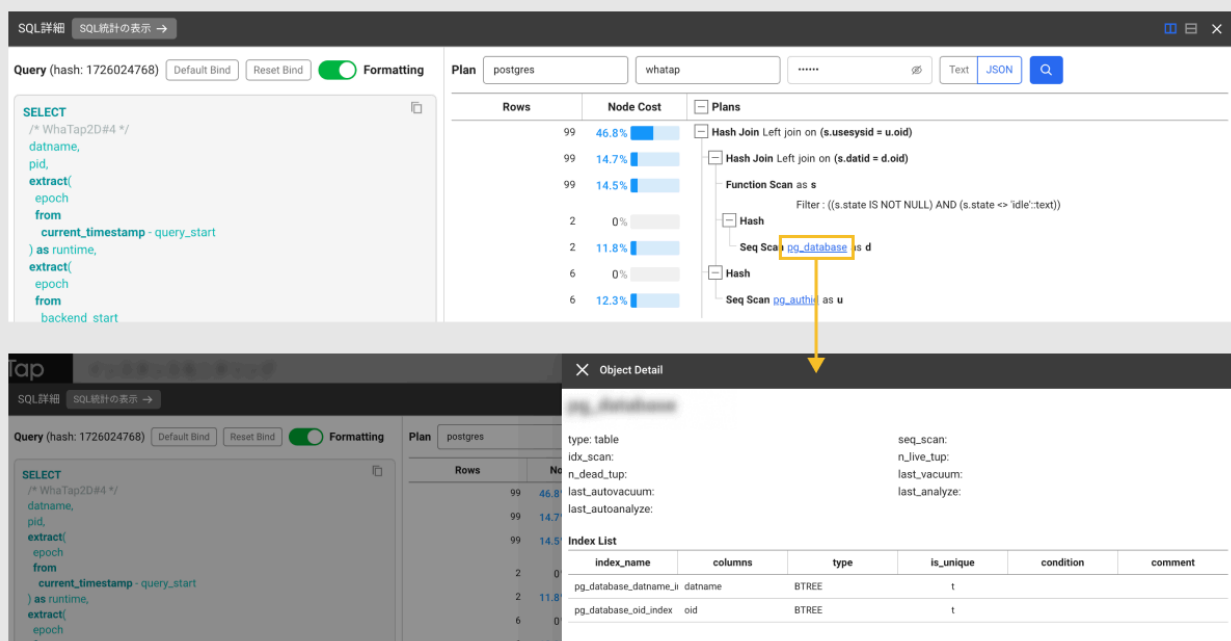
The screenshot displays the SQL Query Detail and Execution Plan interface. At the top, the selected query is shown: `START_REPLICATION ...` and `SELECT /* ...`. Below this, the **SQL 詳細** (SQL Detail) tab is active, showing the full SQL text. The **SQL統計の表示** (Show SQL Statistics) button is visible. The **Plan** tab shows the execution plan for the query, including a table of Rows, Node Cost, and a detailed tree diagram of the plan steps.

Rows	Node Cost	Plans
4	0%	Append
1	13.9%	Aggregate
606149825272	86.1%	Nested Loop Inner join
579686	0%	Seq Scan on boarding_passes as b
1045652	0%	Materialize
1045652	0%	Seq Scan on ticket_flights as a
3	0%	Nested Loop on ((tf.ticket_no = bp.ticket_no) AND (tf.flight_id = bp.flight_id)) Left join
3	0%	Nested Loop Inner join
3	0%	Nested Loop Inner join
3	0%	Nested Loop Inner join
3	0%	Index Only Scan on ticket_flights as tf on (tf.ticket_no = '0005432661915':bpchar)
1	0%	Index Scan on flights as f on (f.flight_id = tf.flight_id)
1	0%	Index Scan on airports_data as ml on (ml.airport_code = f.departure_airport)
3	0%	Index Scan on airports_data as ml_1 on (ml_1.airport_code = f.arrival_airport)
3	0%	Materialize
3	0%	Bitmap Heap Scan on boarding_passes as bp
3	0%	Bitmap Index Scan on (bp.ticket_no = '0005432661915':bpchar)

- **Default Bind:** クエリ文で変数として設定された部分を既定値としてバインディングできます。
- **Reset Bind:** バインディングされた既定値を解除し、変数を確認できます。
- **Formatting:** SQLクエリ文にインデントおよびフォーマットを適用して可読性を高めます。
- **Plan:** Plan情報を確認するにはDB Name、User Name、Passwordを入力してから、 ボタンを選択します。
 - **Text:** テキスト形式でプラン情報を確認できます。
 - **JSON:** JSON形式でプラン情報を照会すると、ボトルネックが発生した箇所がすばやく確認できます。

① JSON形式でプラン情報を照会する機能は、DBXエージェント1.6.12バージョンから対応しています。

- ✔ JSON形式でプラン情報を照会した状態でtableまたはindex項目を選択すると、テーブルのcolumn、index構成が確認できるObject detailウィンドウが表示されます。



The screenshot shows the DBX interface with the SQL Plan window and the Object Detail window. The SQL Plan window displays a query and its execution plan. The Object Detail window shows the details of the 'pg_database' table, including its columns and indexes.

SQL Plan Window:

Query (hash: 1726024768) Default Bind Reset Bind Formatting Plan postgres whatap

SELECT /* WhaTap2D#4 */
datname,
pid,
extract(
epoch
from
current_timestamp - query_start
) as runtime,
extract(
epoch
from
backend_start

Rows Node Cost Plans

99 46.8% Hash Join Left join on (s.usessysid = u.oid)
99 14.7% Hash Join Left join on (s.datid = d.oid)
99 14.5% Function Scan as s
Filter: ((s.state IS NOT NULL) AND (s.state <> 'idle':text))
2 0% Hash
2 11.8% Seq Scan pg_database as d
6 0% Hash
6 12.3% Seq Scan pg_auth as u

Object Detail Window:

pg_database

type: table
idx_scan:
n_dead_tup:
last_autovacuum:
last_autoanalyze:

seq_scan:
n_live_tup:
last_vacuum:
last_analyze:

Index List

index_name	columns	type	is_unique	condition	comment
pg_database_datname_i	datname	BTREE	t		
pg_database_oid_index	oid	BTREE	t		

この機能は、DBXエージェント1.6.15バージョン以降に対応しています。データベース権限に関する設定は、[次の文書](#)を参考にしてください。

AI チューニングガイド

AI チューニングガイドは、SQL クエリ、プラン、統計情報を分析してパフォーマンス問題を診断し、最適化の提案を行う機能です。

開発者や DBA がボトルネックの原因を迅速に特定し、効率的な SQL によりパフォーマンスを改善できるよう支援します。

⚠ 使用条件および注意事項

PostgreSQL、MySQL、SQL Server ではプランの取得が必須です。

プランを取得しない場合、AI チューニングガイドアイコンは （無効）として表示され、機能を使用できません。

- AI が生成する結果は自動分析に基づいており、精度が 100% ではないことにご留意ください。

1. 分析および診断する SQL をクリックし、**SQL詳細** 画面に移動します。
2. **SQL詳細** 画面の右下にある AI チューニングガイドアイコン  をクリックして AI 分析を開始します。
3. AI 分析結果を確認します。

結果項目	説明
クエリプランおよび概要	クエリの目的と実行概要を表示します。 実行回数、累積実行時間、データベース全体の負荷比率を分析し、当該 SQL がシステムパフォーマンスに与える影響を評価します。
パフォーマンス分析	分析結果を総合し、パフォーマンススコアと診断結果を提供します。 CPU 使用率、ディスク使用率、キャッシュヒット率、待機時間など、クエリ実行過程における詳細なリソース使用量を分析し、ボトルネックの発生区間を視覚的に表示します。
発見された主な問題	主な問題点を要約して表示します。
最適化推奨事項	検出された問題に基づき、最適化されたクエリを提案します。

デフォルトプランを照会する

デフォルトプランユーザーファイル作成時、エージェントが事前定義されたDB情報を使用してプランを自動的に照会するように設定できます。 **デフォルトプランの照会**項目のチェックボックスを選択した後、次のように予め入力したDB接続情報を通じてプラン情報を照会します。

• Linux Windows

1. エージェント設定(`whatap.conf`)ファイルに `plan_db` オプションを追加し、DB名を入力します。

`whatap.conf`

```
plan_db=(dbname)
```

2. DB設定後 `planuid.sh` スクリプトを使用してユーザーとパスワードを設定します。

```
./planuid.sh user pwd
```

❗ ユーザー(`user`)とパスワード(`pwd`)に特殊文字が含まれている場合、各特殊文字の前にバックスラッシュ(`\`)を追加します。

例、 `user$example` → `user\example`

1. エージェント設定(`whatap.conf`)ファイルに `plan_db` オプションを追加し、DB名を入力します。

`whatap.conf`

```
plan_db=(dbname)
```

2. DB設定後 `planuid.bat` スクリプトを使用してユーザーとパスワードを設定します。

```
planuid.bat user pwd
```

テーブルデータのフィルタリング



1. テーブルの右上にある  ボタンを選択します。
2. テーブルのカラムヘッダー項目と条件を選択します。
3. **条件を入力してください。** テキストボックスに条件を入力します。
4. **保存** ボタンを選択します。

テーブルカラムの設定

テーブルヘッダーカラムを非表示にしたり、希望する項目を追加できます。カラムの順序を変更することもできます。 ボタンをクリックします。

- ① • 設定を完了した後は、**確認** ボタンを選択すると、設定事項がテーブルに反映されます。
- ③ 検索バーにテキストを入力し、カラム項目を検索できます。入力したテキストと一致するカラム項目が表示されます。
- 画像は、製品またはプロジェクト、メニューによって異なる場合があります。

カラムを追加する

- ① 一覧でテーブルヘッダーカラムに追加する項目を選択します。すべての項目を追加するには、**すべて選択** を選択します。

カラムを削除する

① 一覧から削除するカラム項目のチェックボックスのチェックを解除します。または② 一覧から削除する項目の右側の×ボタンを選択します。

カラムの順序の変更

② 一覧で順序を変更するアイテムをドラッグして、希望する位置に移動できます。

設定事項の初期化

変更事項をすべてキャンセルし、初期化するには、🔄 初期化ボタンを選択します。

カラム情報ガイド

- [Active session](#) Lock tree Process information

項目	説明
db	データベース名
user	SQL実行アカウント
application	クライアントアプリケーション名
client_host	接続されたクライアントのホスト名(client_addrの逆引きDNS参照で報告されます)
client_addr	クライアントのIPアドレス
client_port	クライアントのポート番号(Unixソケットを使用する場合は-1)
query_hash	クエリのハッシュ値(WhaTap内部管理値)
query_start	現在アクティブ状態のクエリが開始された時間(アクティブ状態でない場合、最後にクエリが開始された時間)

項目	説明
state	バックエンドプロセスの現在の状態(active/idleなど)
state_change	stateが変更された最近の時間
xact_start	そのプロセスの現在のトランザクションが開始された時間(アクティブTXがない場合は null)
backend_start	そのバックエンドプロセスが開始された時間
backend_xmin	バックエンドプロセスが処理中の xid の最小値
backend_type	バックエンドプロセスのタイプ、 autovacuum worker、 logical replication launcher、 client backend、 checkpointer など
backend_xid	バックエンドが使用中の xid
wait_event_type	バックエンドプロセスが待機中のイベント type
wait_event	バックエンドプロセスが現在待機中の場合は、待機イベント名

ⓘ WhaTapは、クライアントに関する情報をデフォルトで保存します。

項目	説明
Instance	インスタンス（エージェント）名
pid	セッションID
lock_mode	ホルダーセッションのロックモード - RowExclusiveLock：INSERT、UPDATE、DELETE - AccessExclusiveLock：DDL（ALTER、DROP、TRUNCATE など）によるロック - AccessShareLock：SELECT 時に使用 ほとんどすべてのロックと共存可能 - ShareLock：LOCK TABLE ... IN SHARE MODE
holder_relation	ロックを保有しているセッションがアクセス中のテーブル名またはオブジェクト識別子

項目	説明
waiter_type	ロックがどの対象に対してかかっているかを示します - relation：テーブルにかかっているロック tuple：特定の行（row）にかかっているロック - transactionid：トランザクションにかかっているロック
waiter_relation	ロックを待機しているセッションがアクセス中のテーブル名またはオブジェクト識別子
lock_request	ロックを待機しているセッションのロックモード lock_mode を参照

次の項目は、XOSエージェントをインストールした場合に収集される指標です。

項目	説明	単位
pid	プロセスID	-
cpuusage	プロセスが使用するCPU使用率	%
cputime	プロセスがこれまでに使用した累積CPU時間	秒
elapsed	プロセスが実行されてからの経過時間	秒
vsize	Virtual memory size. 全体の仮想メモリ使用量（RAM + Swap + 予約）	KB
rss	Resident Set Size. 現在のプロセスが占有している実際の物理メモリサイズ	KB
pss	Proportional Set Size. 共有メモリを含む実際の物理メモリ使用量	KB
state	プロセスの現在の状態コード（例：R、S、D、T、Z など）	-
ioread	累積読み込みバイト数（read_bytes）	バイト（B）
iowrite	累積書き込みバイト数（write_bytes）	バイト（B）
uid	このプロセスを実行したユーザーID	-

項目	説明	単位
cmd	実行されたコマンドの短縮形式	-
longcmd	実行されたコマンドの完全な形式（フルパスおよび引数を含む）	-

照会リストをダウンロードする

現在照会中の[Session Table](#)リストをCSVファイルでダウンロードできます。[Session Table](#)セクションの右上の  ボタンを選択します。

ダウンロードしたファイル名の形式は次のとおりです。

- アクティブセッション: [activeSession_HH_MM_SS.csv](#)
- ロックツリー: [lockTree_HH_MM_SS.csv](#)

別ウィンドウで見る

[Session Table](#)リストを別ウィンドウで開き、より広い画面で確認できます。[Session Table](#)セクションの右上の  ボタンを選択します。

スロークエリ

スロークエリ画面では、DBパラメータで設定された基準時間を超えた遅いSQLを監視・確認できます。PostgreSQLはスロークエリをログファイルに記録しますが、確認するにはDBサーバーにターミナルで接続してテキストログを直接読み取る必要があります。スロークエリログをリアルタイムで収集し、視覚的なチャートとリストで提供します。

- **Slow Query**チャートで、遅いクエリの発生時点と実行時間を視覚的に把握できます。
- **Slow Query List**で、個別クエリの詳細情報を確認し、実行計画（Plan）を照会できます。
- リアルタイム（Live）モードと過去時間照会の両方に対応しています。

❗ スロークエリ収集には追加エージェントが必要です。オンプレミス環境では**XOSエージェント**、クラウド（AWS RDSなど）環境では**DBXCエージェント**をインストールしてください。

✔ スロークエリチャートはダッシュボードのSQL Elope Mapチャートと類似していますが、以下の違いがあります。

表 | スロークエリチャートとSQL Elope Mapチャートの違い

区分	スロークエリ	SQL Elope Map
データソース	DBが記録するスロークエリログ	エージェントが5秒間隔で収集するアクティブセッション
実行時間の精度	DBが記録した実際の実行時間で正確	5秒間隔収集のため、最大5秒の誤差が発生する可能性あり
収集対象	設定基準時間（ <code>log_min_duration_statement</code> ）を超えたクエリのみ	収集時点で実行中のすべてのアクティブセッション

事前設定

この画面にデータを表示するには、**DB設定**、**DBXエージェント設定**、環境に応じて**XOSエージェント**または**DBXCエー**

ジェントの設定が必要です。

DB設定

`postgresql.conf`でスロークエリの基準時間を設定します。変更後、PostgreSQLの再起動または `pg_reload_conf()` の実行が必要です。

postgresql.conf

```
log_min_duration_statement = 500
```

パラメータ	説明
<code>log_min_duration_statement</code>	設定値（ミリ秒）を超えるSQLをログに記録します。例： <code>500</code> の場合、500ms以上かかるSQLが記録されます。

DBXエージェント設定

`whatap.conf`に以下の設定を追加します。

whatap.conf

```
slow_query_log=ON
```

設定項目	タイプ	デフォルト値	説明
<code>slow_query_log</code>	String	OFF	スロークエリログ収集の有効化

OSエージェント設定

スロークエリログを収集するには、DB運用環境に合ったエージェントが必要です。

環境	必要なエージェント	主な設定	設定ガイド
オンプレミス（Linux、Windows）	XOSエージェント	なし（自動認識失敗時は <code>slow_query=</code> にログファイルパスを直接指定）	XOSエージェント設定
クラウド（AWS RDS、NCPなど）	DBXCエージェント	<code>slow_query: true</code> （config.yamlのインスタンス別設定）	DBXCエージェント設定

スロークエリ

基本オプション

スロークエリ画面上部のフィルター領域で、照会する期間と対象DBインスタンスを選択できます。

- ・ **時間:** リアルタイム（Live）モードまたは過去の時間を選択します。Liveモードは10秒間隔で自動更新されます。
- ・ **インスタンス:** 照会対象のDBインスタンスを選択します。全体または個別インスタンスを選択できます。

Slow Queryチャート

スロークエリが発生した時点をscatterチャートで表示します。Y軸は実行時間（秒）、X軸は時間です。

- ・ 各点は1つのスロークエリを表し、上に位置するほど時間がかかったクエリです。
- ・ チャート領域をマウスドラッグすると、該当区間のクエリのみ下部リストに表示されます。
- ・ 右上の**^ / v**ボタンでY軸の範囲を調整できます。

Slow Query List

チャートで選択した区間に該当するスロークエリリストを表示します。

カラム	説明
Instance	クエリが実行されたインスタンス名

カラム	説明
DB	データベース名
User	実行ユーザー名
Log Time	スロークエリログの記録時刻
Type	ログタイプ（例：statement）
Coltype	コレクションタイプ
Duration (s)	クエリ実行時間（秒）
SQL	実行されたSQL文

SQL詳細表示

SQLカラムのクエリをクリックすると、詳細情報ウィンドウが開きます。

データ解釈ガイド

主な確認ポイント

- **Duration**が長いクエリ: 個別実行時間が長いクエリ → SQLチューニング、インデックス確認
- 同一のSQLが繰り返し表示される場合: 特定のクエリが継続的に遅い → 根本的なチューニングが必要
- 特定の時間帯に集中している場合: 該当時間のDB負荷、ロック競合、バッチ処理など外部要因を確認

実践活用シナリオ

シナリオ1. 遅いクエリのリアルタイムモニタリング

1. LiveモードでSlow Queryチャートを観察
2. 新しい点が現れたら、該当領域をドラッグして選択
3. Slow Query ListでクエリをSQLをクリックして詳細情報と実行計画を照会

シナリオ2. 特定時間帯のスロークエリ分析

1. 時間フィルターで問題が発生した時間帯を選択
2. チャートでクエリが集中している区間をドラッグ
3. Durationが長いクエリから確認して原因分析

シナリオ3. スロークエリ基準の調整

1. スロークエリが多く収集される場合、`log_min_duration_statement` の値を上げて基準を緩和
2. スロークエリが収集されない場合、`log_min_duration_statement` の値を下げるか、`slow_query_log=ON` 設定を確認

AIで分析する

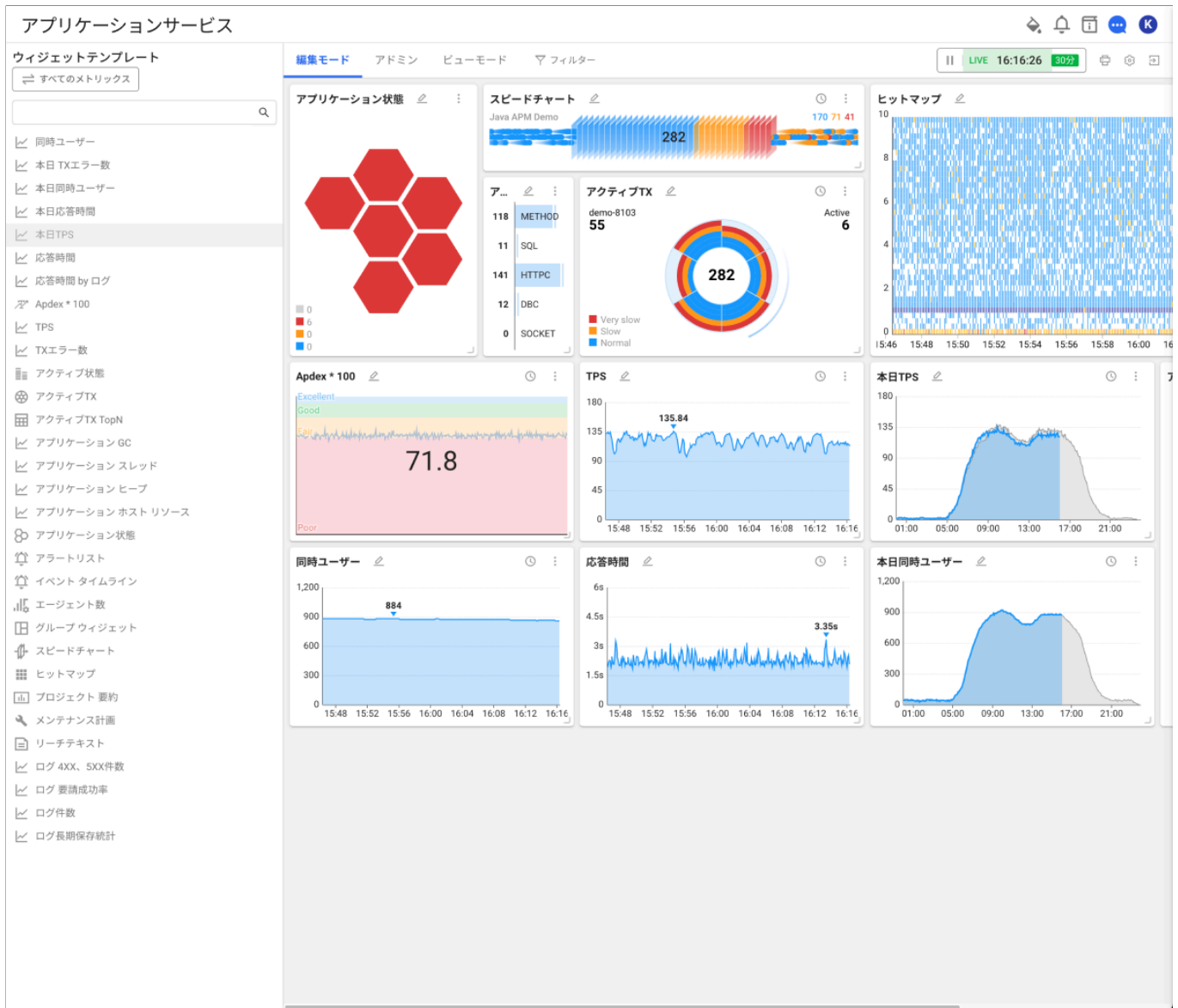
画面で**WhaTap AI**ボタンをクリックすると、WhaTap AIチャットウィンドウが開き、照会した画面データを要約して分析した結果が対話形式で表示されます。結果を読んだ後、同じ対話で続けて質問できます。

- 分析に使用するデータは散布図で選択した項目のうち、所要時間が長い上位20件です。
- 分析対象は画面に照会されたデータです。照会条件を変更してから再度分析すると、変更後のデータを基準に分析します。
- 対話履歴には `AI分析 - {画面名}` 形式で記録されます。

Flexボード

Flexボードはユーザー定義型の統合ダッシュボードです。アプリケーション、サーバー、データベース、コンテナなどのプロジェクトのデータを使用して、リアルタイムのダッシュボードが作成できます。

事前構成されたテンプレートで初期設定を迅速に完了し、希望する形式のダッシュボードを作成できます。データウィジェットを追加したりプロパティを修正したりして、必要な形式で表示することができ、フィルタリング機能によって監視対象を絞り込むこともできます。時間範囲を指定すれば特定の時点のデータを確認でき、補助チャートを活用することで多様な方法で分析できます。ダッシュボードはお気に入りに登録して簡単にアクセスでき、カスタマイズしたダッシュボードは他のアカウントにコピーして再利用することも可能です。



Flexボードは、次のアクセスパスによってさまざまな用途があります。

- ホーム画面 > 統合Flexボード

- ウィジェットを作成すると、表示可能なすべてのプロジェクトを選択オプションとして提供します。
- ユーザーアカウントにダッシュボードが保存され、他のユーザーにコピーする機能を使用して共有できます。
- 個人アカウントダッシュボードで、権限による影響はありませんが、読み取り専用として共有されたダッシュボードの場合は修正できません。

- ホーム画面 > プロジェクト選択 > **ダッシュボード** > **Flexボード**
 - ウィジェットを作成すると対応するプロジェクトの情報が自動入力されます。
 - プロジェクトメンバーで作成したFlexボードは、自動共有されます。
 - プロジェクト**修正権限**、または**プロジェクトのFlexボードの修正権限**、**Site Admin**権限を持つユーザーが修正できます。

✔ Flexボードの編集権限を持つユーザーは、以下の機能を利用できます。

- ダッシュボードをJSONファイルとしてエクスポート/インポート
- ダッシュボード内のデータリクエストおよびレスポンス内容を確認
- ウィジェット設定オプションをJSON形式で参照および修正

❗ プロジェクト内の**Flexボード**メニューでは、ダッシュボードの変更権限のあるユーザーのみが**編集モード**および**アドミンモード**、**フィルター**機能にアクセスできます。

- アクセス可能なメンバー権限
 - プロジェクト**修正権限**
 - **プロジェクトのFlexボードの修正権限**
 - **Site Admin**権限
- **ビューモード**および**フィルター**機能の詳細については、[次の文書](#)を参照してください。
- メンバー権限の詳細については、[次の文書](#)を参照してください。

Flexボードの使用方法

- ✔ ダッシュボードはメトリクスウィジェットを含むさまざまなウィジェットテンプレートを活用して構成できます。
- 各テンプレートの特徴と使用方法については、[ウィジェットテンプレート](#) ドキュメントで詳しく確認できます。

事前テンプレートを適用する

事前構成テンプレートで初期設定を素早く進め、必要に応じて修正して活用してください。

1. **Flexボード** メニューを選択します。
2. 左側の **テンプレート** 一覧から使用するテンプレートを選択します。
3. **Flexボード管理** ウィンドウが表示されたら、**名前変更** に希望の名前を入力します。
 - **レイアウト** でウィジェットを選択し、ドラッグして希望の位置へ移動できます。
4. **保存** ボタンをクリックします。選択したテンプレートを基準に Flex ボードを作成します。
 - 作成したボードは **Flexボード** のボード一覧で確認できます。

Flex ボードを作成する

Flex ボードを作成し、ウィジェットを配置して自分だけのダッシュボードを作れます。

1. **Flexボード** で右上の **ダッシュボード生成** ボタンをクリックします。
2. **ダッシュボード生成** ウィンドウが表示されたら、入力ボックスにダッシュボードの **名前** を入力します。
3. ウィジェットの配置方式を選択し、**ダッシュボード生成** ボタンをクリックします。
 - **固定レイアウトFlexボード**: ピクセル基準の自由配置
 - **レスポンシブFlexボード**: ブラウザサイズ基準のグリッドレスポンシブレイアウト

(グリッドを設定してから **ダッシュボード生成** ボタンをクリック)

4. **ウィジェットテンプレート** 選択画面で、ダッシュボードに配置するウィジェットを選択します。

- メトリクスウィジェットを追加する場合は、**ウィジェットテンプレート** の右側にある **すべてのメトリクス** ボタンをクリックします。

- ❗ **ウィジェットテンプレート**: 一般的な監視状況で重要となる指標を要約し、事前定義されたウィジェット一覧から選択できます。
- **すべてのメトリクス**: ユーザーのプロジェクトで収集中のすべてのメトリクスデータを基準にウィジェットを作成します。メトリクスウィジェットの詳細は[次のドキュメント](#)を参照してください。

5. 配置したウィジェットの位置を移動したりサイズを調整します。

- ウィジェット上部で十字カーソルが表示されたら、クリックしたままドラッグして位置を移動します。
- ウィジェット右下で矢印カーソルが表示されたら、クリックしたままドラッグしてサイズを調整します。

6. 追加するウィジェットの配置が完了したら、**ビューモード** を選択してレイアウトを確認します。レイアウトを再編集する場合は、**編集モード** を選択してウィジェット配置を変更します。

7. すべての工程が完了したら、右上の (**リストへ**) アイコンをクリックします。

- **Flexボード** の **ダッシュボードリスト** で作成したボードを確認できます。作成したボードを選択して新しいFlexボードを確認してください。

- ❗ お客様のご要望を反映してダッシュボードテンプレートやウィジェットの追加を検討しています。必要なテンプレートまたはウィジェットがあれば support@whatap.io までご連絡ください。

Flex ボードを編集する

- ❗ **Flex ボード編集権限**



- **ホーム > 統合Flexボード:** 個人アカウントのダッシュボードで権限の影響はありませんが、閲覧専用で共有されたダッシュボードは編集できません。
- **ホーム > プロジェクト選択 > ダッシュボード > Flexボード:** プロジェクトの編集権限、プロジェクトのフレックスボード編集権限、または Site Admin 権限のユーザーが編集できます。

メンバー権限の詳細は[次のドキュメント](#)を参照してください。

1. **Flexボードのダッシュボードリスト**で、 編集アイコンをクリックします。画面右側に **Flex ボード管理** ウィンドウが表示されます。
2. **Flex ボード管理** ウィンドウで、名前、プロジェクト、レイアウト、widget.json、option.json を設定します。
 - **名前変更:** ダッシュボード名を編集します。
 - **プロジェクト:** ダッシュボードのウィジェットがデータを取得するプロジェクトを選択します。



プロジェクト オプションは **ホーム > 統合Flexボード** メニューでのみ設定できます。

統合Flexボード から入った場合は、ウィジェットごとにデータ取得元のプロジェクトを選択できます。

1. ウィジェット右上の ボタンをクリックし、目的のプロジェクトを検索して単一または複数選択します。
2. プロジェクトを選択したら **適用** ボタンをクリックします。

• レイアウト

- ウィジェットをドラッグして位置を移動します。
- ウィジェット右下をマウスでクリックしたままドラッグしてサイズを調整します。
- ウィジェットを削除するには、ウィジェット右上の アイコンをクリックします。
- **ウィジェット.json:** ダッシュボードのウィジェット設定を JSON 形式で読み込みます。 アイコンをクリックすると JSON をコピーします。
- **オプション.json:** ダッシュボードのオプション設定を JSON 形式で読み込みます。 アイコンをクリックすると JSON をコピーします。

Flex ボードを削除する

1. Flexボード > ダッシュボードリスト で、削除する項目の  アイコンをクリックします。
2. 確認メッセージで **削除** ボタンをクリックします。

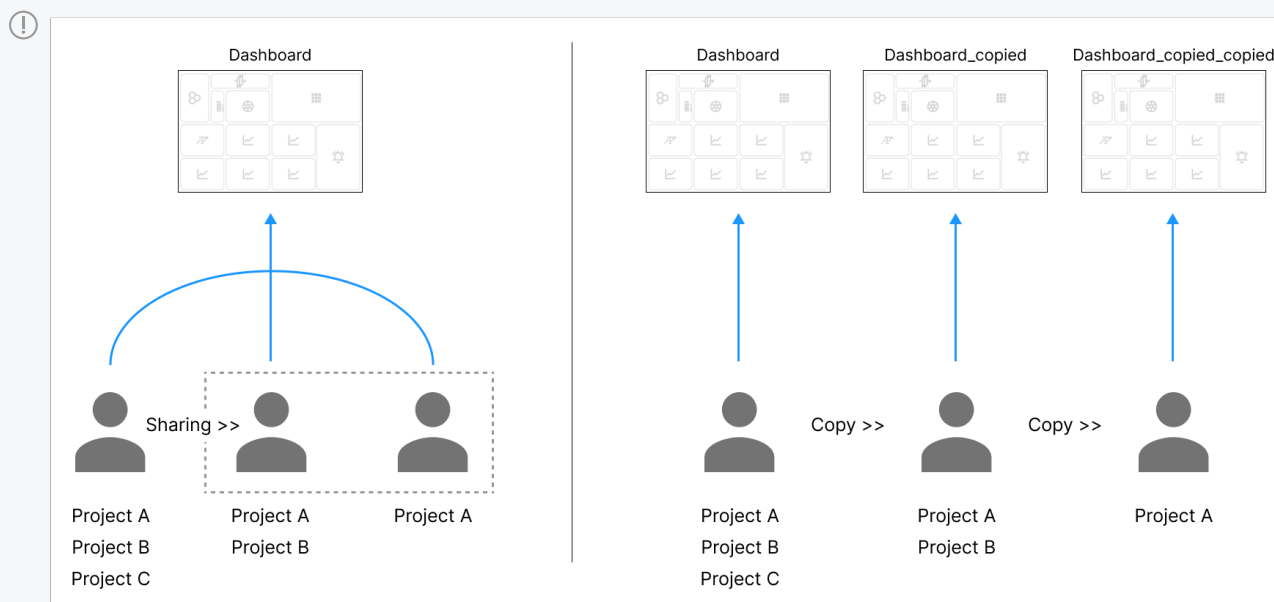
 ダッシュボードの削除は所有者と共有を受けたユーザーで動作が異なります。

- ダッシュボード所有者がダッシュボードを削除すると、共有されたすべてのユーザーアカウントでもそのダッシュボードが削除されます。
- 共有を受けたユーザーがダッシュボードを削除すると、そのユーザーの **ダッシュボードリスト** のみから削除されます。元のダッシュボードは維持されます。

Flex ボードを共有する

 Flex ボードの共有機能は **ホーム > 統合Flexボード** メニューでのみ使用できます。
プロジェクトの **Flexボード** メニューでは提供されません。

 共有とコピーの違い



- 共有: 共有されたメンバーは同じダッシュボードを閲覧します。
- コピー: ダッシュボードを他のメンバーまたは自分にコピーすると、コピーされた別個のダッシュボードを閲覧します。
- 読取り専用 で共有/コピーされたダッシュボードは編集できません。編集モード で共有/コピーされた場合のみ編集できます。
- 読取り専用 で共有を受けたメンバーも、そのダッシュボードを 閲覧専用で再共有 したり コピー できます。
- ダッシュボードに含まれるプロジェクトのうち **少なくとも1つ** のプロジェクトに所属しているメンバーにのみ共有できます。共有またはコピー先のメンバーがプロジェクトに所属しているか確認してください。



統合Flexボード で作成した Flex ボードを他のメンバーと共有できます。所有者が共有されたダッシュボードを編集すると、共有を受けたすべてのメンバーに同じ変更が反映されます。

1. 統合Flexボード の **ダッシュボードリスト** で、共有するダッシュボードの アイコンをクリックします。
2. **ダッシュボード共有** ウィンドウが表示されたら、共有するメンバーを1名または複数名選択するか、アカウントを直接入力します。
 - **読取り専用** を選択すると、共有を受けたメンバーはダッシュボードを編集できません。

- 共有対象は、ダッシュボードに含まれるプロジェクトのうち少なくとも1つのプロジェクトに所属している必要があります。

3. 共有 ボタンをクリックします。

- 共有された項目には **共有** タグが表示されます。**共有** タグにマウスを載せると、共有したメンバー情報を確認できます。
- 閲覧専用で共有された項目には **読取り専用** タグが表示されます。

ダッシュボードを共有で受け取った場合	ダッシュボードを共有した場合（ダッシュボード
 ダッシュボード所有者 owner@whatap.io 更新日時: 2023/06/14 10:22:05 共有 RUM ページ 5 ダッシュボード共有メンバーリスト 更新日時: 2023/06/14 10:19:52 共有 読取り専用 更新日時: 2023/06/05 16:41:36	 ダッシュボード所有者 demoowner@whatap.io 更新日時: 2023/07/06 16:25:06 共有 demoowner@whatap.io ダッシュボード共有メンバーリスト 更新日時: 2023/06/14 10:22:05 共有 更新日時: 2023/06/14 10:19:52

✓ ダッシュボード共有の条件

- ダッシュボード共有機能は **ホーム > 統合Flexボード** のみで提供します。
- だれでも自分のダッシュボードを共有でき、共有で受けたダッシュボードを再共有やコピーできます。
- 読取り専用** で受けたダッシュボードは編集できませんが、**編集モード** で受けた場合は編集できます。
- 所有者** がダッシュボードを削除すると、共有を受けたすべてのアカウントで該当ダッシュボードが削除されます。
- 共有を受けたユーザー** が削除した場合は、そのユーザーの **ダッシュボードリスト** のみから削除され、元のダッシュボードは維持されます。
- 共有で受けたダッシュボードに含まれるプロジェクトのうち、**ビュー** 権限がないプロジェクトはデータを参照できません。

Flex ボードをコピーする

ユーザーは Flex ボードをコピーして他のメンバーに渡せます。

1. 統合Flexボード のダッシュボードリスト で、 アイコンをクリックします。
2. コピーオプションを選択します。
 - **自分にコピー:** 自分のアカウントにコピーします。
 - a. `_copied` 接尾辞が付いた項目が **ダッシュボードリスト** に追加されます。
 - **他人にコピー:** 他のメンバーのアカウントにコピーします。
 - a. **+** **アカウント** をクリックします。
 - b. コピー先メンバーのメールを選択し、**ダッシュボードコピー** ボタンをクリックします。
 - c. そのメンバーの **ダッシュボードリスト** に `_copied` 接尾辞が付いた項目が追加されます。

JSON ファイルで共有する

Flex ボードの設定を JSON ファイルとして他のメンバーに渡したり、他のメンバーの設定を取り込みます。

エクスポート

1. 統合Flexボード またはプロジェクトの **Flexボード** メニューに移動します。
2. **ダッシュボードリスト** で  ボタンをクリックします。
3. エクスポートオプションを選択します。
 - **プロジェクト情報を含む:** ダッシュボードに設定されたプロジェクト情報を JSON ファイルに含めて共有
 - **プロジェクト情報を含まない:** ダッシュボードに設定されたプロジェクト情報を JSON ファイルに含めずに共有
4. JSON ファイルをダウンロードし、共有先のメンバーに渡します。

インポート

1. 統合Flexボード またはプロジェクトの Flexボード メニューに移動します。
2. 画面右上の  インポート ボタンをクリックします。
3. ダウンロードした JSON ファイルを選択します。

❗ この機能は、プロジェクトの 編集 権限を持つメンバーのみ利用できます。

Flex ボードの画面モード

Flex ボードの画面モードに応じて提供機能が異なります。

❗ アクセス可能なメンバー権限

プロジェクト内の Flexボード メニューでは、編集権限のあるユーザーのみが編集モード、アドミン、フィルター 機能にアクセスできます。メンバー権限の詳細は[次のドキュメント](#)を参照してください。

- プロジェクト編集 権限
- プロジェクトのFlexボードの修正 権限
- Site Admin 権限

モード種類

Flexボード は画面モードにより設定できる機能が異なります。

- **編集モード:** ウィジェットテンプレート一覧からレイアウトにウィジェットを配置し、ウィジェットの編集、削除、位置調整ができます。
- **アドミン:** ウィジェットの詳細プロパティを JSON 形式で管理できます。レイアウトに配置したウィジェット右上の  ボタンを選択します。
- **ビューモード:** ウィジェットの配置、サイズ調整、詳細オプションなどの設定を完了後、ダッシュボードのように各ウィジェットのデータをリアルタイムで監視できます。

共通機能

- データフィルタリング
- リアルタイムデータの参照範囲設定
- 印刷モード
- Flex ボード管理
- データ結合オプション

データフィルタリング

データフィルタリングで監視対象を絞り、効率的な Flex ボードを構成できます。

1. 画面上部の **▽ フィルター** タブをクリックします。
2. 画面右に表示される **フィルター** ウィンドウでフィルタリング対象を選択します。すべて選択するには **すべて選択** を選択します。
 - **エージェント オプション**: エージェント種別またはアプリケーション一覧を選択、**検索** 機能あり
 - **メトリクス オプション**: プロジェクトまたはエージェント種別を選択、**+ And** 条件の追加や、**含む**、**除外** による文字列ベースの自動選択が可能
3. フィルタリング対象を選択したら、**適用** ボタンをクリックします。
4. よく使う条件は **フィルター保存** ボタンで保存します。
 - a. フィルター名を入力し、**保存** を選択します。**保存フィルター** は **フィルター** ウィンドウ下部に表示されます。

フィルタリング条件

フィルタリング条件はエージェントタイプとメトリクスタイプに分かれます。

エージェントは基本的な監視対象のみ選択肢として提供し、メトリクスはプロジェクトで収集したすべてのメトリクスデータのタグを選択肢として提供します。メトリクスタイプは、非エージェント形式でデータを収集し、その後フィルタリングを適用する場合に有用です。メトリクスの詳細は[次のドキュメント](#)を参照してください。

タイプ	エージェント	メトリクス
選択オプション	WhaTap プロジェクトの監視対象 (アプリケーション、サーバー、データベース、 コンテナ、Kubernetes の POD と NODE)	エージェントを含むすべてのメトリクスタグ
適用範囲	すべてのウィジェット	メトリクスデータを使用するすべてのウィジェット
フィルタリング方式	チェックボックス	一致, 含む, 除外

参照時間を設定する

Flexボード の画面右上の時間セクターで、ウィジェットのデータ参照時間を設定できます。

- : リアルタイムモードを停止して特定の参照時間に設定
- : リアルタイムモードに戻す

印刷する

Flex ボード画面を PDF で保存して印刷できます。

1. Flex ボード画面の右上にある  アイコンをクリックして印刷モードに切り替えます。
2. オプションを設定します。
 - **フィルターオプション表示:** 画面のフィルターオプションを表示/非表示に切り替え
 - **{mode}方向に切換え / {mode}方向に切換え:** 画面を横向きまたは縦向きに変更
3. 完了後、**PDFダウンロード** を選択します。
4. 印刷モードを終了するには、右上の **× 印刷モード終了** をクリックします。

Flex ボード管理

Flex ボードのレイアウト変更、ウィジェットの位置移動やサイズ調整、名称変更などの管理機能を提供します。Flex ボード管理 機能は、**ダッシュボード > Flexボード メニューのダッシュボードリスト**で  アイコンをクリックして入

することもできます。

1. Flex ボード画面の右上の  アイコンをクリックすると、**Flexボード管理** ウィンドウが開きます。
 - **名前変更:** Flex ボード名の変更
 - **レイアウト:** ウィジェットの位置・サイズ調整および削除
 - **他のユーザーの修正を許可しない:** 変更制限オプション
 - 編集権限のあるメンバーは編集・削除・変更が可能で、オプションを有効化すると **他のユーザーの修正を許可しない: On** に変更されます
 - **ウィジェット.json:** Flex ボードに含まれるウィジェットを JSON 形式で編集・管理
 - **オプション.json:** Flex ボードの設定を JSON 形式で編集・管理（プロジェクト情報、Flex ボードフィルター、時間選択などの設定値）

 **ウィジェット.json と オプション.json** 機能は、ダッシュボード管理者権限のユーザーにのみ提供します。

2. すべての設定を完了し、**保存** ボタンをクリックします。

ダッシュボードをお気に入りに登録する

作成した Flex ボードはお気に入りに登録して素早くアクセスできます。**ダッシュボード > Flexボード** の **ダッシュボードリスト** で、お気に入り登録するダッシュボードの ☆ ボタンをクリックすると、★ に変わります。

ウィジェットの管理

Flexボードに配置したウィジェットを管理して、ダッシュボードをユーザーがカスタマイズします。

- ✓ ダッシュボードはメトリクスウィジェットを含むさまざまなウィジェットテンプレートを活用して構成できます。
各テンプレートの特徴と使用方法については、[ウィジェットテンプレート](#) ドキュメントで詳しく確認できます。

ウィジェット追加

作成済みのFlexボードにウィジェットを追加できます。

1. Flexボードメニューのダッシュボードリストで、ウィジェットを追加するダッシュボードを選択します。
 2. 画面左上の編集モードを選択します。
 3. 左側のウィジェットテンプレート一覧から目的のウィジェットを選択します。メトリクスウィジェットを追加するには、すべてのメトリクスを選択してから、目的のウィジェットをクリックします。
- ホーム画面の統合Flexボードを選択し、ウィジェットを追加した場合は、プロジェクト選択ボタンをクリックして、収集対象のプロジェクトを選択します。
 - 追加するウィジェットがある場合は、同じプロセスを繰り返してください。
4. すべての手順が完了したら、画面上部のビューモードを選択してレイアウトを確認し、 (リストヘ) ボタンをクリックします。

ウィジェットの属性変更

Flexボードメニューのダッシュボードリストで、ウィジェットのプロパティを変更するボードを選択します。編集モードに入ります。

ウィジェット名の変更

1. 名前を変更するウィジェットの名前の右側の  ボタンをクリックします。
2. ウィジェットの題名、フォントサイズ、色を設定し、**保存**ボタンを選択します。
 - 保存する際に、**すべてのウィジェットに適用**トグルボタンの機能を活性化すると、フォントサイズと色が他のウィジェットにも適用されます。
 - ウィジェットのタイトルをロックするには  ボタンを選択し、**保存**ボタンをクリックします。

時間設定

1. 時間を設定するウィジェットの右上の  ボタンを選択します。
2. **時間選択**ウィンドウで時間を設定してください。
 - ダッシュボード画面で設定した時間を適用するには、**ダッシュボード設定に従う**をチェックします。ダッシュボードの時間設定は右上にあります。
 - 特定の時間を設定する場合は、**ダッシュボード設定に従う**のチェックを外します。＜または＞ボタンを使用して照会する時間を設定します。
3. 時間設定が終了したら **×** ボタンをクリックします。

 ウィジェットのデータタイプによっては、時間設定機能をサポートしていない場合があります。

出力データオプションの設定

チャート型ウィジェットの出力データオプションを設定して、ユーザーが必要なデータを選択できます。

1. ウィジェットの右上にある  アイコンをクリックして、出力データのオプションを設定します。
 - プロジェクト、エージェント、アプリケーションなどを選択してデータをフィルタリングできます。
 - 個々のデータまたはマージされたデータを選択して、チャートに表示されるグラフのタイプを変更できます。
 -  >  を選択します。**チャート設定**画面で詳細オプションを変更できます。設定できるオプションは次

のとおりです。

- **チャートタイプ**：集計する現在の数値をチャートに表示するオプションです。
 - **y軸 チャート最大値**：Y軸にチャートの最大値を自動または固定で選択できます。
 - **水平線設定**：追加ボタンを選択し、しきい値または範囲を水平線として表示できます。複数を追加して適用できます。水平線を追加した後は、**適用**ボタンを選択してチャートに反映してください。
 - **チャートの最大値を表示**：チャートに表示されるグラフの最大値を表示します。
 - **サブチャート**：ウィジェットのメインチャートにサブチャートを追加できます。サブチャートの種類を選択し、レイアウトとサイズ、データのマージ有無を選択します。他のウィジェットにも適用する場合は、**すべてのウィジェットに適用**ボタンを選択します。サブチャートは、データをさまざまな方法で確認するとき便利です。
2. オプション設定がすべて完了したら、**チャート設定**画面の左上の×ボタンをクリックするか、**チャート設定**領域の外側をクリックします。

❗ ウィジェットのデータタイプによって選択できるオプションは異なる場合があります。

メトリクスウィジェットデータの変更

メトリクスウィジェットのデータ照会の条件を変更できます。

1. **Flexボード > ダッシュボードリスト**で、ウィジェットデータを変更するダッシュボードを選択します。
2. **編集モード**で、変更したいウィジェットの右上にある  アイコンをクリックします。
3.  アイコンをクリックすると、データ設定編集ウィンドウが表示されます。
4. 次を参照して設定を変更するには、**適用アイコンウィジェットに適用**ボタンをクリックします。
 - ウィジェットを作成すると、デフォルトで生成されるメトリクスのオプションがタブに表示されます。
 - メトリクス オプションには、**カテゴリ**と**フィールド**、**タグ**が含まれます。
 - **カテゴリ**項目は現在、変更を加える機能を制限しています。
 - **フィールド**、**タグ**項目は複数選択できます。選択したタグオプションでチャートデータをグループ化します。
 -  ボタンをクリックすると、元のデータを表示できます。

- オブジェクトのマージおよび時間マージ項目の詳細については、[データマージオプション](#)を参照してください。

データマージオプション

Flexボードのすべてのメトリクス一覧に配置されたマトリクスウィジェットでは、データのマージオプションを提供します。データのマージは、**オブジェクトのマージ**と**時間マージ**方法を提供します。

• オブジェクトのマージ

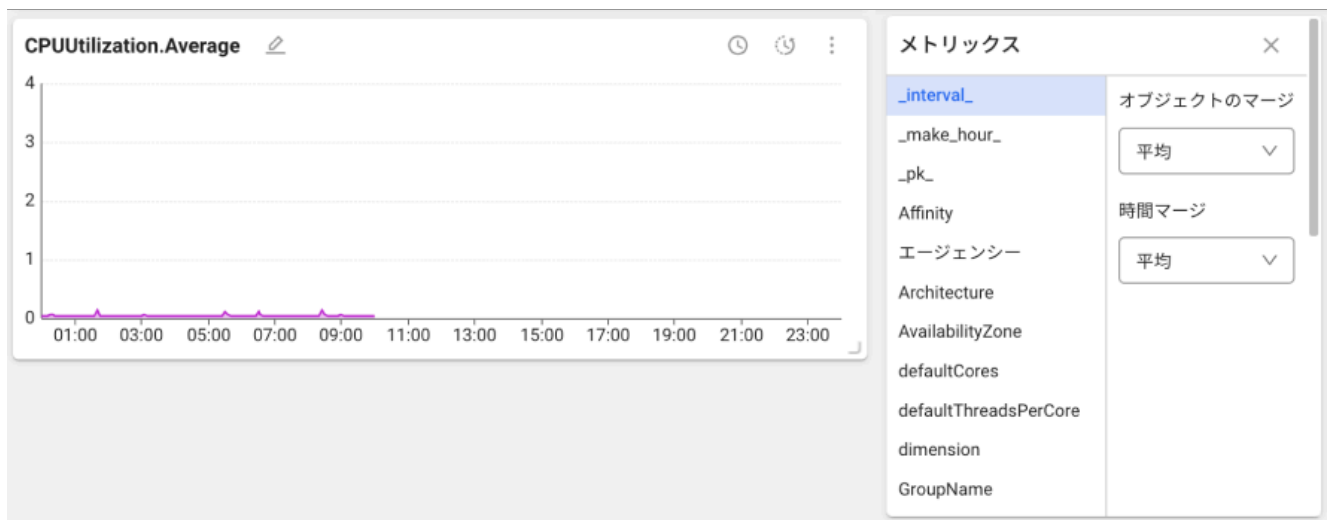
左側で選択したタグ(例：エージェント名)に基づいてチャートデータをグループ化するために使用します。この場合、異なるフィールド(例：CPU)値を持つデータ間でタグが一致する場合、そのデータは結合されます。

• 時間マージ

主に検索時間が長い場合に使用します。ソースデータとフィールド値が同じデータを時間範囲に基づいて一定の時間間隔(5分または1時間)でデータを結合します。この結合方法は、時間結合と呼ばれます。

• AWS CloudWatchメトリクス

AWSモニタリング指標の場合、統計手法はフィールド名の後にサフィックス(接尾辞)として追加されます。データマージの既定値は、このサフィックスを参照しています。



1. 編集モードですべてのメトリクスボタンを選択し、ウィジェットテンプレート一覧をすべてのメトリクス一覧に変更します。
2. メトリクスウィジェットをレイアウトに配置します。

3. メトリクス ウィジェットの右上にある  ボタンを選択します。
4. タグ(例: エージェント名)オプションを選択すると、データ結合オプションが表示されます。ウィジェットのサイズによっては、メトリクス オプションの位置が異なる場合があります。

ウィジェットのコピー

1. ウィジェットの右上にある  ボタンをクリックします。
2.  ボタンをクリックします。
3. コピー画面が表示されたら、**+ ウィジェット追加**を選択します。
4. 一覧にあるウィジェットの名前を変更します。
5. **適用**ボタンを選択します。
6. **編集モード**画面でコピーしたウィジェットで確認できます。

ウィジェットの削除

1. ウィジェットの右上にある  ボタンをクリックします。
2.  ボタンをクリックします。
3. 確認メッセージの画面が表示されると、**削除**ボタンを選択します。削除を取り消すには**キャンセル**ボタンをクリックします。

カウント推移の比較

ホーム画面 > プロジェクト選択 > 分析 > カウント推移比較

分析 > 推移比較 で、データベースの主要指標を時間帯ごとに確認し、パフォーマンスの変化を追跡します。

異なる日付の指標を比較することで、運用の傾向を分析できます。アクティブセッション テーブルでは、現在実行中のセッションと長時間実行中のセッションの両方を確認できます。

基本画面ガイド



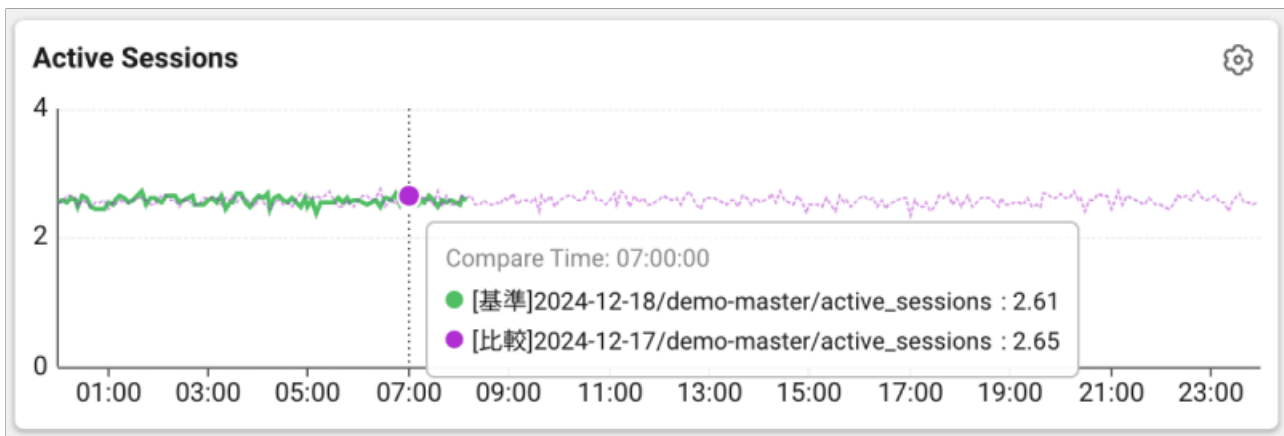
- ① 基準日、② 比較日、インスタンス項目を選択すると、すべてのウィジェットに選択した項目を基準に、データが自動的に反映されます。
- ウィジェットに表示されている指標の意味を知りたい場合は、名前の横の ⓘ ボタンを選択するか、[次の文書](#)を参照してください。

① ⓘ ボタンのツールチップ機能は多言語ではサポートされていません。

- 他のウィジェットの位置を変更する場合は、ウィジェットの上部を選択した状態でドラッグします。ただし、ウィジェットのサイズは変更できません。
- ③ アクティブセッションテーブルの左側には、**基準日**、右側には、**比較日**に該当するデータを表示します。
- ③ アクティブセッションテーブル一覧の文字の色は黒 → オレンジ → 赤の順でセッションの実行速度が遅いことを意味します。

比較対象の選択

① **基準日、インスタンス**を選択し、② **比較日、インスタンス**を選択します。チャートの上にマウスをオーバーすると表示されるツールチップを通じて、その時点の詳細データを確認できます。ツールチップの内容は、マウスの位置に応じて値が更新されます。



- シリーズごとに固有の色が表示され、視覚的に簡単に区別できます。実線は**基準日**のデータ、点線は**比較日**のデータを意味します。
- **Compare Time**は、現在マウスポインターが存在するデータの時間値を表示します。
- 選択したデータ時点のメトリクス値を表示し、シリーズごとに区別して提供します。各シリーズには、データポイントの日付とエージェント(インスタンス)名、指標名、値が含まれます。

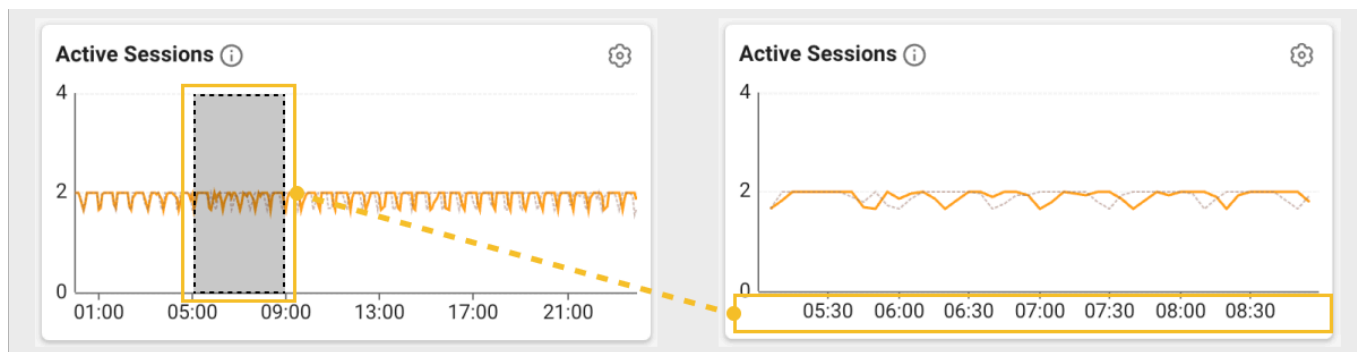
- ① • 画面上部の**データベース**項目で比較したいデータベースを選択することもできます。
- 画面上部の**データベース**項目の色は、グラフチャートで同じ色で表されます。

アクティブセッション情報の比較

基準 (2024-12-04)								比較 (2024-12-03)							
instance	pid	datname	cpuusage	client_addr	username	runtime ↓	st	instance	pid	datname	cpuusage	client_addr	username	runtime ↓	st
master	22204	dvdrental	0.00%	10.0.0.5	tom	1629.5	active	master	31155	dvdrental	0.00%	10.0.0.5	tom	2603.6	active
master	22205	dvdrental	0.00%	10.0.0.5	tom	1628.6	active	master	31154	dvdrental	0.00%	10.0.0.5	tom	2603.6	active
master	22207	dvdrental	0.00%	10.0.0.5	tom	1626.7	active	master	31156	dvdrental	0.00%	10.0.0.5	tom	2603.2	active
master	13571	postgres	0.00%	10.0.0.5	whatap	0	active	master	20391	postgres	0.00%	10.0.0.5	whatap	0	active
								master	20396	postgres	0.00%	10.0.0.5	whatap	0	active

- データを表示した時間帯は、**4**で確認できます。
- アクティブセッションデータは5秒単位で収集され、テーブル上の5秒単位で移動できるボタンを選択すると、その時間のデータを表示することができます。
- 右上にある◀または▶を選択すると、1分単位でデータを表示できます。5秒単位で移動するには、◀または▶ボタンを選択します。
- アクティブセッションテーブル一覧の文字の色は黒 → オレンジ → 赤の順にセッションの実行速度が遅いことを意味します。
- グラフチャートの特定の時点をクリックすると、**(3)**その時点を赤い線で表示して収集されたアクティブセッションを確認できます。

特定時間の比較



特定の時間を拡大してデータを比較できます。画面に配置されたウィジェットから目的の時間帯を選択して、チャートをドラッグしてください。すべてのウィジェットとアクティブセッションテーブルに、ドラッグした特定の時間帯の

データを表示します。

プリセット

インスタンスモニタリングメニューから、ユーザーが設定したウィジェットの設定とアクティブセッションテーブルの配置状態をプリセットとして読み込むことができます。

❗ プリセット設定の詳細については、[次の文書](#)を参考にしてください。

カラム情報ガイド

- Active session Lock tree Process information

項目	説明
db	データベース名
user	SQL実行アカウント
application	クライアントアプリケーション名
client_host	接続されたクライアントのホスト名(client_addrの逆引きDNS参照で報告されます)
client_addr	クライアントのIPアドレス
client_port	クライアントのポート番号(Unixソケットを使用する場合は-1)
query_hash	クエリのハッシュ値(WhaTap内部管理値)
query_start	現在アクティブ状態のクエリが開始された時間(アクティブ状態でない場合、最後にクエリが開始された時間)
state	バックエンドプロセスの現在の状態(active/idleなど)

項目	説明
state_change	stateが変更された最近の時間
xact_start	そのプロセスの現在のトランザクションが開始された時間(アクティブTXがない場合は null)
backend_start	そのバックエンドプロセスが開始された時間
backend_xmin	バックエンドプロセスが処理中の xid の最小値
backend_type	バックエンドプロセスのタイプ、 autovacuum worker、 logical replication launcher、 client backend、 checkpointer など
backend_xid	バックエンドが使用中の xid
wait_event_type	バックエンドプロセスが待機中のイベント type
wait_event	バックエンドプロセスが現在待機中の場合は、待機イベント名

ⓘ WhaTapは、クライアントに関する情報をデフォルトで保存します。

項目	説明
Instance	インスタンス（エージェント）名
pid	セッションID
lock_mode	ホルダーセッションのロックモード - RowExclusiveLock：INSERT、UPDATE、DELETE - AccessExclusiveLock：DDL（ALTER、DROP、TRUNCATE など）によるロック - AccessShareLock：SELECT 時に使用 ほとんどすべてのロックと共存可能 - ShareLock：LOCK TABLE ... IN SHARE MODE
holder_relation	ロックを保有しているセッションがアクセス中のテーブル名またはオブジェクト識別子
waiter_type	ロックがどの対象に対してかかっているかを示します

項目	説明
	- relation：テーブルにかかっているロック tuple：特定の行（row）にかかっているロック - transactionid：トランザクションにかかっているロック
waiter_relation	ロックを待機しているセッションがアクセス中のテーブル名またはオブジェクト識別子
lock_request	ロックを待機しているセッションのロックモード lock_mode を参照

次の項目は、XOSエージェントをインストールした場合に収集される指標です。

項目	説明	単位
pid	プロセスID	-
cpuusage	プロセスが使用するCPU使用率	%
cputime	プロセスがこれまでに使用した累積CPU時間	秒
elapsed	プロセスが実行されてからの経過時間	秒
vsize	Virtual memory size. 全体の仮想メモリ使用量（RAM + Swap + 予約）	KB
rss	Resident Set Size. 現在のプロセスが占有している実際の物理メモリサイズ	KB
pss	Proportional Set Size. 共有メモリを含む実際の物理メモリ使用量	KB
state	プロセスの現在の状態コード（例：R、S、D、T、Z など）	-
ioread	累積読み込みバイト数（read_bytes）	バイト（B）
iowrite	累積書き込みバイト数（write_bytes）	バイト（B）
uid	このプロセスを実行したユーザーID	-
cmd	実行されたコマンドの短縮形式	-

項目	説明	単位
longcmd	実行されたコマンドの完全な形式（フルパスおよび引数を含む）	-

ロックツリー

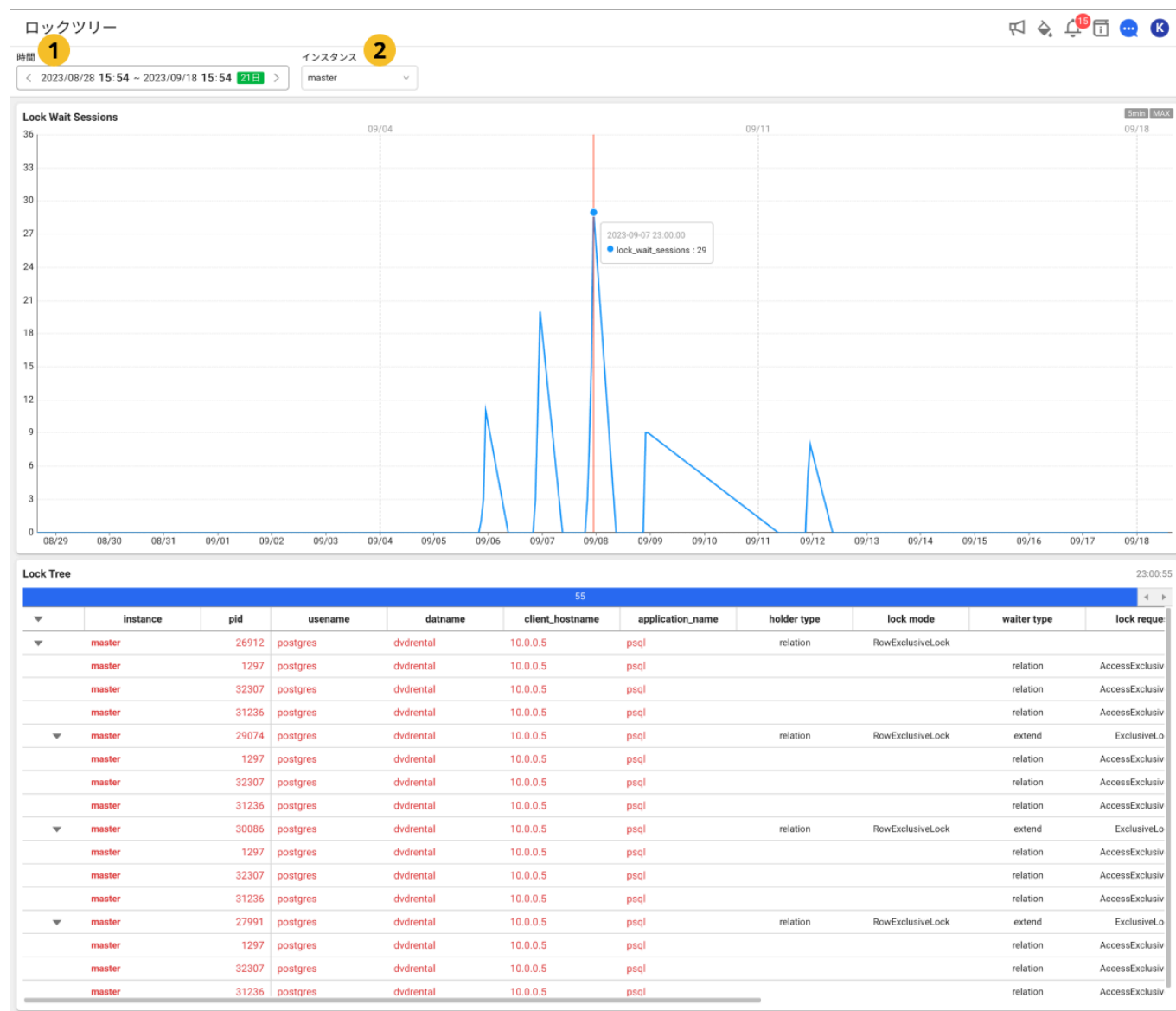
ホーム画面 > プロジェクト選択 > 分析 > ロックツリー

一日の間に発生したロック(lock)の推移を分析するツールです。この機能によってロックを発生させたセッション(lock holder)と待機しているセッション(lock waiter)に対するツリー構造を視覚的に確認できます。特定時点で発生したロック情報のholderとwaiter間の関係を分析できます。WhaTapの**ロックツリー**機能により、データベース管理者はロック関連の問題を迅速に識別して解決でき、データベースの性能を最適化できます。

主な機能は次のとおりです。

- **ロック推移分析:** 設定した照会期間に発生したロックの推移を時間とともに追跡し、グラフで表示します。これにより、ロックがどのように発生し、解消されるかを視覚的に把握できます。
- **ロックHolderおよびWaiter確認:** 各ロックに対するholderとwaiterをツリー構造で確認できます。これにより、特定のロックが発生した状況でロックを発生させたセッションと待機するセッションとの関係を把握できます。
- **セッション間の関係分析:** ロックを発生させたセッションと待機するセッションとの関係を分析し、クエリー実行およびトランザクション処理過程で発生したロックの原因を識別できます。
- **時間別ロックトレンドの視覚化:** 1-Min Lock TrendsとLock Wait Sessions機能により、ロック発生データを時間単位で分析できます。

基本的な使い方



① 時間、② インスタンスで照会する日付と時間、インスタンスを選択します。設定した時間にロックが発生した場合は、Lock Wait Sessionsと下段のLock Treeテーブルに照会されているデータが表示されます。

- 1つのチャート内で最大2週間のデータを一度に確認できます。グラフチャートで特定の時間をクリックすると、画面下のテーブルから選択した時間のロック情報を確認できます。画面下の詳細情報から秒単位ボタンをクリックす

ると、5秒単位でロック情報を照会できます。

❗ **Lock Wait Sessions**チャートの右上に表示されているタグを通じて照会の単位を確認できます。

- グラフチャートで特定時間領域をドラッグ(ドリルダウン)するほど、詳細時間のチャートとロックツリーを照会できます。
- チャートで3時間以内にドラッグして、5秒単位のデータを照会できます。ただし、5秒単位のデータは、最近1か月の範囲に限り照会できます。1か月前のデータは、5分要約データでしか閲覧できません。例えば、40日前のロックツリーデータをドラッグ(ドリルダウン)すると、5分要約データは照会できますが、5秒データは照会できません。

Lock Wait Sessions

Lock Wait Sessionsは、ロック待機セッションの情報を分析できる機能です。照会期間によってロック待機セッションデータを3時間以上照会すると5分間隔で、3時間未満照会すると5秒間隔で提供します。また、チャートをドラッグして時間範囲を流動的に調整できます。

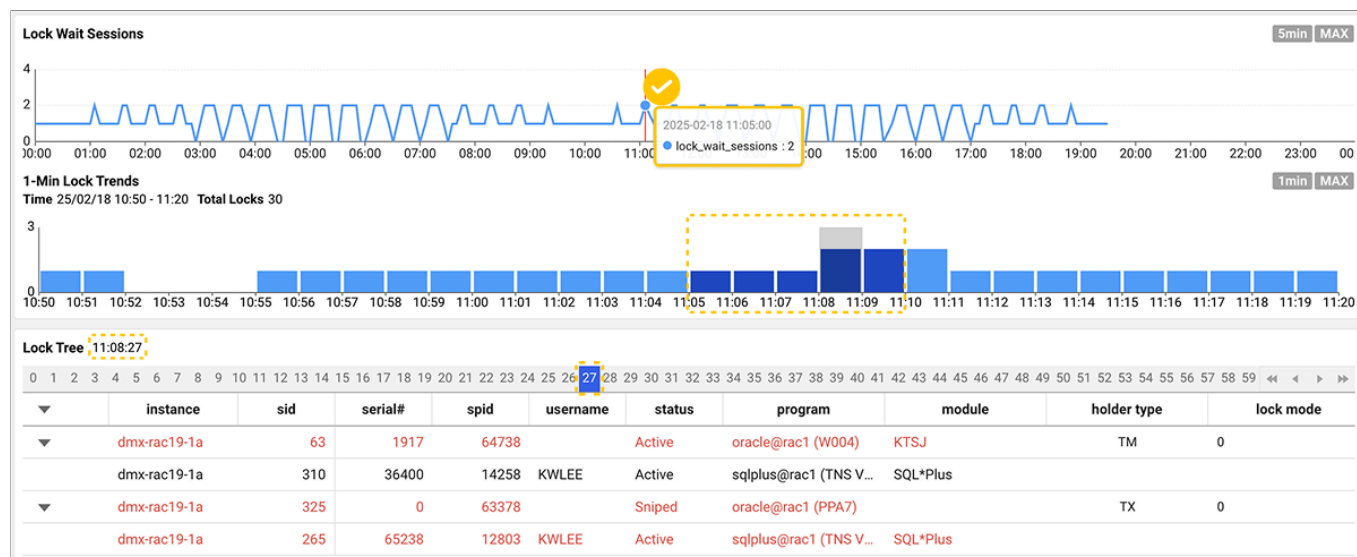
1-Min Lock Trends

Lock Wait Sessionsで特定の時間帯をクリックすると、その時間基準で15分前から15分後までの計30分間のロック発生状況を可視化して表示します。これにより、30分単位でロック発生トレンドを効果的に分析できます。

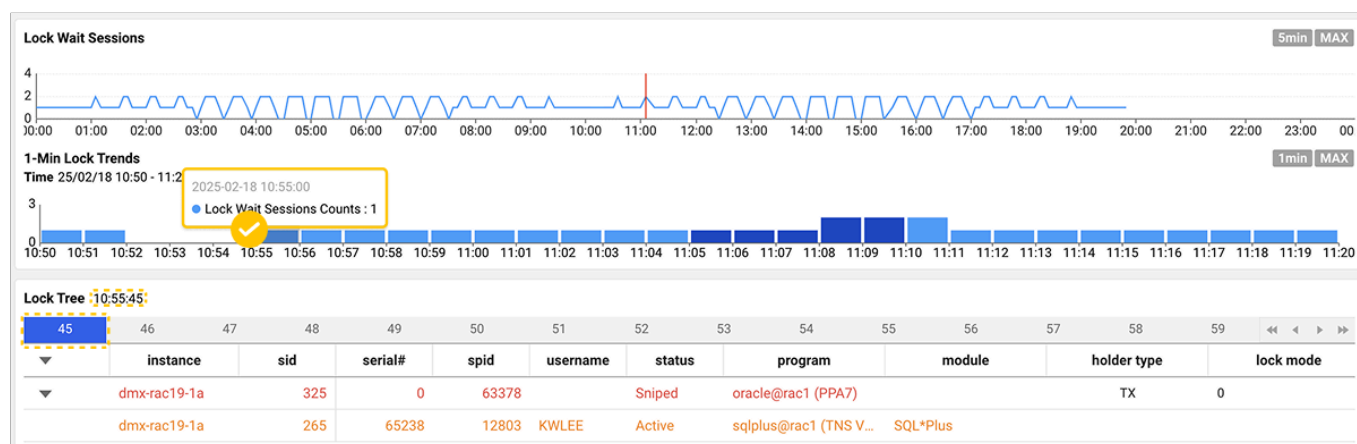
❗ **Total Locks**は、その時間帯に発生したロックの合計個数を意味します。

時間照会

Lock Wait Sessionsでクリックした時間に発生した最大ロック数を基準に**1-Min Lock Trends**と**Lock Tree**テーブルの時間が設定されます。これにより、ロックが最も多く発生した区間を把握できます。



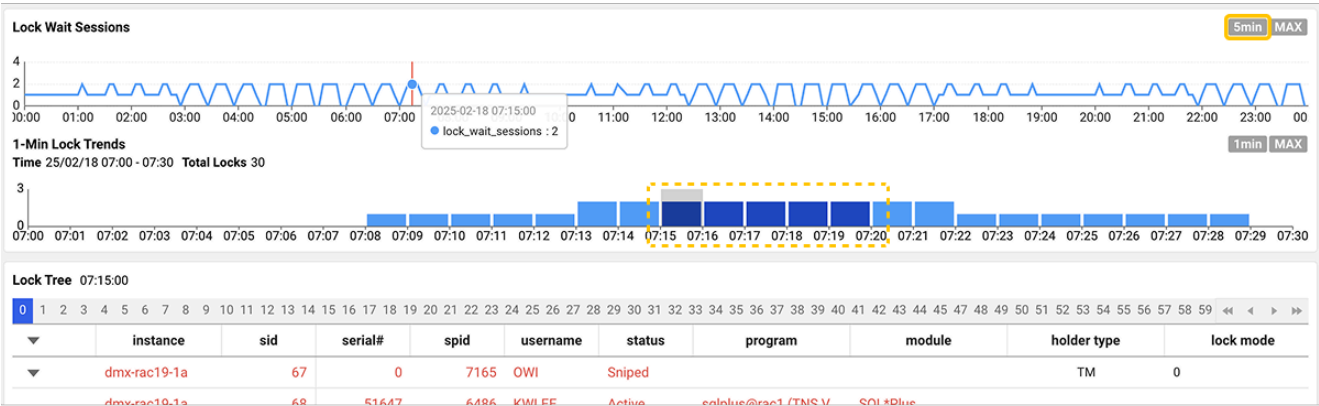
Lock Wait Sessionsはそのままにして1-Min Lock Trendsチャートで他の時間帯のバーをクリックした場合、該当する分の最大ロック数を持つ秒の時点でLock Treeテーブルの時間範囲が同期されます。これにより、時間帯を変更する時にLock Treeテーブルに反映された情報を確認できます。



・3時間以上照会した場合 (5分単位 of データ使用)

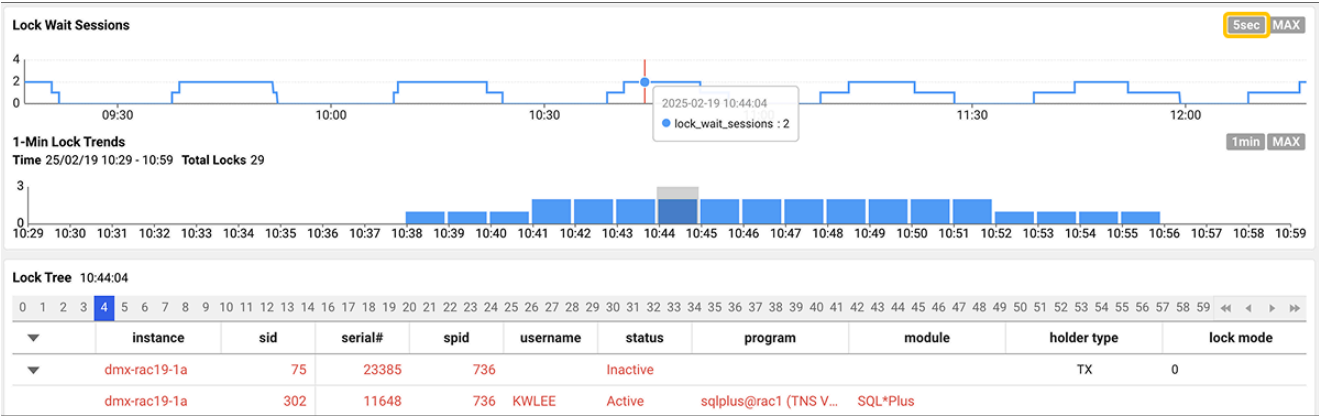
照会時点から5分の区間でバーの色が濃く表示されます。

07:15:00 時点をクリックした場合、07:15:00~07:20:00 区間バーのチャートの色が濃く表示されます。



3時間未満照会した場合 (5秒単位 of データ使用)

短い時間間隔で発生したロック状況を分析する時に役立ちます。



カラム情報ガイド

- Active session Lock tree Process information

項目	説明
db	データベース名
user	SQL実行アカウント
application	クライアントアプリケーション名

項目	説明
client_host	接続されたクライアントのホスト名(client_addr の逆引きDNS参照で報告されます)
client_addr	クライアントのIPアドレス
client_port	クライアントのポート番号(Unixソケットを使用する場合は-1)
query_hash	クエリのハッシュ値(WhaTap内部管理値)
query_start	現在アクティブ状態のクエリが開始された時間(アクティブ状態でない場合、最後にクエリが開始された時間)
state	バックエンドプロセスの現在の状態(active/idleなど)
state_change	stateが変更された最近の時間
xact_start	そのプロセスの現在のトランザクションが開始された時間(アクティブTXがない場合は null)
backend_start	そのバックエンドプロセスが開始された時間
backend_xmin	バックエンドプロセスが処理中のxidの最小値
backend_type	バックエンドプロセスのタイプ、autovacuum worker、logical replication launcher、client backend、checkpointer など
backend_xid	バックエンドが使用中のxid
wait_event_type	バックエンドプロセスが待機中のイベント type
wait_event	バックエンドプロセスが現在待機中の場合は、待機イベント名

ⓘ WhaTapは、クライアントに関する情報をデフォルトで保存します。

項目	説明
Instance	インスタンス（エージェント）名
pid	セッションID
lock_mode	ホルダーセッションのロックモード - RowExclusiveLock：INSERT、UPDATE、DELETE - AccessExclusiveLock：DDL（ALTER、DROP、TRUNCATE など）によるロック - AccessShareLock：SELECT 時に使用 ほとんどすべてのロックと共存可能 - ShareLock：LOCK TABLE ... IN SHARE MODE
holder_relation	ロックを保有しているセッションがアクセス中のテーブル名またはオブジェクト識別子
waiter_type	ロックがどの対象に対してかかっているかを示します - relation：テーブルにかかっているロック tuple：特定の行（row）にかかっているロック - transactionid：トランザクションにかかっているロック
waiter_relation	ロックを待機しているセッションがアクセス中のテーブル名またはオブジェクト識別子
lock_request	ロックを待機しているセッションのロックモード lock_mode を参照

次の項目は、XOSエージェントをインストールした場合に収集される指標です。

項目	説明	単位
pid	プロセスID	-
cpuusage	プロセスが使用するCPU使用率	%
cputime	プロセスがこれまでに使用した累積CPU時間	秒
elapsed	プロセスが実行されてからの経過時間	秒
vsize	Virtual memory size. 全体の仮想メモリ使用量（RAM + Swap + 予約）	KB

項目	説明	単位
rss	Resident Set Size. 現在のプロセスが占有している実際の物理メモリサイズ	KB
pss	Proportional Set Size. 共有メモリを含む実際の物理メモリ使用量	KB
state	プロセスの現在の状態コード（例：R、S、D、T、Z など）	-
ioread	累積読み込みバイト数（read_bytes）	バイト（B）
iowrite	累積書き込みバイト数（write_bytes）	バイト（B）
uid	このプロセスを実行したユーザーID	-
cmd	実行されたコマンドの短縮形式	-
longcmd	実行されたコマンドの完全な形式（フルパスおよび引数を含む）	-

セッション履歴

照会期間中に収集されたセッション情報を時系列で照会できる画面です。特定の時間帯にどのセッションがどのSQLを実行したか、各セッションのステータスとリソース使用量はどうかを分析できます。

- 5秒間隔で収集されたアクティブセッションのスナップショットをもとにデータを提供します。
- **セッション推移**チャートを通じて、時間帯別のセッション数の変化を視覚的に把握できます。
- フィールドパネルとフィルターを活用して、特定の条件のセッションのみを照会できます。

セッション履歴を照会する

上部フィルターで照会条件を設定し、フィールドパネルでセッションをフィルタリングした後、セッション推移チャートと一覧で時間帯別のセッション変化とステータスを確認できます。

基本オプション

上部フィルター領域で時間、対象インスタンス、フィルターなどを設定できます。

- **時間**: 照会時間範囲を選択できます。デフォルトで10分単位に設定されており、左右の矢印で移動可能です
- **インスタンス**: 照会対象のDBインスタンスを選択可能
- **フィルター**: フィールドごとに条件を追加してデータをフィルタリング。複数条件はANDロジックが適用されます
- **表示モード**: テーブルビューまたはツリービューに切り替え可能
- **ダウンロード**: CSV形式でエクスポート

フィールドパネル

画面左側のフィールドパネルから、フィルター条件を簡単に追加できます。

1. フィールド名をクリックすると条件入力ポップアップが表示されます。
2. 演算子と値を設定してから**適用**ボタンをクリックすると、上部フィルターに条件が追加されます。
3. 上部の検索バーでフィールド名を検索できます。フィルター条件を適用すると、条件を満たすセッションのみがチャートとテーブルに表示されます。

- 例えば `query` フィールドに特定のキーワードを含む条件を設定すると、そのキーワードが含まれるSQLを実行するセッションのみを照会できます。複数条件を適用する場合はANDロジックで動作します。

セッション推移チャート

照会期間中のアクティブセッション数とロック待機セッション数をバーチャートで表示します。X軸は時間、Y軸はセッション数です。最大1時間の範囲で照会できます。

- 全アクティブセッション数のうち、**ロック待機（Lock Wait）セッションは赤色**で区別して表示します。
- 特定の時間帯の総アクティブセッション数とそのうちロック待機セッションの割合を視覚的に把握できます。
- チャート領域をマウスドラッグすると、その区間にドリルダウンして詳細分析できます。
- チャートの特定の時間をクリックすると、その時間帯のアクティブセッションのみが下部テーブルに表示されます。

セッション一覧テーブル

収集されたセッション情報を時系列で表示します。下部のページネーションでページを移動でき、ページあたりの表示件数を設定できます。

カラムガイド

各カラムのサポート環境は、そのカラムのデータを照会できる条件を意味します。

- **PostgreSQL:** `pg_stat_activity` ビューから収集します。PostgreSQLのバージョンによって提供されるカラムが異なる場合があります。
- **XOS:** XOSエージェントを追加インストールした場合のみ照会できます。
- **共通:** DBのバージョンやエージェントのインストール有無にかかわらず、常に提供されます。

基本情報

カラム	説明	サポート環境
time	セッション情報が収集された時刻	共通
instance	DBインスタンス名	共通
pid	バックエンドプロセスID	PostgreSQL
datname	接続中のデータベース名	PostgreSQL
username	接続ユーザー名	PostgreSQL
client_addr	クライアントIPアドレス	PostgreSQL
client_hostname	クライアントホスト名	PostgreSQL
client_port	クライアントポート番号	PostgreSQL
application_name	クライアントアプリケーション名	PostgreSQL

実行情報

カラム	説明	サポート環境
state	セッションの現在ステータス（active、idle、idle in transactionなど）	PostgreSQL
runtime	現在のクエリの実行時間（秒）	PostgreSQL
xact_runtime	現在のトランザクションの実行時間（秒）	PostgreSQL
wait_event	現在待機中のイベント名	PostgreSQL
wait_event_type	待機イベントのタイプ（Lock、LWLock、IO、IPCなど）	PostgreSQL 9.6以上

カラム	説明	サポート環境
state_change	最後にステータスが変更された時刻	PostgreSQL
query	実行中のSQL文	PostgreSQL
query_param	SQLのバインドパラメーター値	共通
query_hash	SQLハッシュ値	共通

時間情報

カラム	説明	サポート環境
xact_start	現在のトランザクション開始時刻	PostgreSQL
query_start	現在のクエリ開始時刻	PostgreSQL
backend_start	バックエンドプロセス開始時刻	PostgreSQL

バックエンド情報

カラム	説明	サポート環境
backend_type	バックエンドプロセスタイプ (client backend、autovacuum workerなど)	PostgreSQL 10以上
backend_xid	バックエンドの現在のトランザクションID	PostgreSQL 9.4以上
backend_xmin	バックエンドのxmin horizon	PostgreSQL 9.4以上
query_id	クエリID	PostgreSQL 14以上

リソース使用量

カラム	説明	サポート環境
<code>cpuusage</code>	CPU使用率	XOS
<code>cputime</code>	CPU使用時間	XOS
<code>rss</code>	Resident Set Size（物理メモリ使用量）	XOS
<code>pss</code>	Proportional Set Size（比例メモリ使用量）	XOS
<code>vsize</code>	仮想メモリサイズ	XOS
<code>ioread</code>	I/O読み取り量	XOS
<code>iowrite</code>	I/O書き込み量	XOS

データ解釈ガイド

主な確認ポイント

カラム	意味	解釈ポイント
<code>state</code>	セッションの現在ステータス	active: SQL実行中、 idle in transaction: トランザクション内アイドル状態 → 長時間維持するとロックの原因になる可能性
<code>runtime</code>	現在のクエリの実行時間	値が高いほど長時間実行中のクエリ → チューニング対象
<code>xact_runtime</code>	現在のトランザクションの実行時間	値が高いほど長時間 持中のトランザクション → ロック競合、vacuum遅延の原因になる可能性

カラム	意味	解釈ポイント
wait_event	待機中のイベント	Lock系: ロック競合発生、IO系: ディスクI/Oボトルネック
cpuusage	CPU使用率	高い場合はCPU集約型クエリ → クエリの最適化が必要

異常セッションの識別

- runtime が異常に高いセッション: 長時間実行中のクエリ → 実行計画の確認、インデックスチューニング
- state がidle in transactionで xact_runtime が長いセッション: トランザクションが開いたままアイドル状態 → 他のセッションのロック待機原因、vacuum遅延を引き起こす
- wait_event がLock系のセッションが多数: 特定のテーブルやRowへのロック競合 → トランザクションパターンの確認
- wait_event_type がIOのセッション: ディスクI/Oボトルネック → クエリの最適化、shared_buffersなどのメモリ設定の確認

実践活用シナリオ

シナリオ1. 特定の時間帯のセッション急増分析

1. セッション推移チャートでセッション数が急増した区間を確認
2. その時間帯のセッション一覧で state 、 wait_event を確認
3. 同じ query が繰り返されているか、特定の client_addr から集中しているかを分析
4. フィルターを活用して state または wait_event_type 基準でセッションを分類

シナリオ2. 長時間実行セッションの追跡

1. フィールドパネルで runtime フィルターを追加して長時間実行セッションを照会
2. そのセッションの query を確認し、どのSQLが長時間実行されているかを把握
3. wait_event を確認してクエリ実行中かロック待機中かを判断
4. 必要に応じてそのSQLの実行計画 (Plan) を確認

シナリオ3. idle in transactionセッションの確認

1. フィールドパネルで `state` フィルターに「idle in transaction」条件を追加
2. `xact_runtime` が長いセッションを確認し、トランザクションがどれだけ長く開いているかを把握
3. そのセッションの `application_name` と `client_addr` で原因アプリケーションを特定
4. 長時間維持されるトランザクションは他のセッションのロック待機とvacuum遅延を引き起こす可能性があるため注意

シナリオ4. ロック競合原因の分析

1. セッション推移チャートで赤色（ロック待機）の割合が高い区間を確認
2. フィールドパネルで `wait_event_type` フィルターに「Lock」条件を追加
3. 待機中のセッションの `query` と `datname` を確認して競合対象を把握
4. `xact_runtime` が長いセッションがロックを保持していないかを確認

追加機能

フィルターの使い方

1. フィルター追加ボタンをクリックするか、フィールドパネルでフィールド名をクリックします。
2. フィルターキーを選択します。
 - 文字列 (String) : `=` (含む)、`!=` (含まない)
 - 数値 (Number) : `==` (等しい)、`>=` (以上)、`<=` (以下)
3. 値を入力して適用します。
4. 複数条件はANDロジックを適用します。

テーブルカラム設定

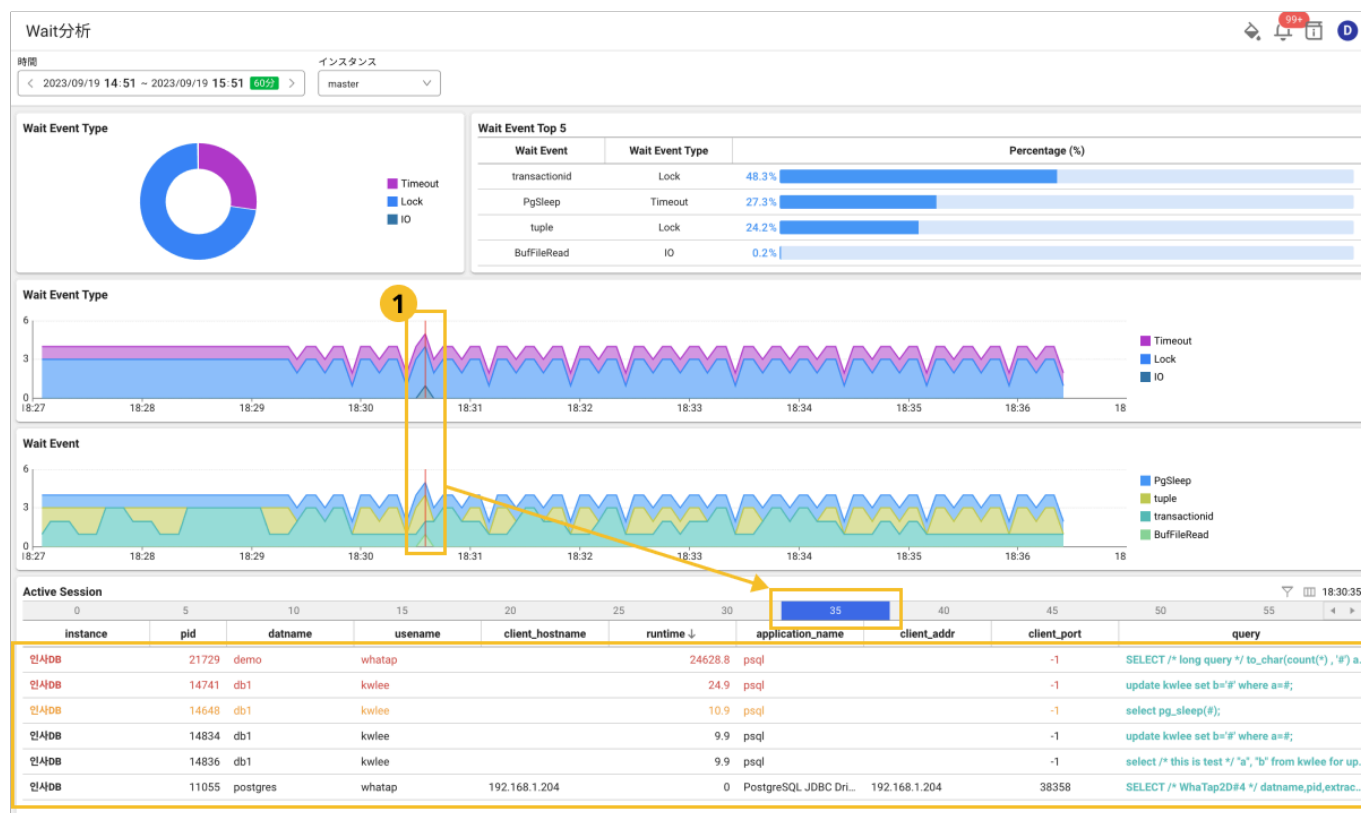
テーブルヘッダーのカラム選択機能を通じて表示するカラムを追加/削除し、ドラッグで順序を変更できます。

Wait分析

遂行中のセッションが何の待機イベントで待機しているのか、頻繁に発生する待機イベントはどのようなのかを確認することができます。5秒ごとに収集したアクティブセッションを待機(Wait)の観点から分析することができます。

- ① この機能は、DBXエージェント1.6.10バージョン以降に対応しています。
- 収集した待機情報は、`pg_stat_activity` ビューの `wait_event` と `wait_event_type` です。該当指標の説明は、[次の文書](#)を参照してください。

基本画面ガイド



照会する時間とインスタンスを選択します。設定した時間とインスタンスに該当する待機情報を確認することができます。照会期間中に待機(遅延)が頻繁に発生したアクティブセッションがあったら、どのようなタイプの待機イベント

が発生したのかを分析するのに活用できます。

Wait Event Top 5		
Wait Event	Wait Event Type	Percentage (%)
transactionid	Lock	48.3%
PgSleep	Timeout	27.3%
tuple	Lock	24.2%
BufFileRead	IO	0.2%

WaitEvent Top5で最も多く発生した待機イベントを確認できます。上記のイメージを例に、最も多く発生した待機イベントのタイプは**Lock**です。グラフチャートから①詳細時間を選択すると、その時間のアクティブセッションを照会することができ、待機イベントが発生したSQL文を確認することができます。

- ① • **Active Session**セクションの一覧で**instance**カラム項目を選択すると、**インスタンスモニタリングメニュー**に移動します。 **インスタンスモニタリングメニュー**の詳細は[次の文書](#)を参照してください。
- **Active Session**セクションの一覧でセッション項目を選択すると、**セッション詳細画面**が表示されます。 **セッション詳細画面**についての詳細は、[次の文書](#)を参照してください。
- **Active Session**セクションの一覧で**query**カラム項目を選択すると、**SQL詳細**画面が表示されます。 **SQL詳細画面**の詳細については[次の文書](#)を参照してください。

テーブルデータのフィルタリング



1. テーブルの右上にある🔍ボタンを選択します。

2. テーブルのカラムヘッダー項目と条件を選択します。
3. 条件を入力してください。テキストボックスに条件を入力します。
4. 保存ボタンを選択します。

テーブルカラムの設定

テーブルヘッダーカラムを非表示にしたり、希望する項目を追加できます。カラムの順序を変更することもできます。 ボタンをクリックします。

- ❗ • 設定を完了した後は、**確認**ボタンを選択すると、設定事項がテーブルに反映されます。
- ③ 検索バーにテキストを入力し、カラム項目を検索できます。入力したテキストと一致するカラム項目が表示されます。
- 画像は、製品またはプロジェクト、メニューによって異なる場合があります。

カラムを追加する

① 一覧でテーブルヘッダーカラムに追加する項目を選択します。すべての項目を追加するには、**すべて選択**を選択します。

カラムを削除する

① 一覧から削除するカラム項目のチェックボックスのチェックを解除します。または② 一覧から削除する項目の右側の×ボタンを選択します。

カラムの順序の変更

② 一覧で順序を変更するアイテムをドラッグして、希望する位置に移動できます。

設定事項の初期化

変更事項をすべてキャンセルし、初期化するには、 **初期化**ボタンを選択します。

Vacuum分析

Vacuum分析は、Top オブジェクトで提供するBloating、Dead Tuple、Ageなどの情報をVacuum観点で視覚的に再構成した画面です。Summaryカード、Top 5チャート、Vacuum実行履歴などを通じて、インスタンスのVacuum状態を一目で把握できます。

- **Vacuum Health Summary**を通じて、インスタンスの全体的なVacuumの健全性を素早く把握できます。
- **Table Analysis**で、Dead TupleとBloat Sizeを基準に管理が必要なテーブルを識別できます。
- **Vacuum Execution History**を通じて、Vacuum実行パターンと履歴を時系列で確認できます。

❗ Vacuumとは？

PostgreSQLはMVCC（Multi-Version Concurrency Control）方式を使用するため、UPDATEやDELETEを実行しても既存のRowは即座に削除されず**Dead Tuple**として残ります。VacuumはこのようなDead Tupleをクリーンアップして、ディスク空間を再利用できるようにするメンテナンス作業です。

Vacuumが適切なタイミングで実行されないと、以下の問題が発生する可能性があります。

- **テーブルBloatの増加**: 不必要な空間が増え、ディスク使用量の増加とクエリパフォーマンスの低下が発生
- **Transaction ID Wraparound**: Transaction Ageが限界に達すると、データ損傷防止のためDBが自動的に書き込みを停止するセーフガードが作動

事前設定

DB設定

エージェントがスキーマ情報を照会できるように、モニタリングアカウントに権限を付与します。

```
-- スキーマアクセス権限の付与
GRANT USAGE ON SCHEMA {schema_name} TO {DB_User};

-- 既存テーブルへの照会権限の付与
GRANT SELECT ON ALL TABLES IN SCHEMA {schema_name} TO {DB_User};
```

```
-- 今後作成されるテーブルへの照会権限の自動付与
```

```
ALTER DEFAULT PRIVILEGES FOR USER {owner_name} IN SCHEMA {schema_name} GRANT SELECT ON TABLES TO {DB_User};
```

構文	説明
GRANT USAGE ON SCHEMA	スキーマにアクセスできる権限を付与します。
GRANT SELECT ON ALL TABLES IN SCHEMA	スキーマ内の既存テーブルへの照会権限を付与します。
ALTER DEFAULT PRIVILEGES ... GRANT SELECT	今後そのスキーマに新しく作成されるテーブルにも自動的に照会権限が付与されます。

エージェント設定

whatap.confで以下の設定を確認します。pg_object のデフォルト値は true のため、別途設定なしで収集されます。

```
pg_object=true
```

設定項目	タイプ	デフォルト値	説明
pg_object	boolean	true	Top オブジェクトの情報収集有無を設定します。デフォルト値は true です。
pg_object_hour	int	5	Top オブジェクトの情報を収集する時刻を設定します。デフォルト値は 5 で、午前5時に収集を開始します。

データ収集ソース

Vacuum分析画面のデータはPostgreSQLのシステムカタログと統計ビューから収集されます。

セクション	主なソース	備考
Dead Tuples	<code>pg_class</code> 、 <code>pg_namespace</code>	<code>pg_stat_get_live_tuples()</code> 、 <code>pg_stat_get_dead_tuples()</code> 関数で Tuple 数を照会します。
Top Bloat Size	<code>pg_class</code> 、 <code>pg_stats</code> 、 <code>pg_stat_user_tables</code>	カラム統計（ <code>null_frac</code> 、 <code>avg_width</code> ）に基づいて Bloat サイズを推定計算します。
Transaction Age	<code>pg_catalog.pg_database</code> 、 <code>pg_catalog.pg_settings</code>	<code>age(datfrozenxid)</code> でデータベース別 Transaction Age を計算し、 <code>autovacuum_freeze_max_age</code> 設定と比較します。
Vacuum Execution History	<code>pg_stat_progress_vacuum</code> 、 <code>pg_stat_activity</code>	実行中の Vacuum の進行状態、待ちイベント、クエリ情報を照会します。

Vacuumを分析する

Vacuum 関連指標に基づいてテーブルの状態とメンテナンスの必要性を分析できます。主な指標は以下の通りです。

基本オプション

上部フィルター領域で時間、対象インスタンスを設定できます。

- **時間:** 照会する日付を選択します。
- **インスタンス:** 照会対象の DB インスタンスを選択します。

Vacuum Health Summary

インスタンスの Vacuum 関連の重要指標を3つの Summary カードで提供します。Dead Tuples と Top Bloat Size は **1日**に **1回** 収集され（`pg_object_hour` 設定時刻基準）、カードタイトル横の括弧内に収集時刻が表示されます。下部の Table Analysis データも同じ時刻に収集されます。

カード	説明
Dead Tuples	全Dead Tuple数とデータベース別分布を表示します。
Top Bloat Size	最大のテーブルBloatサイズ（MB）とデータベース別分布を表示します。
Transaction Age	最も古いトランザクションのAgeと全体限界に対する割合（%）を表示します。データベース別の割合も確認できます。

Dead Tuples

Dead Tuple数が高い場合、Vacuumが適切なタイミングで実行されていない可能性があります。データベース名をクリックすると、下部の**Table Analysis**のDead Tuple Top 5が該当データベースでフィルタリングされます。

⚠ Dead Tuplesが高い場合

- VACUUM テーブル名; を実行
- Dead Tuplesはクエリパフォーマンスの低下とディスク使用量増加の原因になります。
- 設定でAutovacuumの有効化の有無、閾値が高すぎないかを確認してください。

Top Bloat Size

Bloatとは、Dead Tupleなどによってテーブルが実際のデータに対して過度に大きい状態を意味します。Bloatが大きいと不要なディスク空間を占め、シーケンシャルスキャンのパフォーマンスが低下します。データベース名をクリックすると、下部の**Table Analysis**のBloat Size Top 5が該当データベースでフィルタリングされます。

⚠ Bloat Sizeが高い場合

- VACUUM FULL テーブル名; を実行
- VACUUM FULLはテーブルロックが発生するため、業務時間外での実行を推奨します。
- Bloatがひどいとクエリパフォーマンスの低下、インデックス非効率、ディスク浪費が発生します。
- Autovacuumの設定、UPDATE/DELETEパターン、HOT Update可能かどうか合わせて確認してください。



- `pg_repack` を使用するとテーブルロックなしで空間回収が可能です。

Transaction Age

PostgreSQLのTransaction IDは約20億（2 billion）個を使用できます。Transaction Ageがこの限界に近づくとXID Wraparoundが発生し、データ損傷防止のためDBが強制終了される可能性があります。

- エージェントバージョンv2.59.01以上が必要で、5秒間隔で収集されます。



Transaction Ageが高い場合

- `VACUUM FREEZE` データベース名; を実行
- Transaction IDが20億（2 billion）を超えるとXID wraparoundが発生し、データ損傷防止のためDBが強制終了される可能性があります。
- 必ず原因を把握して解決してください。
 - 長時間実行されているトランザクション
 - 未処理のReplication Slot
 - Disabled autovacuumなど

閾値設定

画面右上の閾値設定ボタンをクリックすると、Dead TuplesとBloat Sizeの閾値を設定できます。閾値を設定すると、現在設定された値が上部に表示され、閾値を超えたカードは背景色が変更されて異常状態をすぐに認識できます。設定を解除するには、閾値設定ウィンドウで設定解除ボタンをクリックします。

指標	設定項目	説明
Dead Tuples	Warning ≥	警告レベルのDead Tuple数
	Critical ≥	危険レベルのDead Tuple数
Bloat Size (GB)	Warning ≥	警告レベルのBloatサイズ（GB）

指標	設定項目	説明
	Critical ≥	危険レベルのBloatサイズ（GB）

- **Warning**超過時：カードの背景が**オレンジ色**に変更されます。
- **Critical**超過時：カードの背景が**赤色**に変更されます。

Table Analysis

テーブル単位でVacuum関連指標を分析できるTop 5チャートを提供します。

Dead Tuple Top 5

Dead Tupleが最も多い上位5つのテーブルを水平棒グラフで表示します。テーブル名は `database.schema.table` 形式で表示されます。

- Dead Tupleが多いテーブルはVacuumが必要な状態です。
- フィルターおよびテーブルビュー切り替えボタンで詳細データを確認できます。

Bloat Size Top 5

Bloat Sizeが最も大きい上位5つのテーブルを水平棒グラフで表示します。単位はバイトです。

- Bloat Sizeが大きいテーブルは `VACUUM FULL` または `pg_repack` などで空間を回収することを検討できます。
- フィルターおよびテーブルビュー切り替えボタンで詳細データを確認できます。

Vacuum Execution History

選択した期間中のVacuum実行履歴を時系列チャートで表示します。

- Vacuumが定期的に実行されているか、特定の時間帯に集中して実行されているかのパターンを把握できます。
- Vacuum実行がない区間が長い場合は、Autovacuumの設定を確認する必要があります。

Vacuum & Analyze History

チャートで特定の時間帯をクリックすると下部に**Vacuum & Analyze History**テーブルが表示され、該当時間帯（1時

間単位) に実行されたVacuumおよびAnalyze作業の詳細を確認できます。

カラム	説明
Time	Vacuum/Analyze実行時刻
Database	対象データベース名
Phase	Vacuum実行フェーズ (例: scanning heap、vacuuming indexesなど)
PID	Vacuumを実行したプロセスID
Query	実行されたVacuum/Analyzeクエリ
Heap Scanned	スキャンしたHeapブロック数
Heap Total	全Heapブロック数
Heap Vacuumed	Vacuum処理したHeapブロック数
Index Vacuum	インデックスVacuum実行回数
Max Dead Tuples	一度に処理できる最大Dead Tuple数
Dead Tuples	発見されたDead Tuple数
Runtime	Vacuum実行時間
Scanned %	全Heapに対するスキャン割合
Vacuumed %	全Heapに対するVacuum処理割合
Wait Event	待ちイベント名
Wait Event Type	待ちイベントタイプ

Configuration

現在のDBインスタンスに設定されているVacuum関連パラメータを参考用に表示します。PostgreSQLの様々なDBパラメータのうちVacuumに関連する項目のみを選別して表示するため、Autovacuumの動作やI/Oコスト設定をDB接続なしで直接確認できます。

AUTOVACUUM SETTINGS

パラメータ	説明
<code>autovacuum</code>	Autovacuumデーモンの有効化の有無
<code>autovacuum_max_workers</code>	同時に実行できるAutovacuumワーカープロセスの最大数
<code>autovacuum_naptime</code>	Autovacuum実行間隔（単位：秒）
<code>autovacuum_vacuum_threshold</code>	Vacuumをトリガーするための最小Dead Tuple数
<code>autovacuum_vacuum_scale_factor</code>	Vacuumトリガーのためのテーブルサイズに対するDead Tuple比率
<code>autovacuum_vacuum_cost_delay</code>	AutovacuumのI/Oコスト遅延時間（単位：ミリ秒）
<code>autovacuum_vacuum_cost_limit</code>	Autovacuumが一時停止する前までに蓄積できるI/Oコストの上限

Autovacuumは $\text{autovacuum_vacuum_threshold} + \text{autovacuum_vacuum_scale_factor} \times \text{テーブルRow数}$ 分のDead Tupleが発生すると、該当テーブルにVacuumを実行します。

I/O IMPACT SETTINGS

パラメータ	説明
<code>vacuum_cost_page_hit</code>	共有バッファでページが見つかった場合のコスト
<code>vacuum_cost_page_miss</code>	ディスクからページを読み取る必要がある場合のコスト
<code>vacuum_cost_page_dirty</code>	ページを変更（dirty）する際のコスト

データ解釈ガイド

主な指標の解釈

指標	正常	注意	危険
Dead Tuples	テーブルに対して少量	継続的に増加傾向	大量蓄積、Vacuum未実行
Bloat Size	テーブルサイズの20%以下	テーブルサイズの20～50%	テーブルサイズの50%超過
Transaction Age	5%以下	50%以上	80%以上 → 即時対応が必要

対処が必要な状況

- Dead Tupleが継続的に増加

Autovacuumがワークロードに追いつけていない → `autovacuum_vacuum_scale_factor` を下げる、`autovacuum_max_workers` を増やす

- Bloat Sizeが大きいテーブル

通常のVacuumでは空間がOSに返されない → `VACUUM FULL` または `pg_repack` を検討（注意： `VACUUM FULL` はテーブルロックが発生）

- Transaction Age比率が高い

Vacuumが古いトランザクションをクリーンアップできていない → 長時間維持されているトランザクションを確認、手動 `VACUUM FREEZE` の実行を検討

- Vacuum Execution Historyに空白区間

Autovacuumが無効化されているか、ワーカーが不足して実行されていない → Autovacuum設定を確認

Top オブジェクトを活用した分析

Vacuum分析はTop オブジェクトで提供するBloating、Dead Tuple、Age情報をVacuum観点で視覚的に再構成した画

面です。両画面は同じデータ（`pg_object_hour` 時刻に収集）を基にしていますが、分析目的に応じて異なる観点を提供します。

比較項目	Vacuum分析	Top オブジェクト
分析観点	Vacuumメンテナンス中心	オブジェクト（テーブル/インデックス）中心
データ範囲	Bloating、Dead Tuple、Age + Vacuum実行履歴	Bloating、Scan、DML、Analyze Time、Age、Dead Tupleなど6種類
表現方式	Summaryカード + Top 5チャート + 時系列履歴	ソート可能なテーブルリスト（Top N）
固有機能	Vacuum Health Summary、Vacuum実行履歴、閾値アラート、DBパラメータ照会	Scan/DML分析、Object Detail、インデックス分析

活用ガイド

- ✓ 全体的なVacuum状態の把握が目的であれば → Vacuum分析画面でSummaryカードとTop 5チャートで素早く現状を確認してください。
- 特定テーブルの詳細分析が必要であれば → Top オブジェクト画面でさまざまなタブとソート基準で細かく分析して💎💎💎ださい。
- Vacuumが必要なテーブルを識別した後の原因分析には、両画面を合わせて活用すると効果的です。

Topオブジェクト

個別インスタンスのオブジェクト（テーブル、インデックス）別のBloatingサイズ、Scan数、DML実行数、Dead tupleが多いオブジェクトを提供します。また、使用の多いオブジェクトを把握し、VacuumやAnalyzeを適切な時期に実行できるように情報を提供します。

Topオブジェクト

Bloating

Scan

DML

Analyze Time

Age

Dead Tuple

Table

Index

日付

インスタンス

照会基準

照会件数

<

2023-11-07

>

bloat_ratio

50

datname	schemaname	tablename	est_rows	table_size	bloat_size	bloat_ratio ↓	time
db1	public	whatap	42	24KiB	16KiB	67%	2023-11-07 16:25:40
db1	public	kwlee	3	16KiB	8KiB	50%	2023-11-07 16:25:40
demo	bookings	t1	4000000	253.3MiB	85.1MiB	34%	2023-11-07 16:25:40
demo	bookings	t2	4000000	253.3MiB	85.1MiB	34%	2023-11-07 16:25:40
demo	bookings	t3	4000000	253.3MiB	85.1MiB	34%	2023-11-07 16:25:40
demo	bookings	ticket_flights	1045650	68MiB	8.1MiB	12%	2023-11-07 16:25:40
demo	bookings	tickets	366733	48MiB	1.7MiB	4%	2023-11-07 16:25:40
demo	bookings	flights	33121	3MiB	24KiB	1%	2023-11-07 16:25:40
demo	bookings	bookings	262788	13MiB	8KiB	0%	2023-11-07 16:25:40
demo	bookings	seats	1339	64KiB	0	0%	2023-11-07 16:25:40
db2	public	db2_kwlee	2	8KiB	0	0%	2023-11-07 16:25:40
demo	bookings	airports_data	104	24KiB	0	0%	2023-11-07 16:25:40
demo	bookings	aircrafts_data	9	8KiB	0	0%	2023-11-07 16:25:40
demo	bookings	boarding_passes	579686	33.3MiB	48KiB	0%	2023-11-07 16:25:40

- 画面で照会したい日付とインスタンス、照会基準、照会件数を設定した上で🔍ボタンを選択します。選択した条件に応じた結果がテーブルに表示されます。
- 照会リストでtablenameまたはindexnameカラム項目を選択するとcolumn、index構成が確認できるObject detailウィンドウが表示されます。

Top Object

Bloating

Scan

DML

Analyze Time

Age

Dead Tuple

Table

Index

Date

Instance

Lookup criteria

Number of hits

< 2023-12-06 >

DBX-2-60-5444

bloat_ratio

50

datname	schemaname	tablename	
whatap	public	whatap	3000010
whatap	public	whatap3	10
whatap	public	whatap2	11
whatap	whatap	whatap	10
whatap	Test1	Test	3
whatap	whatap2	whatap	1
whatap	public	whatap5	2
whatap	public	whatap4	10
whatap	public	Whatap	1

Object Detail

whatap

type: table

seq_scan: 0

idx_scan: 0

n_live_tup: 0

n_dead_tup: 0

last_autovacuum:

last_analyze:

last_autoanalyze:

Column List

No	column_name	data_type	is_nullable	column_default	comment
1	a	integer	YES		whatap column a
2	b	character varying(20)	YES		

Index List

index_name	columns	type	is_unique	condition	comment
whatap_idx	a	BTREE	f		
whatap_b_idx	b	BTREE	f		

① Object detail機能は、DBXエージェントの1.6.15バージョン以降に対応します。データベース権限に関する設定は、[次の文書](#)を参考にしてください。

- ①
- Topオブジェクトについての情報は1日に1回収集します。関連するエージェント設定の詳細については、[次の文書](#)を参照してください。
 - データ照会のためには、DBモニタリングアカウントに次の権限が必要です。

```
grant select on all tables in schema {schema_name} to whatap;
```

- この機能は、DBXエージェント1.6.13バージョン以降に対応しています。

Bloating

Bloatingは実際に使用されていないtupleが増加してオブジェクトサイズが大きくなる現象です。

エージェント設定([whatap.conf](#))の `table[index]_bloat_ratio[bytes]` に設定された収集基準よりも大きな値を持つテーブルとインデックスをリストに表示します。該当するオブジェクトがないと、データが照会されない場合があります。

✓ チューニングTips

`bloat_ratio` が高いテーブルはVacuumの実行を検討してください。詳細については、次の文書を参照してください。

- [PostgreSQLテーブルの膨張現象への対処方法](#)
- [Dealing with significant Postgres database bloat — what are your options?](#)

次は**Bloating**を照会すると表示されるカラム項目です。

区分	カラム名	説明
Table	<code>datname</code>	データベース名
	<code>schemaname</code>	スキーマ(schema)名
	<code>tablename</code>	テーブル名
	<code>est_rows</code>	Dead tuple + Live tuple 件数で予測した値
	<code>table_size</code>	テーブルサイズ
	<code>bloat_size</code>	Dead tuple で膨らませた予想テーブルサイズ
	<code>bloat_ratio</code>	膨らませたサイズの比率です。
Index	<code>datname</code>	データベース名
	<code>schemaname</code>	スキーマ(schema)名
	<code>tablename</code>	テーブル名
	<code>indexname</code>	インデックス名
	<code>table_size</code>	テーブルサイズ
	<code>index_size</code>	インデックスサイズ

区分	カラム名	説明
	bloat_size	Dead tupleによって膨らませた予想インデックスサイズ
	bloat_ratio	膨らませたサイズ比率
	index_scans	インデックスを使用した場合、index scan回収

Scan

Seq_scan はインデックスを使用していないFull Scanを意味し、idx_scan はインデックスを使用した数を意味します。

✓ チューニングTips

Seq_scan が高いテーブルはインデックス作成を、idx_scan が低いインデックスは削除を検討してください。詳細については、[次の文書](#)を参照してください。

次はScanを照会すると表示されるカラム項目です。

区分	カラム名	説明
Table	datname	データベース名
	schemaname	スキーマ(schema)名
	tablename	テーブル名
	seq_scan	該当テーブルを順次スキャン(Full scan)した数
	seq_tup_read	順次スキャンから読取りされたライブ行数
	idx_scan	該当テーブルのインデックススキャン数
	idx_tup_fetch	インデックススキャンで読み取ったライブ行数

区分	カラム名	説明
Index	<code>datname</code>	データベース名
	<code>schemaname</code>	スキーマ(schema)名
	<code>tablename</code>	テーブル名
	<code>indexname</code>	インデックス名
	<code>idx_scan</code>	インデックススキャン実行数
	<code>idx_tup_fetch</code>	該当インデックスを使用してインデックススキャンから抽出された有効テーブル行数
	<code>idx_tup_read</code>	インデックススキャンに返されたインデックス項目の数

DML

`dml_count` 値が高いテーブルは、使用の多い主要なテーブルです。

✓ チューニングTips

使用するシステムの主要なテーブルを把握し、テーブルの変更やアーキテクチャ設計の際に参考にしてください。

次は**DML**を照会すると表示されるカラム項目です。

カラム名	説明
<code>datname</code>	データベース名
<code>schemaname</code>	スキーマ(schema)名
<code>tablename</code>	テーブル名

カラム名	説明
dml_count	n_tup_ins + n_tup_upd + n_tup_del
n_tup_ins	挿入(Insert)された行数
n_tup_upd	アップデート(Update)された行数
n_tup_del	削除>Delete)された行数
n_tup_hot_upd	HOTアップデートされた行数（例：別途のインデックスアップデートが不要）

Analyze Time

AnalyzeとVacuum実行の日付が古いオブジェクトの一覧です。

✓ チューニングTips

日付の古いテーブルは、統計情報が正しくないことがあります。実行を検討してください。詳細については、[次の文書](#)を参照してください。

次は**Analyze Time**を照会すると表示されるカラム項目です。

カラム名	説明
datname	データベース名
schemaname	スキーマ(schema)名
tablename	テーブル名
last_analyze	テーブルを手動分析した最後の時間
last_autoanalyze	autovacuumデーモンでテーブルを分析した最後の時間

カラム名	説明
<code>last_autovacuum</code>	autovacuumデーモンでテーブルをバキューム(vacuum)した最後の時間
<code>analyze_count</code>	手動で分析した回数
<code>last_vacuum</code>	テーブルが手動でバキューム(vacuum)した最後の時間(VACUUM FULLは除く)
<code>autoanalyze_count</code>	autovacuumデーモンで分析した回数
<code>autovacuum_count</code>	autovacuumデーモンでバキューム(vacuum)した回数
<code>vacuum_count</code>	手動でバキューム(vacuum)した回数(VACUUM FULLを除く)
<code>n_mod_since_analyze</code>	最後の分析(analyze)から変更された行数(推定値)

Age

PostgreSQLではXID(transaction id)を循環的に使用しているため、ある時点でXIDがラップアラウンド(Wraparound)されることがあります。この現象を防止するために、`xid_age` (Current XID - 作成時点のXID)が増え続けないように管理する必要があります。`autovacuum_freeze_max_age` を超えたらAnti-Wraparound Vacuumが自動的に行われるため、`autovacuum_freeze_max_age` 以下で `xid_age` を管理します。

この一覧は、以下の基準に該当するvacuum対象があるか照会します。対象がないと、データが照会されない場合があります。

- dead tupleの数がvacuum threshold(`autovacuum_threshold` + `autovacuum_scale_factor` * `number-of-tuples`) より多いテーブル
- `age( relfrozenxid )`が `autovacuum_freeze_max_age` より大きなテーブル

✓ チューニングTips

`xid_age` が増加し続けている場合は、現在設定されたAuto Vacuum条件としてXID整理作業が進まない可能性があります。auto vacuumパラメータを調整するか、手動でVacuumの実行を検討してください。

詳細については、[次の文書](#)を参照してください。

次は**Age**を照会すると表示されるカラム項目です。

カラム名	説明
<code>datname</code>	データベース名
<code>schemaname</code>	スキーマ(schema)名
<code>tablename</code>	テーブル名
<code>xid_age</code>	テーブルのmax age、最も古いレコードの <code>xid</code> (トランザクションID)
<code>per_to_wraparound</code>	<code>xid</code> (トランザクションID) ラッピングまでの残りの余裕をパーセンテージで表示した指標 計算式： $\text{per_to_wraparound}(\%) = \text{xid_age} / \text{Autovacuum_freeze_max_age} * 100$
<code>table_size</code>	テーブルサイズ
<code>autovacuum_vacuum_tuples</code>	この値よりDead tupleの数が多くなると、autovacuumを実行します。
<code>dead_tuples</code>	Dead tuple数
<code>autovacuum_freeze_max_age</code>	既定値は2億に設定

Dead Tuple

Dead tupleはDeleteやUpdateですでに削除された資料です。free spaceに変えるためにVacuum実行を検討します。

✓ チューニングTips

Dead tupleを減らすためにはVacuum実行を検討してください。詳細については、[次の文書](#)を参照してください。

次は**Dead Tuple**を照会すると表示されるカラム項目です。

カラム名	説明
<code>datname</code>	データベース名
<code>schemaname</code>	スキーマ(schema)名
<code>tablename</code>	テーブル名
<code>dead_tuple</code>	DeleteやUpdate等により使用しないタプル(Tuple)の数
<code>dead_tuple_ratio</code>	使用しないタプル(Tuple)の割合
<code>live_tuple</code>	使用するタプル(Tuple)の数
<code>live_tuple_ratio</code>	使用するタプル(Tuple)の割合
<code>total_relation_size</code>	リレーシヨンの全体サイズ、インデックスとTOASTデータを含む
<code>total_tuple</code>	全体タプル(Tuple)の数

エージェント設定

次はTopオブジェクトに関する情報を照会するためのエージェント設定です。 [whatap.conf](#)ファイルに必要なオプションを設定してください。

- **pg_object** Boolean

既定値 `true`

Topオブジェクト情報の収集有無を設定します。情報を収集するには `true` に変更してください。

- **pg_object_hour** Int

既定値 `5`

Topオブジェクト情報の収集時間を設定します。既定値は `5` で、午前5時に収集を始めます。

- **table_bloat_ratio** Percentage

既定値 `50`

テーブルbloating ratio値が設定値以上の場合は、情報を収集します。既定値は50%です。

- **table_bloat_bytes** Byte

既定値 10485760

テーブルbloatingbytes値が設定値以上の場合、情報を収集します。既定値は10MBです。

- **index_bloat_ratio** Percentage

既定値 50

インデックスbloatingratio値が設定値以上の場合、情報を収集します。既定値は50%です。

- **index_bloat_bytes** Byte

既定値 10485760

インデックスbloating bytes値が設定値以上の場合、情報を収集します。既定値は10MBです。

- **autovacuum_list_limit** Int

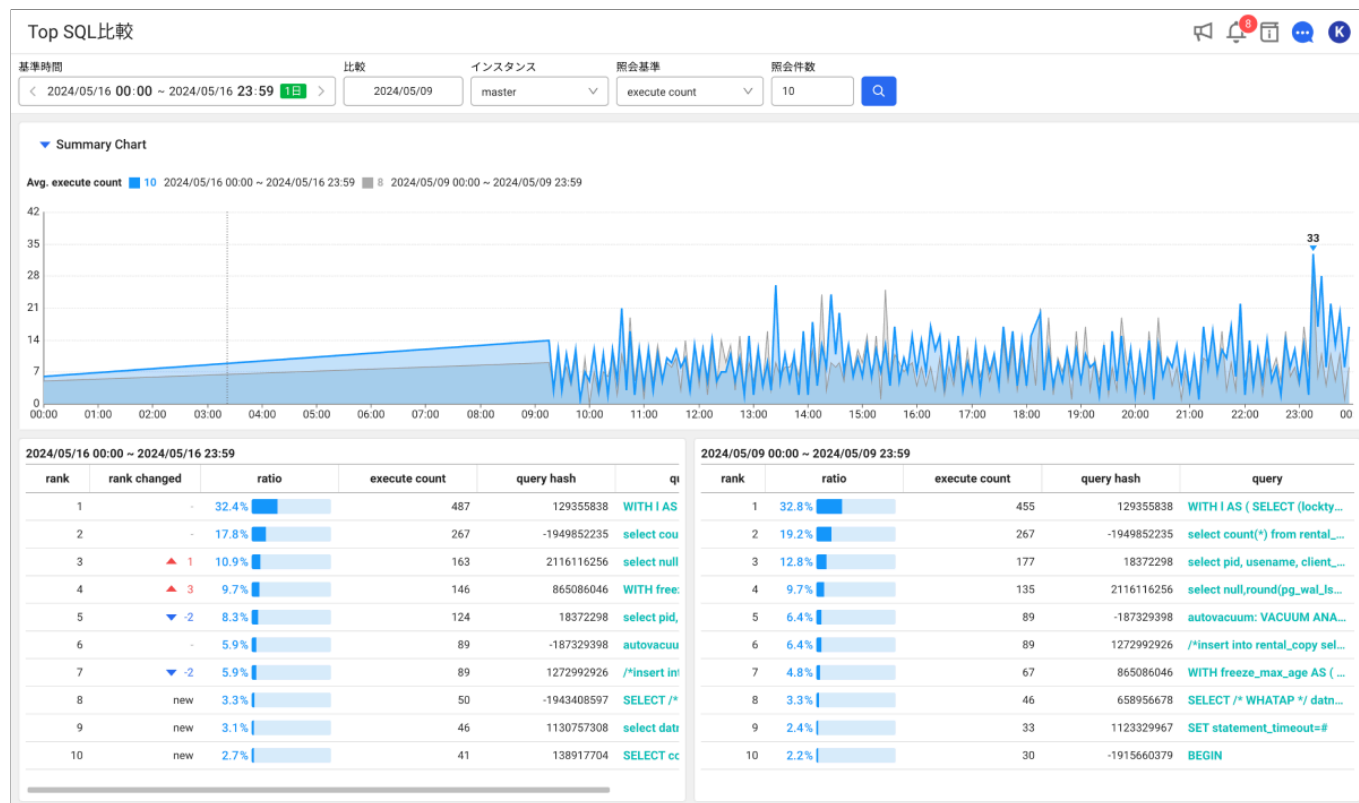
既定値 50

日付の古いage(relfrozenxid)順に設定値程のテーブル情報を収集します。既定値は50件です。

Top SQL比較

ホーム画面 > プロジェクト選択 > 分析 > Top SQL比較

特定の日付を基準に設定された時間帯のSQLの推移を比較できます。ユーザは複数の基準を選択してSQLの性能を比較し**Summary Chart**を通じて時系列グラフで確認できます。また、画面下のテーブルリストでは、基準日付と比較日付の**Top SQL**順位と騰落を比較できます。



- **SQL性能把握:** ユーザーは**Top SQL比較**メニューを通じてSQLの性能を簡単に把握できます。SQLの実行回数、最大経過時間、総経過時間、待機時間などを比較することで、システムの性能問題を迅速に感知できます。
- **時間別の推移確認:** **Summary Chart**を通じて時間によるSQLの性能変化を視覚的に確認できます。これにより、ユーザーはシステムの負荷が発生した時間帯や性能低下が発生した区間を素早く把握して対応できます。
- **Top SQL比較:** 画面下のテーブルリストでは、基準日付と比較日付の**Top SQL**順位を比較できます。これにより、ユーザーはSQLの性能変化を確認し、比較して性能改善に必要な措置を取ることができます。

Top SQL比較

画面の上でTop SQL比較のための条件を設定した後、 ボタンを選択します。

- **基準時間:** 比較のための基準時間を設定します。最大24時間まで設定できます。緑のボタンをクリックすると、照会時間を選択できるオプションが表示されます。ユーザーが希望する時間を直接設定するには、日付と時間エリアを選択します。

- ①
- 照会できる時間範囲はm、最大24時間です。
 - **基準時間**オプションの利用方法の詳細については。

- **比較:** 基準時間で設定した時間と比較する日付または時間を選択します。希望する条件を選択した後、**確認**ボタンを選択します。

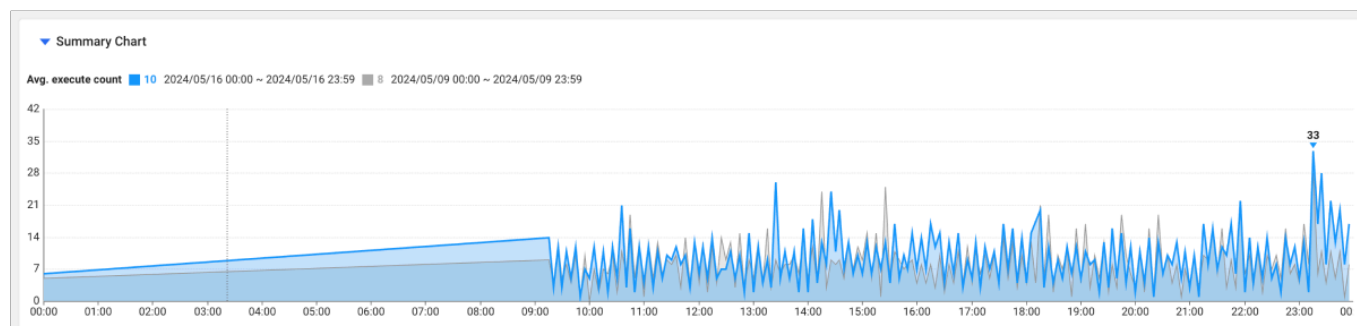
日付	時間
比較 2024/05/16 インスタンス master 日付 時間 1日前 2024/05/16 7日前 2024/05/10 30日前 2024/04/17 カスタマイズ 2024-05-16 キャンセル 確認	比較 2024/05/16 インスタンス master 日付 時間 (n) 時間前 1 キャンセル 確認

- **インスタンス:** SQL比較のためにインスタンス対象を選択します。
- **照会基準:** 比較基準を選択します。選択した項目に基づいてチャートを作成し、順位をソートします。
 - **excute count:** 実行回数に基づいて照会結果をソートします。
 - **elapsed max:** 最大経過時間に基づいて照会結果をソートします。

- **elapsed time:** 合計経過時間に基づいて照会結果をソートします。
- **elapsed wait:** 待ち時間に基づいて照会結果をソートします。
- **照会件数:** 画面の下にあるSQL順位テーブルのリストに表示される個数を設定してください。

Summary Chart

Summary Chartでは、基準日付と比較日付のSQL推移を比較できる時系列グラフを提供します。これにより、ユーザーは時間によるSQLの性能変化を容易に把握できます。



- ■ 青色のグラフ: 基準日付を意味し、右側の数値は平均値を意味します。
- ■ グレーのグラフ: 比較の日付を意味し、右側の数値は平均値を意味します。

SQL順位の比較

画面下のSQL順位テーブルでは、上位SQL一覧が表示されます。表示される個数は、画面上の**照会件数**で設定できます。

2024/05/16 00:00 ~ 2024/05/16 23:59

rank	rank changed	ratio	execute count	query hash	qi
1	-	32.4%	487	129355838	WITH I AS
2	-	17.8%	267	-1949852235	select cou
3	▲ 1	10.9%	163	2116116256	select null
4	▲ 3	9.7%	146	865086046	WITH free
5	▼ -2	8.3%	124	18372298	select pid,
6	-	5.9%	89	-187329398	autovacuu
7	▼ -2	5.9%	89	1272992926	/*insert ini
8	new	3.3%	50	-1943408597	SELECT /*
9	new	3.1%	46	1130757308	select datr
10	new	2.7%	41	138917704	SELECT cc

2024/05/09 00:00 ~ 2024/05/09 23:59

rank	ratio	execute count	query hash	query
1	32.8%	455	129355838	WITH I AS (SELECT (lockty...
2	19.2%	267	-1949852235	select count(*) from rental...
3	12.8%	177	18372298	select pid, username, client...
4	9.7%	135	2116116256	select null,round(pg_wal_ls...
5	6.4%	89	1272992926	/*insert into rental_copy sel...
6	6.4%	89	-187329398	autovacuum: VACUUM ANA...
7	4.8%	67	865086046	WITH freeze_max_age AS (...
8	3.3%	46	658956678	SELECT /* WHATAP */ datn...
9	2.4%	33	1123329967	SET statement_timeout=#
10	2.2%	30	-1915660379	BEGIN

左側に配置されたテーブルは、基準日付に該当するSQL順位が、右側には比較日付に該当するSQL順位が表示されます。基準日付テーブルの**rank changed**カラムでは、比較日付に比べて変化した騰落を確認できます。

例えば、ユーザーが**elapsed time**(経過時間)を基準に比較したとき、比較日付に比べて基準日付に同じSQL文の経過時間がさらに増えた場合は、当該SQL文の性能を検討する必要があります。クエリ最適化またはインデックスを使用して性能を最適化してください。

クエリプレビュー

クエリプレビュー

SQL順位テーブルで**query**カラム項目にマウスオーバーすると、該当SQL文のクエリを確認できます。

SQL 詳細情報の確認

詳細なSQL情報を確認するには、SQL順位テーブルの**query**カラム項目を選択します。**SQL詳細画面**が表示されます。SQLクエリ文とPlan情報を確認することができます。

SQL統計の表示 → ボタンをクリックすると、選択したSQLクエリに関連する統計情報を確認できる**SQL統計**に移動します。

query

```
START_REPLICATION ...
SELECT /* ... */
```

SQL詳細 SQL統計の表示 →

Query (hash: 1098839306) Default Bind Reset Bind ☒ Formatting Plan demo whatap Text JSON

```
SELECT
/* long query */
to_char(count(*), 'DD.MM.YYYY') as
when,
'n' as departure,
'c' as arrival,
's' as status,
'o' as seat_no
FROM
bookings.ticket_flights a,
bookings.boarding_passes b
union all
SELECT
to_char(f.scheduled_departure, 'DD.MM.YYYY') AS
when,
f.departure_city || ' ' || f.departure_airport || ' ' AS departure,
f.arrival_city || ' ' || f.arrival_airport || ' ' AS arrival,
f.status,
bp.seat_no
FROM
ticket_flights tf
JOIN flights_v f ON tf.flight_id = f.flight_id
LEFT JOIN boarding_passes bp ON tf.flight_id = bp.flight_id
AND tf.ticket_no = bp.ticket_no
WHERE
1 = 1
and tf.ticket_no = '0005432661915';
```

Rows	Node Cost	Plans
4	0%	Append
1	13.9%	Aggregate
606149825272	86.1%	Nested Loop Inner join
579686	0%	Seq Scan on boarding_passes as b
1045652	0%	Materialize
1045652	0%	Seq Scan on ticket_flights as a
3	0%	Nested Loop on ((tf.ticket_no = bp.ticket_no) AND (tf.flight_id = bp.flight_id)) Left join
3	0%	Nested Loop Inner join
3	0%	Nested Loop Inner join
3	0%	Nested Loop Inner join
3	0%	Index Only Scan on ticket_flights as tf on (tf.ticket_no = '0005432661915'::bpchar)
3	0%	Index Scan on flights as f on (f.flight_id = tf.flight_id)
1	0%	Index Scan on airports_data as ml on (ml.airport_code = f.departure_airport)
1	0%	Index Scan on airports_data as ml_1 on (ml_1.airport_code = f.arrival_airport)
3	0%	Materialize
3	0%	Bitmap Heap Scan on boarding_passes as bp
3	0%	Bitmap Index Scan on (bp.ticket_no = '0005432661915'::bpchar)

- **Default Bind:** クエリ文で変数として設定された部分を既定値としてバインディングできます。
- **Reset Bind:** バインディングされた既定値を解除し、変数を確認できます。
- **Formatting:** SQLクエリ文にインデントおよびフォーマットを適用して可読性を高めます。
- **Plan:** Plan情報を確認するにはDB Name、User Name、Passwordを入力してから、 ボタンを選択します。
 - **Text:** テキスト形式でプラン情報を確認できます。
 - **JSON:** JSON形式でプラン情報を照会すると、ボトルネックが発生した箇所がすばやく確認できます。

① JSON形式でプラン情報を照会する機能は、DBXエージェント1.6.12バージョンから対応しています。

- ✓ JSON形式でプラン情報を照会した状態でtableまたはindex項目を選択すると、テーブルのcolumn、index構成が確認できるObject detailウィンドウが表示されます。



The screenshot displays the SQL execution tool interface. The top panel shows a query plan for a PostgreSQL database. The query is a SELECT statement with a subquery. The plan shows a Hash Join Left join on (s.sesysid = u.oid) and a Hash Join Left join on (s.datid = d.oid). The object detail window is open, showing the details of the pg_database table. The table has columns: type, idx_scan, n_live_tup, n_dead_tup, last_autovacuum, last_autoanalyze, seq_scan, n_live_tup, last_vacuum, and last_analyze. The index list shows two indexes: pg_database_datname_idx and pg_database_oid_index.

この機能は、DBXエージェント1.6.15バージョン以降に対応しています。データベース権限に関する設定は、[次の文書](#)を参考にしてください。

AI チューニングガイド

AI チューニングガイドは、SQL クエリ、プラン、統計情報を分析してパフォーマンス問題を診断し、最適化の提案を行う機能です。

開発者や DBA がボトルネックの原因を迅速に特定し、効率的な SQL によりパフォーマンスを改善できるよう支援します。

⚠ 使用条件および注意事項

PostgreSQL、MySQL、SQL Server ではプランの取得が必須です。

プランを取得しない場合、AI チューニングガイドアイコンは  (無効) として表示され、機能を使用できません。

- AI が生成する結果は自動分析に基づいており、精度が 100% ではないことにご留意ください。

1. 分析および診断する SQL をクリックし、**SQL詳細** 画面に移動します。

2. **SQL詳細** 画面の右下にある AI チューニングガイドアイコン  をクリックして AI 分析を開始します。
3. AI 分析結果を確認します。

結果項目	説明
クエリプランおよび概要	クエリの目的と実行概要を表示します。 実行回数、累積実行時間、データベース全体の負荷比率を分析し、当該 SQL がシステムパフォーマンスに与える影響を評価します。
パフォーマンス分析	分析結果を総合し、パフォーマンススコアと診断結果を提供します。 CPU 使用率、ディスク使用率、キャッシュヒット率、待機時間など、クエリ実行過程における詳細なリソース使用量を分析し、ボトルネックの発生区間を視覚的に表示します。
発見された主な問題	主な問題点を要約して表示します。
最適化推奨事項	検出された問題に基づき、最適化されたクエリを提案します。

デフォルトプランを照会する

デフォルトプランユーザーファイル作成時、エージェントが事前定義されたDB情報を使用してプランを自動的に照会するように設定できます。 **デフォルトプランの照会**項目のチェックボックスを選択した後、次のように予め入力したDB接続情報を通じてプラン情報を照会します。

• Linux Windows

1. エージェント設定(**whatap.conf**)ファイルに `plan_db` オプションを追加し、DB名を入力します。

whatap.conf

```
plan_db=(dbname)
```

2. DB設定後 `planuid.sh` スクリプトを使用してユーザーとパスワードを設定します。

```
./planuid.sh user pwd
```

❗ ユーザー(`user`)とパスワード(`pwd`)に特殊文字が含まれている場合、各特殊文字の前にバックスラッシュ(`\`)を追加します。

例、 `user$example` → `user\$example`

1. エージェント設定(`whatap.conf`)ファイルに `plan_db` オプションを追加し、DB名を入力します。

`whatap.conf`

```
plan_db=(dbname)
```

2. DB設定後 `planuid.bat` スクリプトを使用してユーザーとパスワードを設定します。

```
planuid.bat user pwd
```

データベースパラメータ

ホーム画面 > プロジェクト選択 > 分析 > パラメータ

データベース設定値の参照機能を提供し、過去の日付の時点でのデータベース設定を表示することができます。比較機能を使用すると、データベースのパラメータ値を日付別に比較できます。デフォルトでは、現在時刻のデータが表示されます。

データベースにパラメータ値が原因で問題が発生した場合、データベースパラメータ比較機能を使用すると、すばやく原因を特定できます。

パラメータ				
基準日	比較日	インスタンス		
2024/06/14	2024/06/13	master		
			変更を表示	
			2024/06/14 09:23:28 本日	2024/06/13 09:00:37
Key	Value		Value	
uptime	76 days 09:08:33.217941		75 days 08:45:41.923636	
start_time	2024-03-29 15:14:55.323539+00		2024-03-29 15:14:55.323539+00	
zero_damaged_pages	off		off	
xmloption	content		content	
xmlbinary	base64		base64	
work_mem	4096		4096 kB	
wal_writer_flush_after	128		128 * 8kB	
wal_writer_delay	200		200 ms	
wal_sync_method	fdatasync		fdatasync	

1. 基準日と比較日を設定します。
2. インスタンス項目で目的のインスタンスを選択します。
3.  ボタンをクリックします。

現在と選択した過去の日付との変更点を一目で確認する場合は、**変更を表示**ボタンを選択します。

パラメータ				
基準日	比較日	インスタンス		
2024/06/14	2024/06/13	master		
			変更を表示	
			2024/06/14 09:29:57 本日	2024/06/13 09:00:37
Key	Value		Value	
uptime	76 days 09:15:01.651907		75 days 08:45:41.923636	

🟢 今日の日付のデータベースパラメータを比較する

基準日	比較日	インスタンス	
2024/06/14	2024/06/14	master	変更を表示

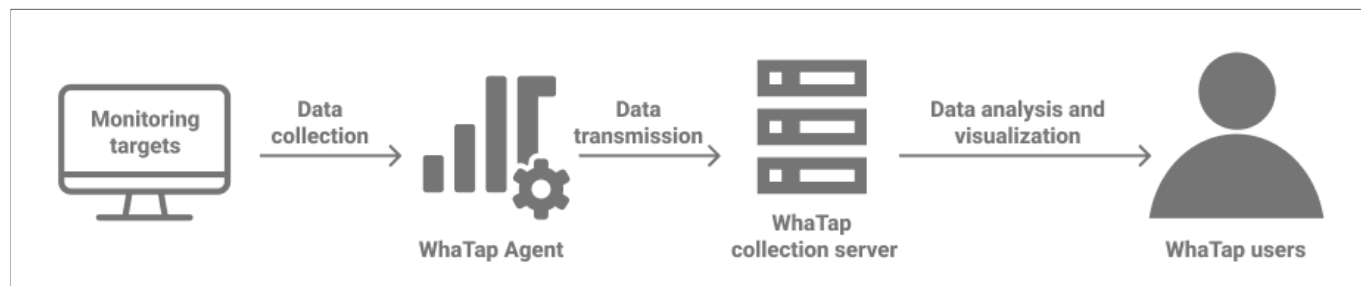
	2024/06/14 09:22:40 本日	2024/06/14 09:00:41 本日
Key	Value	Value
uptime	76 days 09:07:45.453608	76 days 08:45:46.45443
start_time	2024-03-29 15:14:55.323539+00	2024-03-29 15:14:55.323539+00
zero_damaged_pages	off	off
xmloption	content	content

基準日と比較日をすべて今日の日付に設定すると、リアルタイムデータと保存された最後のデータを照会して比較できます。データの最終保存時刻は、データベースサーバーの環境によって異なる場合があります。

メトリクス

メトリクスとは？

WhaTapは、モニタリング対象からデータを収集し、ユーザーに提供します。エージェントから収集されるデータをメトリクスと呼びます。



メトリクスは、次の情報で構成されています。

- **Category**：関連する指標をバインドする単位
- **Tags**：収集対象を特定できる個別情報を含むデータ
- **Fields**：エージェントから収集された指標
- **Time**：メトリクスが収集された時間
- **Oid**：メトリクスを収集したエージェントの固有番号

収集したデータを表示するには、ホーム画面でプロジェクトを選択し、[サイトマップ](#) > [メトリクス表示](#)を選択します。

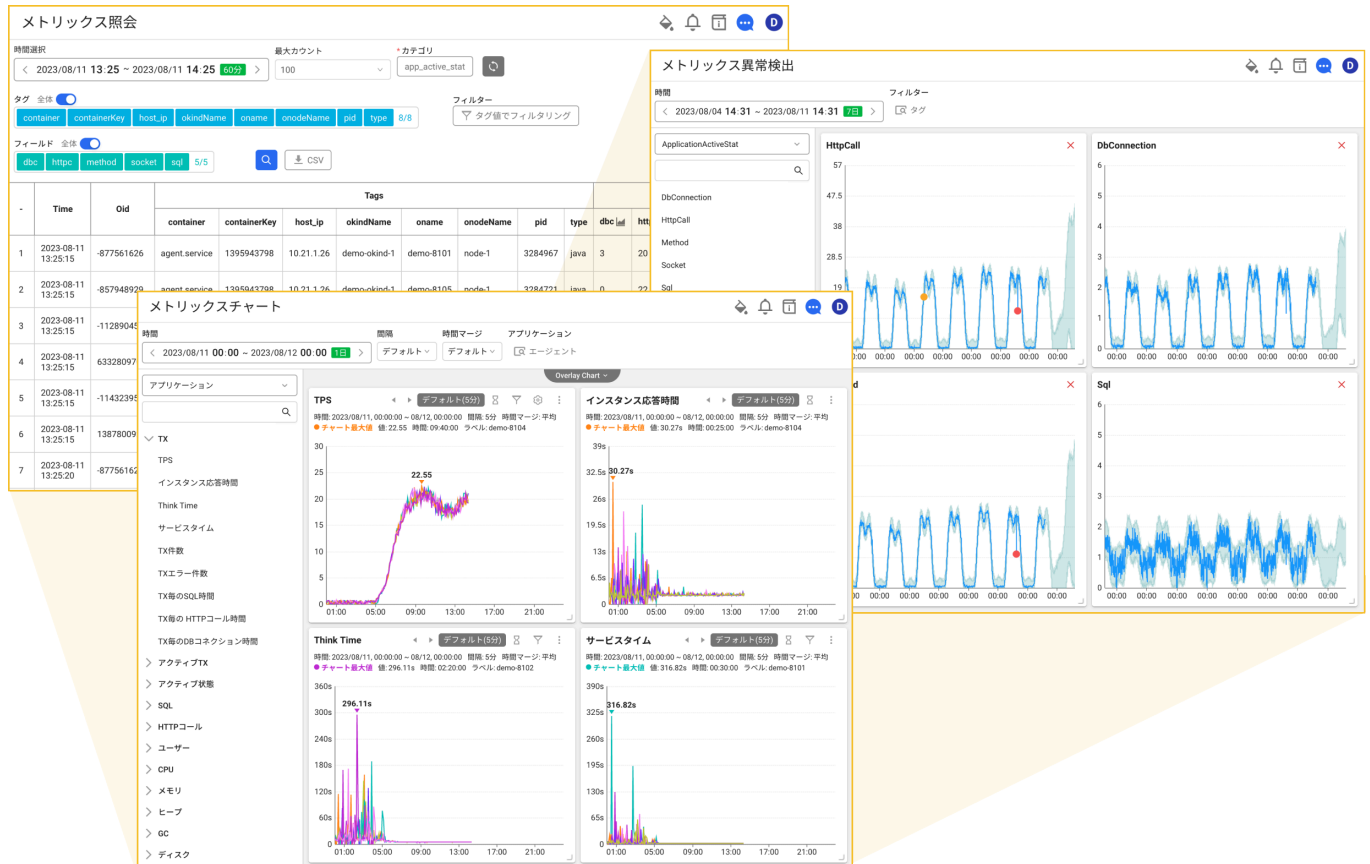
例えば、`db_postgresql_counter` という名前のカテゴリを選択し、関連する指標を検索すると、次のようにカテゴリに対応するタグ(**Tags**)とフィールド(**Fields**)項目を表示できます。

.

メトリクスデータの検索と可視化

WhaTapは収集したメトリクスをさまざまな方式で照会できるメニューを提供します。メトリクスの原本データを照会

できるメトリクス照会、可視化したチャートでメトリクスデータを照会できるメトリクスチャート、AIが学習したメトリクスのパターンと比較して予想パターンを外れた異常を探知できるメトリクス異常検出メニューを確認してください。



PostgreSQLメトリクス

データベースメトリクスは、データベースの作業量とリソース使用量を包括的に収集します。WhaTapモニタリングサービスは、継続的にメトリクスを収集してデータベースのボトルネックを特定し、性能問題の根本原因を特定するのに役立つデータを提供します。

その他のメトリクスに関する詳細については、[次のリンク](#)を参照してください。

agent_status_summary

エージェントの状態に関するメトリクスを10秒間隔で収集したカテゴリです。

Fields

Field	Type	Unit	Description
inActTime	-	ミリ秒(ms)	エージェントが無効になったままの時間
isActive	Boolean	-	現在のエージェントの活性化状態
isRestart	Boolean	-	エージェントが最近再起動したかどうか (<code>true</code> / <code>false</code>)
lastActTime	-	ミリ秒(ms)	最後にエージェントが活性化された状態の時刻 (<code>0</code> : 非活性化された場合)
oid	-	-	プロジェクトに含まれる各エージェントの固有の識別子
startTime	-	ミリ秒(ms)	エージェントが開始された時点でのタイムスタンプ

db_postgresql_counter

Tags

次は db_postgresql_counter カテゴリに収集されるタグ値のガイドです。

Tag	説明	備考
agentIp	エージェントごとのIPアドレスです。	-
dbIsMulti	マルチテナント対応可否です。	-
dbName	データベース名です。	-
dbType	データベースの種類です。	-
dbVersion	利用中のデータベースのバージョンです。	-
oname	エージェント名です。	一意の値

Field

次は db_postgresql_counter カテゴリに収集されるフィールド値のガイドです。

Metrics	Description	Category
active_sessions	現在、活性化されているセッションの数です。	Connection
CPUUtilization	データベースインスタンスで使用中のCPUリソースの比率です。	Performance
FreeStorageSpace	インスタンスの使用可能なストレージ空間です。	I/O
FreeableMemory	システムですぐに使用できるメモリの量です。	I/O
NetworkReceiveThroughput	インスタンスが受信するネットワークデータの速度です。	Troughput

Metrics	Description	Category
NetworkTransmitThroughput	インスタンスから送信するネットワークデータの速度です。	Troughput
blk_read_time	バックエンドでデータブロックを読み取るのにかった時間 (milisecond)です。	I/O
blk_write_time	バックエンドでデータブロックを書き込むのにかった時間 (milisecond)です。	I/O
blks_hit	ディスクブロックがすでにバッファキャッシュで検出され、読み取りが不要な合計回数です。	I/O
blks_hit_ratio	全体照会のうち、キャッシュで照会された割合(%)です。 算出方式： $\text{sum}(\text{blks_hit}) / \text{sum}(\text{blks_hit} + \text{blks_read}) * 100$	I/O
blks_read	ディスクで直接読み取ったデータブロックの数です。	I/O
buffers_checkpoint	チェックポイントの間にフラッシュされたバッファ数です。	Info
check_point	データベースのチェックポイントの発生回数です。	Troughput
checkpoint_write_time	チェックポイントの間、データを書き込むのにかった時間 (milisecond)です。	Troughput
checkpoints_timed	checkpoint_timeout の環境設定値によって発生したチェックポイントの数です。	Troughput
commit_count	トランザクションコミットの回数です。	Troughput
confl_bufferpin	バッファピンの衝突によるセッションの衝突回数です。	Troughput
confl_deadlock	デッドロックの衝突回数です。	Troughput
confl_lock	ロック衝突によるセッション衝突の数です。	Troughput
confl_snapshot	スナップショットの衝突によるセッション衝突の回数です。	Troughput

Metrics	Description	Category
<code>confl_tablespace</code>	テーブルスペースの衝突によるセッション衝突の回数です。	Troughput
<code>conflicts</code>	データベース復旧作業との競合エラーでキャンセルされたクエリの数です。	Troughput
<code>connect_error</code>	データベース接続失敗の回数です。	Connection
<code>deadlock_count</code>	発生したデッドロックの合計回数です。	Performance
<code>dml_count</code>	DML(データ操作言語) 作業の合計回数です。	Troughput
<code>fatal_count</code>	クリティカルなエラーが発生した回数です。	Info
<code>idle_trx</code>	アイドル状態のトランザクションの数です。	Connection
<code>idle_trx_time</code>	アイドル状態のトランザクションの累積時間です。	Connection
<code>idx_scan</code>	インデックスLeafブロックに保存されているキーを利用してテーブルレコードにアクセスする方式です。このメトリクスは、1分ごとに表示する値です。	Performance
<code>index_hit_ratio</code>	インデックスでキャッシュ照会された比率(%)です。	Performance
<code>lock_wait_sessions</code>	ロック待機中のセッションの数です。	Performance
<code>long_running_sessions</code>	長時間実行中のセッションの数です。	Performance
<code>max_connections</code>	許可された最大接続数です。 <code>pg_settings</code> に設定された値で、この値が不足すると、'FATAL ERROR: too many clients already.' エラーが発生する可能性があります。	Info
<code>max_runtime</code>	最も長く実行されたクエリの実行時間です。	Performance
<code>node_state</code>	Replication関係で構成されたデータベースであり、 <code>1</code> : Master、 <code>2</code> : Slaveを意味します。	Info

Metrics	Description	Category
numbackends	データベースに接続されている現在のバックエンドプロセスの数です。	Info
oldest_cur_xid	最も古い現在のトランザクションIDです。	Info
per_to_emergency_autovac	非常自動vacuumの発生回数です。	Performance
per_to_wraparound	トランザクションIDラップアラウンドの防止vacuum作業の回数です。	Performance
replication_count	有効になっているReplicationデータベース(Slave)の数です。	Performance
replication_delay	複製の遅延時間です。Replication構造の場合、replicationデータベースに反映される遅延時間で、Slave DBで測定される指標です。	Performance
replication_lag	複製されたデータがソースデータに比べて遅れている程度です。	Performance
rollback_count	トランザクションのロールバック回数です。	Performance
seq_scan	テーブルをFull Scanしながらレコードを読み取る方式です。インデックスが存在しない、またはインデックスが存在しても読み取り範囲が広い場合に選択します。このメトリクスは、1分ごとに表示する値です。	Performance
table_hit_ratio	テーブルでキャッシュ照会された割合です。	Performance
temp_bytes	データベースのクエリによって一時ファイルに記録された合計バイト数です。	Performance
temp_files	データベースでクエリとして生成された一時ファイルの数です。	Performance
total_runtime	すべてのクエリの合計実行時間です。	Performance
total_sessions	現在接続されているセッションの合計数です。	Connection
tps	1秒当たりのトランザクション処理件数です。	Performance

Metrics	Description	Category
tup_deleted	データベースでクエリによって削除されたタブルの数です。	Troughput
tup_fetched	データベースからクエリに読み込まれたタブルの数です。	Troughput
tup_inserted	データベースでクエリによって挿入されたタブルの数です。	Troughput
tup_returned	データベースのクエリから返されたタブルの数です。	Troughput
tup_updated	データベースでクエリによって更新されたタブルの数です。	Troughput
uptime	データベースが実行中の時間です。	Info
vacuum_running_count	現在実行中のvacuum作業の数です。	Performance
wait_sessions	待機状態のセッションの数です。	Performance
wal_count	記録されたWAL(Write-Ahead Logging) 項目の数です。WALファイル数の急増は、一般的に短期間で多数の変更や一時テーブルおよびオブジェクトを生成するvacuumと類似したメンテナンス作業が原因で発生します。これはゆっくりと正常レベルに戻る必要があります。これにより、通常、ディスクI/OとCPUのアクティビティが多くなり、正常に戻るまでアプリケーションクエリの実行速度が遅くなります。このメトリクスは、1分ごとに表示する値です。	Performance
warning_count	警告が発生した回数です。	Info

XOS関連メトリクス

XOSエージェントから収集されるメトリクスについての説明です。

Metric	Unit	Description
cpu(xos)	pct	CPU使用率

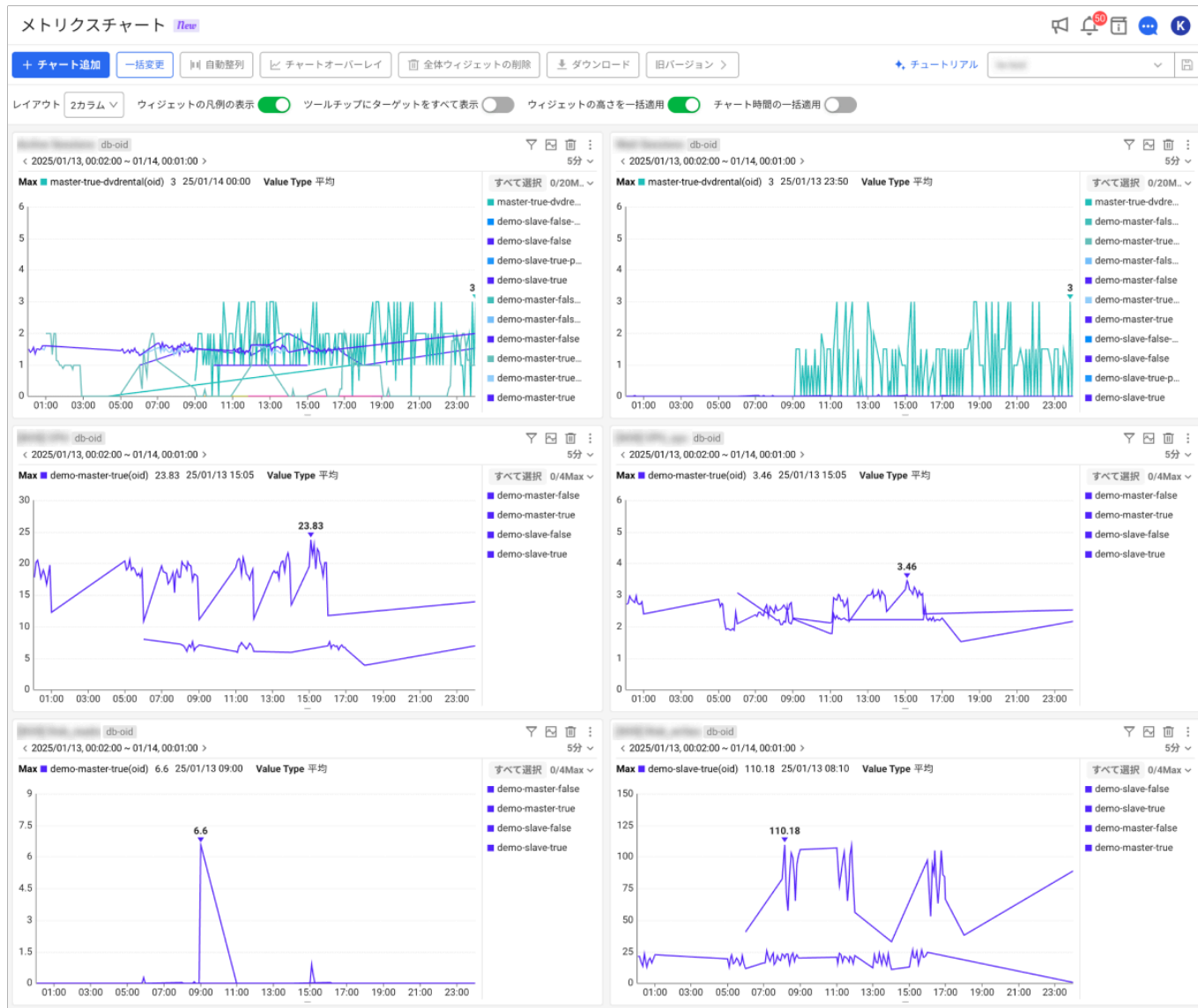
Metric	Unit	Description
		<code>cpu_user</code> + <code>cpu_sys</code> + <code>cpu_iowait</code> の合計
<code>cpu_idle(xos)</code>	pct	CPUが遊休状態の時間
<code>cpu_iowait(xos)</code>	pct	CPUがI/O作業を待機した時間
<code>cpu_sys(xos)</code>	pct	CPUがカーネルコードの実行に使用された時間
<code>cpu_user(xos)</code>	pct	CPUがユーザーコードの実行に使用された時間
<code>disk_read_bytes(xos)</code>	バイト	ディスクから読み取ったデータ量
<code>disk_reads(xos)</code>	block	ディスクで読み取られたブロック数
<code>disk_write_bytes(xos)</code>	バイト	ディスクに記録されたデータ量
<code>disk_writes(xos)</code>	block	ディスクに記録されたブロック数
<code>mem(xos)</code>	pct	メモリ使用率
<code>mem_available(xos)</code>	MB	使用可能なメモリサイズ
<code>mem_buffcache(xos)</code>	MB	buff/cache用途に使用中のメモリサイズ
<code>mem_free(xos)</code>	MB	使用していないメモリサイズ
<code>mem_swapfree(xos)</code>	MB	未使用のスワップ領域サイズ スワップはディスク領域を使用するため、使用時に性能が低下する可能性がある
<code>mem_swaptotal(xos)</code>	MB	全体スワップ(Swap)サイズ 物理メモリ不足時に使用
<code>mem_total(xos)</code>	MB	全体メモリのサイズ

Metric	Unit	Description
<code>mem_used(xos)</code>	MB	使用中のメモリサイズ
<code>net_rcv_bytes(xos)</code>	バイト	ネットワークから受信したデータ量
<code>net_rcv_packets(xos)</code>	count	ネットワークから受信したパケット数
<code>net_send_bytes(xos)</code>	バイト	ネットワークで送信したデータ量
<code>net_send_packets(xos)</code>	count	ネットワークで送信したパケット数

メトリクスチャート

ホーム画面 > プロジェクト選択 > 分析 > メトリクスチャート *New*

メトリクスチャート *New* メニューは、WhaTapモニタリングソリューションの主要機能であり、様々なモニタリング対象から収集したデータを時系列チャートで視覚化して提供します。これにより、ユーザーはシステムの性能を効果的にモニタリングし、異常をすばやく識別して、問題解決に必要なインサイトを確認できます。



- 様々なメトリクスを時系列チャートを通じて視覚的に確認でき、障害発生時点の様々なメトリクスを一度に分析できます。
- 特定の時間範囲を選択して過去のデータを照会し、その期間の性能を分析できます。
- 特定の時間帯や条件に合ったデータをフィルタリングすることで、より正確な分析ができます。
- ユーザーは、目的のチャートを画面に配置し、レイアウトを自由に調整できます。
- ユーザーが希望するメトリクスを配置し、プリセットとして保存し、必要に応じて読み込みできます。

❗ 既存のメトリクスチャートメニューとの違い

Service 2.8.0 アップデート以降の変更点は以下の通りです。 アップデート後の変更事項を画面で確認するには、右上の**チュートリアル**ボタンを選択します。

- ユーザーが時間、対象、インターバルオプションを設定した後、モニタリングする対象のメトリクスを特定し、チャートを追加できるようにユーザー経験(UX)を改善しました。
- ダッシュボードに配置されたウィジェットを削除することなく、チャートの照会時間の範囲を変更できます。
- ダッシュボードに配置したすべてのウィジェットのオプションを一括で適用できる**一括変更機能**を提供します。
- プリセット機能により、ダッシュボードに配置されたウィジェットの情報を保存し、読み込みできます。
- 提供する様々なオプションにより、ダッシュボードのレイアウトやウィジェットの機能を調整できます。

チャートウィジェットを追加する

ダッシュボードにメトリクスをウィジェットとして追加するには、画面左上の**チャート追加**ボタンを選択します。

照会時間の設定

時間でメトリクスを照会する日付と時間を設定します。

対象を選択する

プロジェクトに含まれるエージェントの中から、対象を選択します。 個々のエージェントを選択するか、エージェン

ト設定で分類されたグループ単位で選択できます。

対象

全体

エージェント

種類別

サーバー...

対象検索

Q

すべて選択 0/9

すべて折りた...

Q

▼ Project

▼ demo-okind-0

demo-8104

demo-8102

demo-8100

▼ demo-okind-1

demo-8103

demo-8105

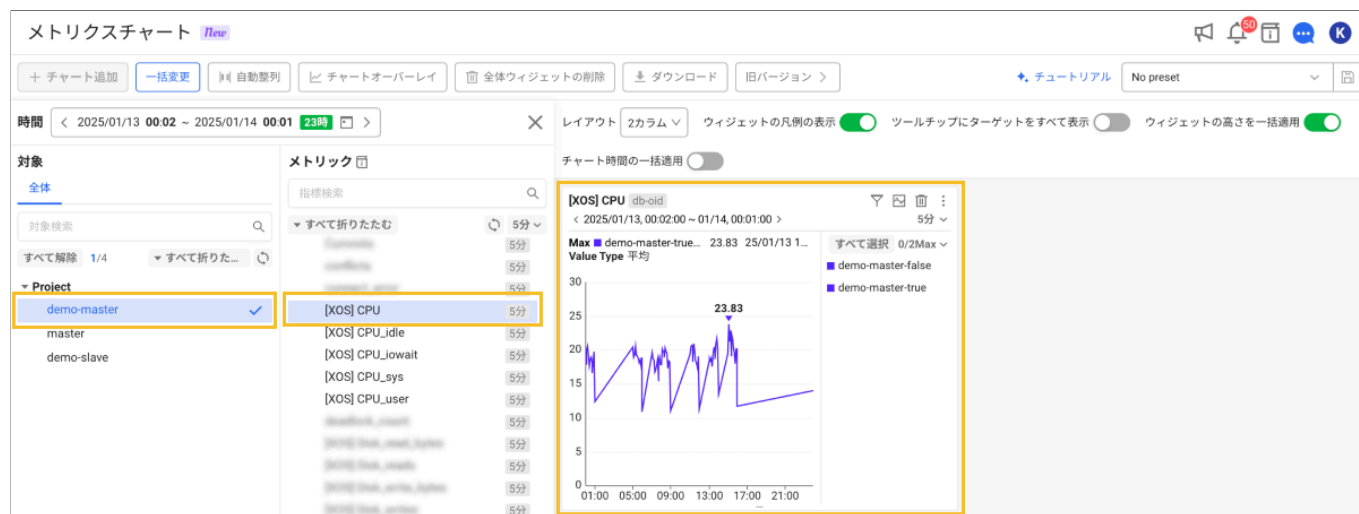
demo-8101

- **エージェント**: 個別エージェントまたはすべてを選択できます。
- **種類別**: エージェント設定で `whatap.okind` に分類されたグループ単位でエージェントをモニタリングします。
- **サーバー別**: エージェント設定で `whatap.onode` に分類されたグループ単位でエージェントをモニタリングします。

- ❗
- 設定した時間と日付によって、選択できる対象が変更される場合があります。
 - エージェントの一覧からマウスをクリックして、対象を選択または解除できます。 `Shift` キーを使用して複数のエージェントを選択できます。
 - **すべて選択**または**すべて解除**ボタンを選択すると、関連する下位対象をすべて選択または解除できます。
 - 希望のエージェントが一覧に表示されない場合は、**リフレッシュ**ボタンを選択します。
 - **すべて折りたたむ**または**すべて開く**ボタンを選択してグループ単位で一覧を隠したり、展開したりできます。

指標を選択する

モニタリングする対象エージェントを選択した後、**メトリックパネル**に選択できる一覧が表示されます。一覧からご希望のメトリクスを選択します。



選択したメトリクスに対応するチャートウィジェットが右側のダッシュボードに配置されます。他のメトリクスを選択し、チャートウィジェットを追加できます。チャート領域を拡大して見るには、**時間オプション**の右にある×ボタンを選択します。**対象とメトリックパネル**が消え、チャート領域のみが画面に表示されます。

- ① **指標検索入力欄**に文字列を入力すると、希望のメトリクスをすばやく検索できます。
- メトリクスのカテゴリとフィールド名に基づいても検索できます。カテゴリおよびフィールドの詳細については、[次の文書](#)を参照してください。
- すべて折りたたむ**または**すべて開く**ボタンを選択して、カテゴリ単位でメトリクス一覧を非表示または表示できます。
- 希望するメトリクス項目が一覧に表示されない場合は、**リフレッシュ**ボタンを選択します。

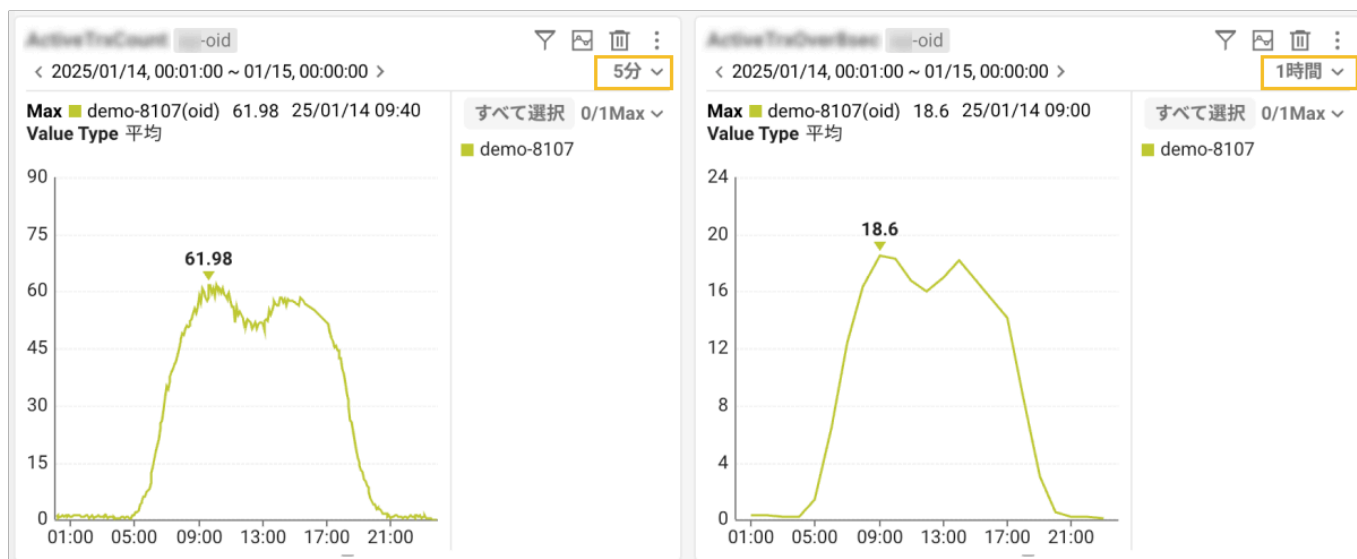
インターバルを設定する



チャートごとにインターバルを設定でき、データの集計単位を調整して、様々な時間間隔のデータを可視化できます。メトリックパネルの右上にある🕒ボタンを選択すると、インターバルオプションを選択できます。

メトリクスごとにデフォルトで設定されたインターバル時間が異なり、照会時間の範囲によってインターバル時間が変更されます。

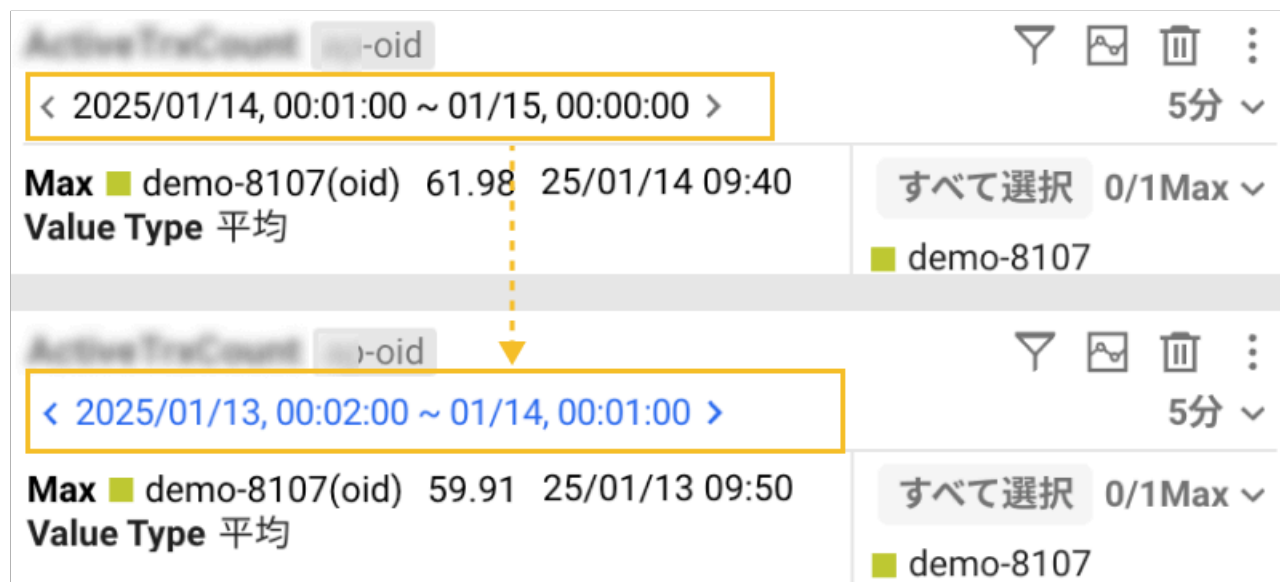
例えば、**5分間**インターバルは5分単位の平均値を表示し、**1時間**インターバルは、1時間単位の平均値を表示します。



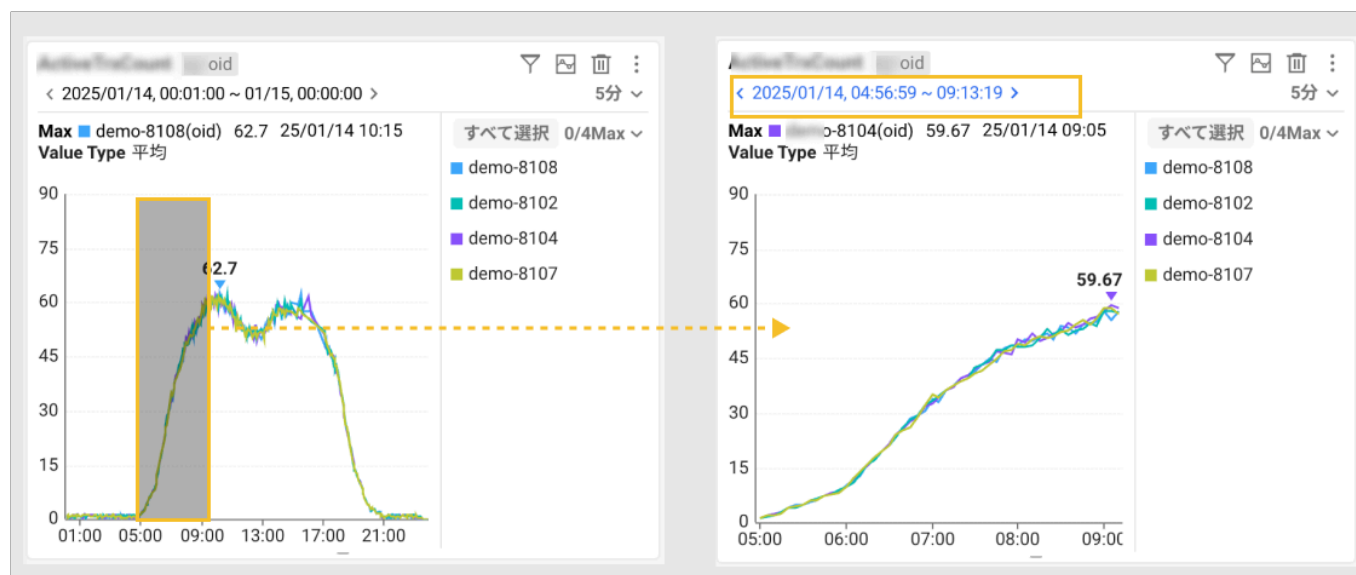
- ✓ インターバルオプションを調整し、短い間隔の詳細データから長い間隔の全体的な推移まで、多様な観点からデータを分析できます。

照会時間を変更する

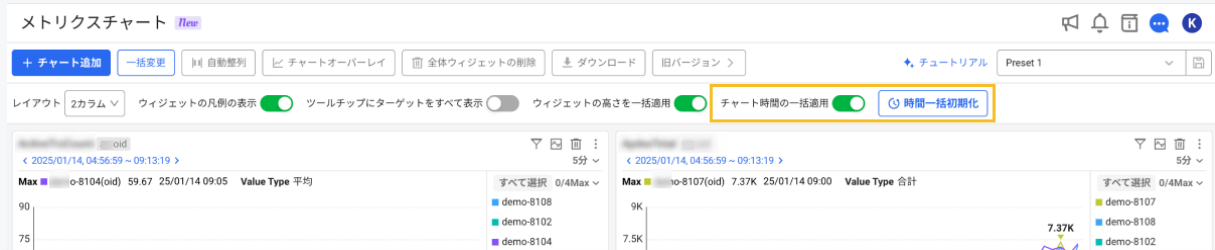
ウィジェットのチャート照会時間の範囲を変更するには、ウィジェット上部のくまたは>ボタンを選択します。時間変更範囲は、時間オプションに設定された時間基準と同じです。例えば、時間オプションで1日間に設定されている時、くまたは>ボタンを選択すると、時間を1日単位で変更できます。



またはチャートの特定領域をドラッグすると、ドラッグした範囲に時間を変更して照会できます。



- ✓ 画面の上部にある**チャート時間の一括適用**オプションを有効にして、ウィジェットの時間を変更すると、すべてのウィジェットの時間を変更できます。

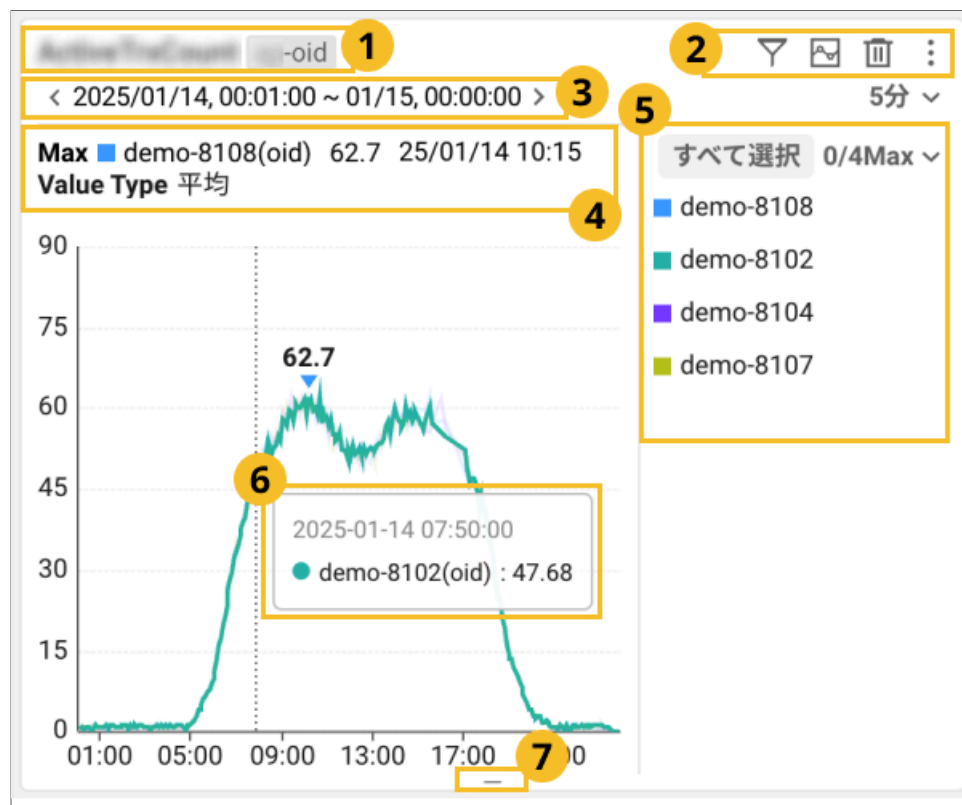


- 照会時間の範囲を変更した後、直近の時間に再度変更するには、 ボタンを選択します。
- もし画面の上部にある**チャート時間の一括適用**オプションを有効にした場合、 **時間バッチ初期化**ボタンが表示されます。このボタンを選択すると、すべてのウィジェットの時間を初期化できます。

- ⓘ **一括変更機能**を利用してウィジェット全体の時間を変更し、モニタリング対象とインターバルも修正できます。詳細については、[次の文書](#)を参照してください。

チャートウィジェットの詳細

チャートウィジェットに表示される各アイテムの詳細を確認します。



- **① メトリクス名とカテゴリ:** チャートウィジェットのメトリクス名を表示します。メトリクスの右側には、照会対象のカテゴリをタグ形式で表示します。ご利用の商品によりカテゴリ名称は異なる場合がございます。
 - `*oid` : 個々のエージェントを選択した時に表示されます。
 - `project` : プロジェクト単位で対象を選択した時に表示されます。
 - `okind` : エージェント設定で `whatap.okind` で分類されたグループを選択した時に表示されます。
 - `onode` : エージェント設定で、`whatap.onode` で分類されたグループを選択した時に表示されます。
- **② ウィジェットオプション:** チャートウィジェットのオプションメニューです。詳細については、[次の文書](#)を参照してください。インターバルオプションの詳細については、[次の文書](#)を参照してください。
- **③ 時間:** チャートのデータを照会する時間範囲を表示します。〈 または 〉 ボタンを選択して照会時間範囲を調整できます。
- **④ データ:** モニタリング対象のうち照会したデータの最大値(Max)とデータ結合方式(Value Type)を確認できます。
- **⑤ 凡例:** モニタリング対象を凡例として表示します。凡例表示の詳細については[次の文書](#)を参照してください。
- **⑥ チャート:** 照会したデータを時系列グラフで表示します。マウスをグラフの上にかざすと、詳細な情報を確認できます。チャートの特定領域をドラッグすると、ドラッグした範囲に時間を変更して照会できます。

- **7 サイズ調整:** チャートウィジェットの真ん中の下部領域をドラッグすると、ウィジェットのサイズを調整できます。

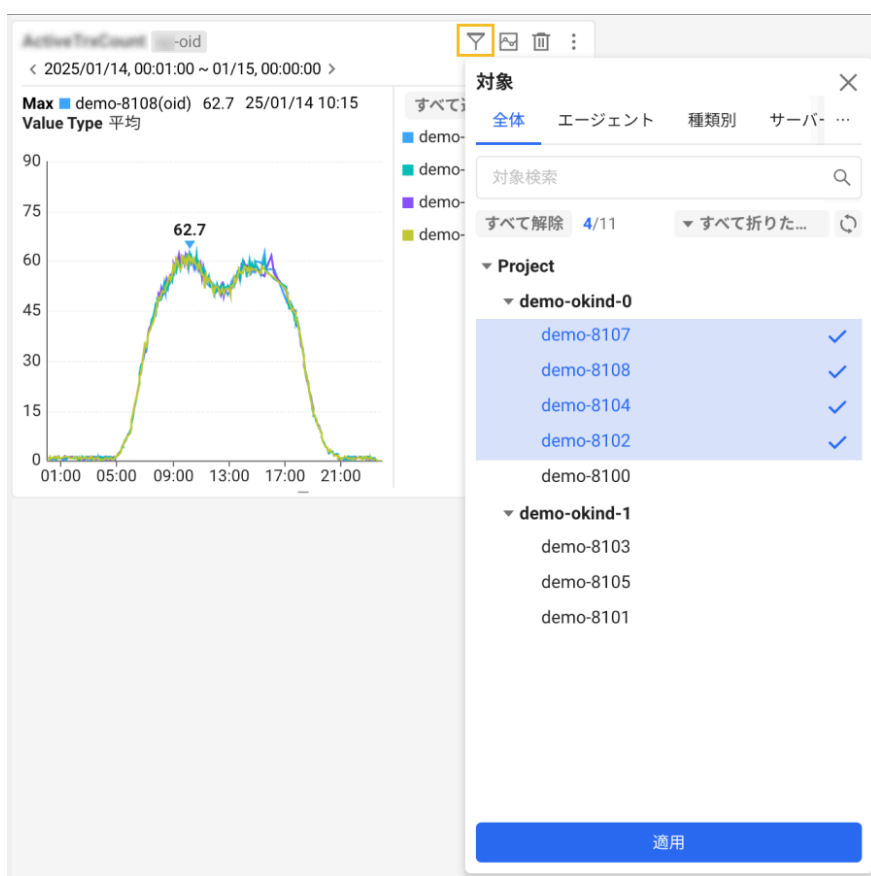
チャートウィジェットオプションを利用する

チャートウィジェットの右上にあるオプションボタンを使用すると、様々な機能が利用できます。

🚫対象をフィルタリングする

画面に追加したチャートウィジェットに、モニタリング対象を追加または除外できます。

1. 🚫ボタンを選択すると、モニタリング対象を選択できる画面メニューが表示されます。



2. 画面の案内に従って希望する対象を選択します。

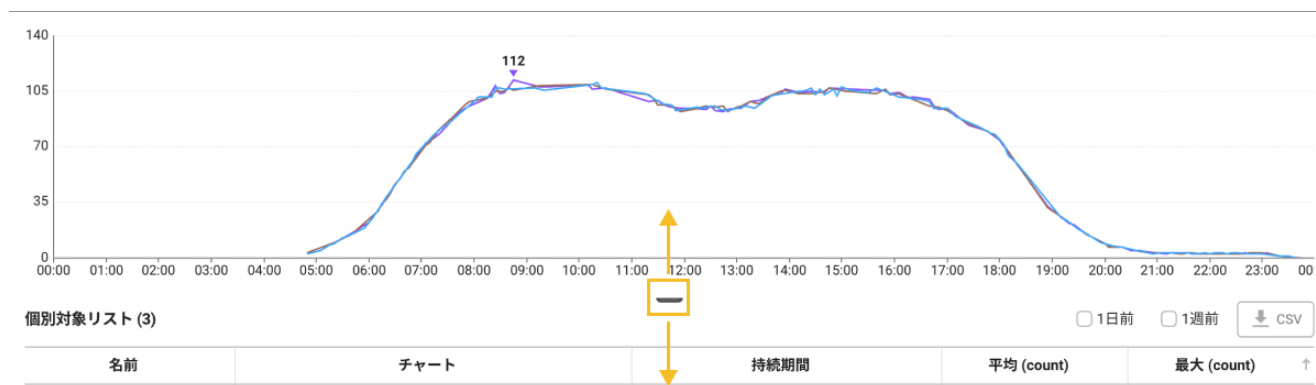
3. 適用ボタンを選択します。

詳細を照会する

チャートのデータをより広い画面で詳細情報を確認するにはボタンを選択します。詳細検索画面が表示されます。個々の対象に対するチャートと平均および最大値を照会できます。



- **1日前**: 1日前のデータと比較して確認できます。
- **1週間前**: 1週間前のデータと比較して確認できます。
- **↓ CSV**: 照会したデータをCSV形式のファイルとしてダウンロードできます。
- グラフチャートの下部にあるアイコンをドラッグすると、チャートのサイズを調整できます。

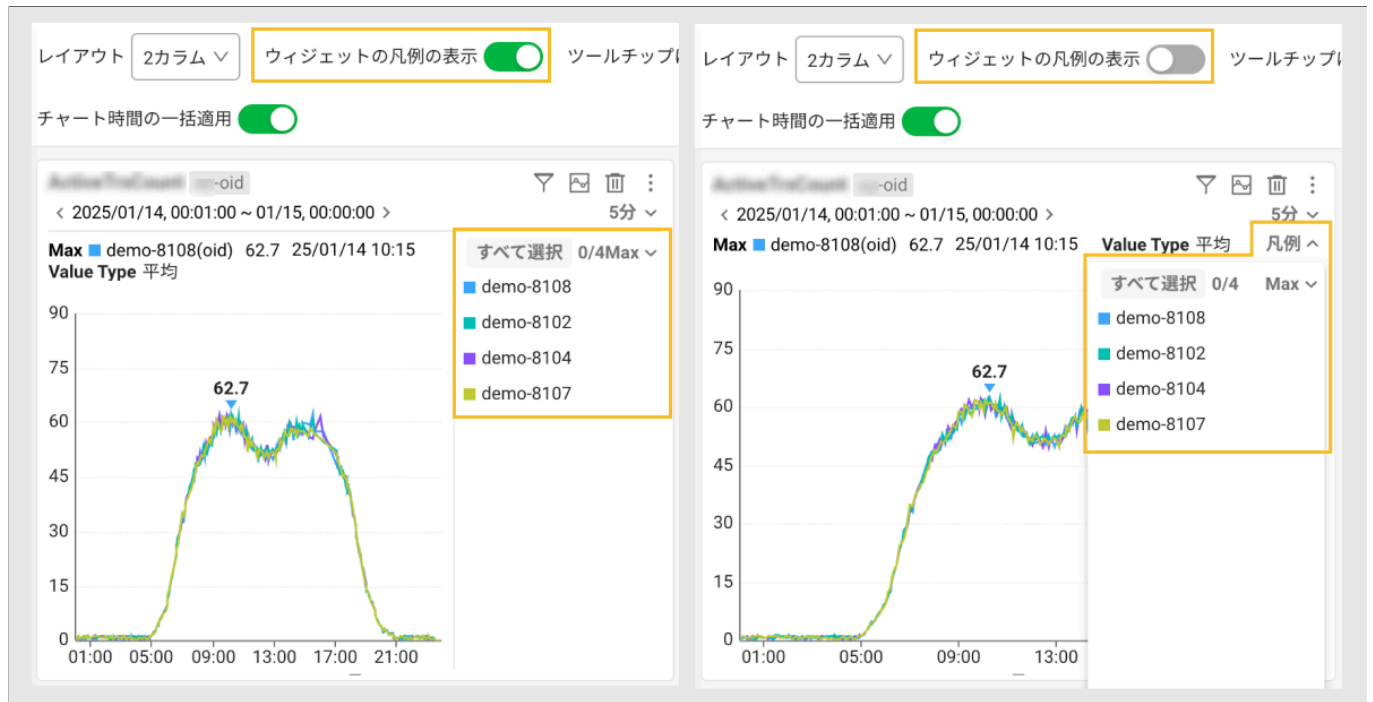


🗑️ チャートウィジェットを削除する

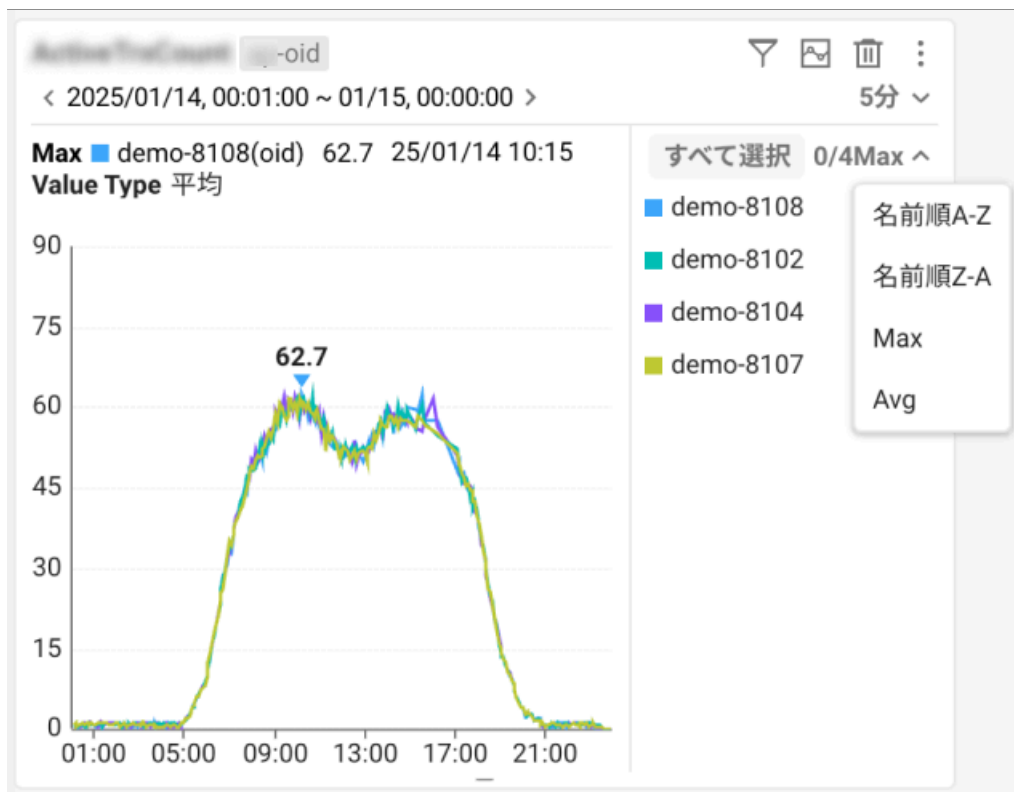
画面に配置したチャートウィジェットを削除するには、🗑️ ボタンを選択します。

凡例を活用する

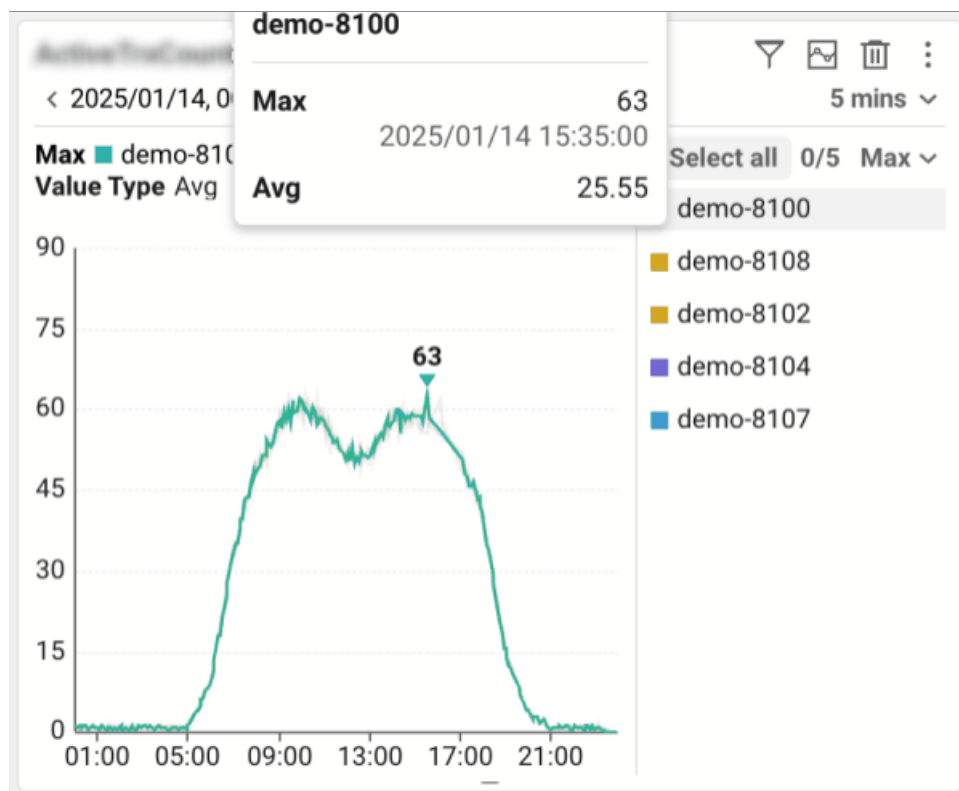
凡例を通じてチャートに表示されるモニタリング対象を確認し、希望する基準で整列したりフィルタリングできます。画面上部のウィジェットの**凡例の表示**オプションをオンまたはオフにすると、ウィジェットのチャートの右側に凡例を表示・非表示できます。オプションをオフにした状態では**凡例**をクリックしてモニタリング対象を確認して選択できます。



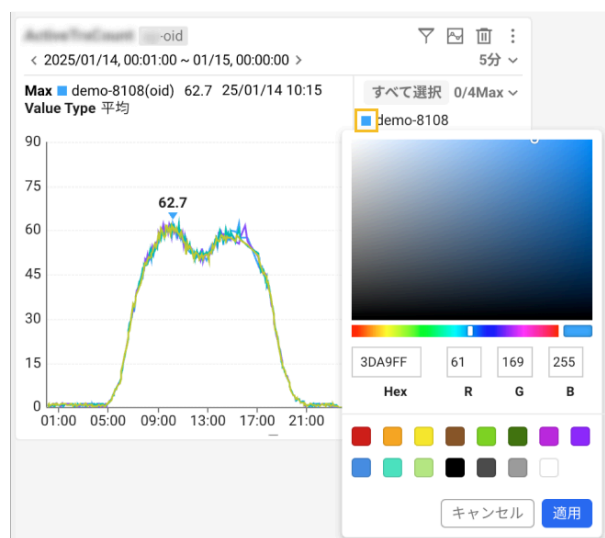
- すべて選択/すべて解除ボタンを選択してモニタリング対象を選択または解除できます。
- 凡例一覧では、モニタリング対象を名前順、最大値、平均値に基づいてソートできます。



- 凡例一覧の各項目にマウスをオーバーすると、そのモニタリング対象に対するグラフを強調して確認でき、Max値が記録された特定時間をツールチップで表示します。または、凡例一覧の対象を選択してもフィルタリングできます。



凡例色を変更する



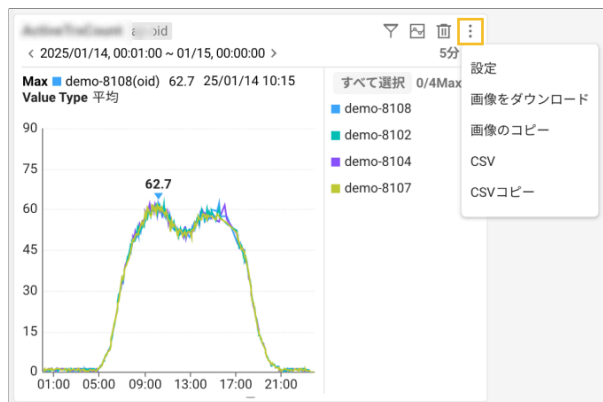
モニタリング対象に該当するグラフの色を変更できます。凡例一覧から色を変更したい対象の左側にある四角い箱

を選択します。希望する色を選択した後**適用**ボタンをクリックします。

❗ 設定した色は、プリセット機能によって保存されません。この機能は、今後のアップデートで提供される予定です。

追加オプションを利用する

チャートウィジェット右上にある  ボタンを選択すると、次の追加オプションが利用できます。



- **設定**
 - **自動Y軸を使用:** 照会したデータ範囲内でY軸の大きさを自動調整できます。
 - **最大値表示:** チャートのグラフ領域に最大値を表示できます。
- **画像をダウンロード:** チャートを画像形式のファイルとしてダウンロードできます。
- **画像のコピー:** チャートを画像形式でクリップボードにコピーできます。
- **CSV:** チャートで照会したデータをCSV形式のファイルにダウンロードできます。
- **CSVコピー:** チャートで照会したデータをCSV形式でクリップボードにコピーできます。

プリセットを設定する

ダッシュボードからユーザーが設定したウィジェットの設定とレイアウト状態を保存して読み込むことができます。ウィジェットのサイズを調整し、希望する場所に配置して新しいプリセットを作成できます。

- ❗ プロジェクト修正権限のあるメンバーのみがプリセットを保存または変更できます。メンバー権限の詳細については、[次の文書](#)を参照してください。

新しいプリセットを作成する

1. ダッシュボードから、希望する形にウィジェットを配置してみてください。サイズを変更して、頻繁に確認するウィジェットのみを配置することもできます。
2. 画面の右上に🏠ボタンを選択します。
3. 新しいプリセット名を入力します。



4. 保存ボタンを選択します。

プリセット一覧から新しく保存したプリセットを確認できます。

- ❗
- 新しく作成したプリセットに変更が加えられた場合は、再度プリセットを保存してください。🏠ボタンを選択し、同じ名前でプリセットを保存します。既存のプリセットに変更を上書きします。
 - ダッシュボードの変更を保存せずに別のメニューに移動すると、変更は保存されません。
 - プリセットはプロジェクト単位で保存され、他のユーザーと共有できます。
 - プリセットに保存される対象は、ダッシュボードのレイアウトとダッシュボードに配置したウィジェット、ウィジェットで修正したオプション(対象, 間隔)です。画面上部の時間オプションは保存されません。

- ❗ プリセットを読み込んでからウィジェットの照会時間を変更するには、**一括変更機能**を利用します。詳細については、[次の文書](#)を参照してください。

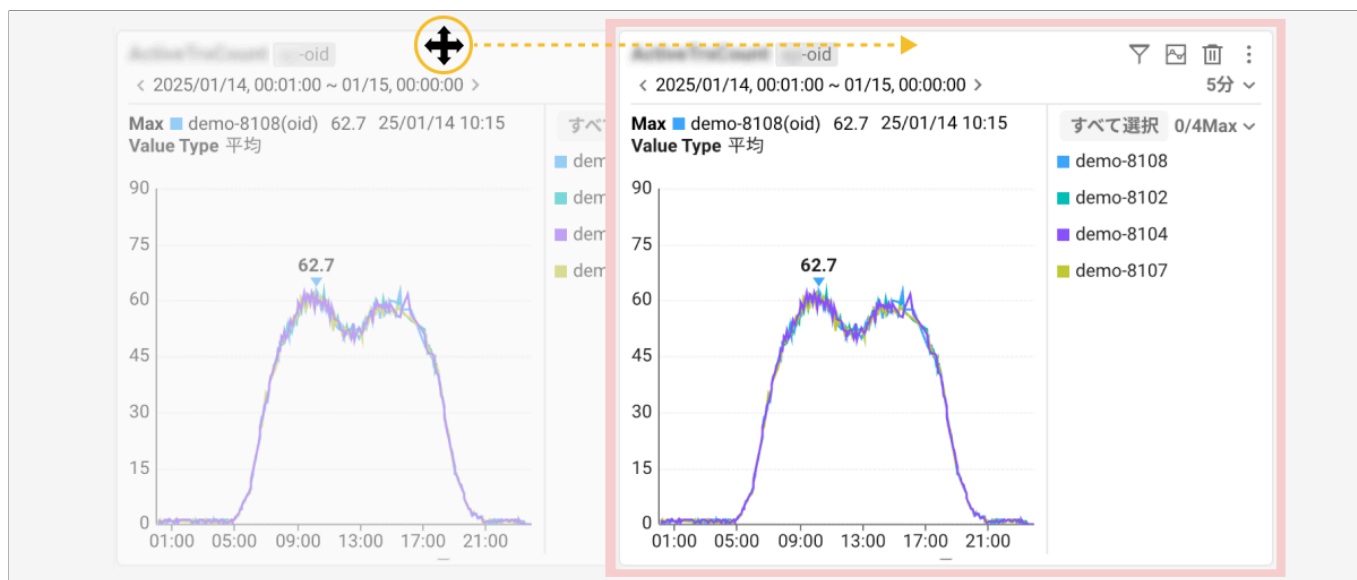
プリセットを削除する

使用していないプリセットは、プリセット一覧から削除できます。プリセット一覧から削除する項目の右側にある × ボタンを選択します。

ダッシュボードウィジェットの編集

ダッシュボードに配置したウィジェットをユーザーが希望する場所に配置したり、サイズを調整できます。

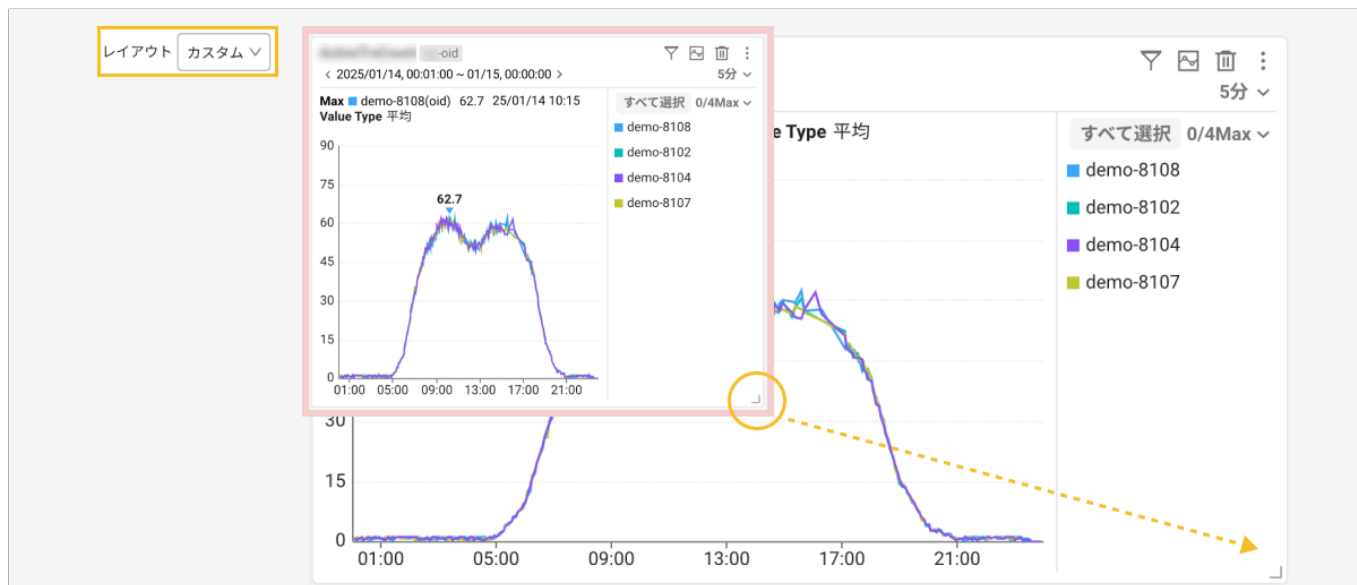
ウィジェットの移動



ウィジェットの上部にマウスをかざすと、**+**に変更されます。この時、クリックしたまま希望の位置にドラッグするとウィジェットを移動できます。

ウィジェットのサイズ変更

ダッシュボードオプションのレイアウトがカスタムに設定されていると、ウィジェットのサイズを調整できます。



ウィジェットの右下にある「」をマウスでクリックした状態で希望するサイズにドラッグします。

- ⓘ レイアウトがカスタムに設定されていない場合は、ウィジェットの高さだけを調整できます。ウィジェットの真ん中の下部領域をマウスでクリックしたままサイズを調整します。

ダッシュボードオプションを利用する

画面上部にあるオプションメニューを利用して、チャートウィジェットに適用された様々な設定を一度に適用できます。また、ダッシュボードのレイアウトを希望する形に設定できます。



チャートウィジェットオプションを一括変更する

画面に配置されたチャートウィジェットに、詳細オプションを一度に適用できる機能を提供します。画面左上に一括変更ボタンを選択します。一括変更画面が表示されたら、オプションを設定し、適用ボタンを選択します。

一括変更

時間

< 2025/01/15 00:00 ~ 2025/01/15 23:59 23時

対象

全体

対象検索

すべて解除 1/4

すべて折りたたむ

▼ Project

demo-master

master

demo-slave

適用ウィジェット

すべて選択

☐ Active Sessions

2025/01/13, 00:02:00 ~ 01/14, 00:01:00 db-oid 5分

☐ Wait Sessions

2025/01/13, 00:02:00 ~ 01/14, 00:01:00 db-oid 5分

☐ [XOS] CPU

2025/01/13, 00:02:00 ~ 01/14, 00:01:00 db-oid 5分

☐ [XOS] CPU_sys

2025/01/13, 00:02:00 ~ 01/14, 00:01:00 db-oid 5分

☐ [XOS] Disk_reads

2025/01/13, 00:02:00 ~ 01/14, 00:01:00 db-oid 5分

☐ [XOS] Disk_writes

2025/01/13, 00:02:00 ~ 01/14, 00:01:00 db-oid 5分

間隔

変更しない

5分

1時間

キャンセル

適用

- **時間:** チャートのデータ照会範囲を設定できます。
- **対象:** モニタリング対象を選択します。個々のエージェントを選択したり、エージェント設定を通じて分類したグループ単位で選択できます。

304

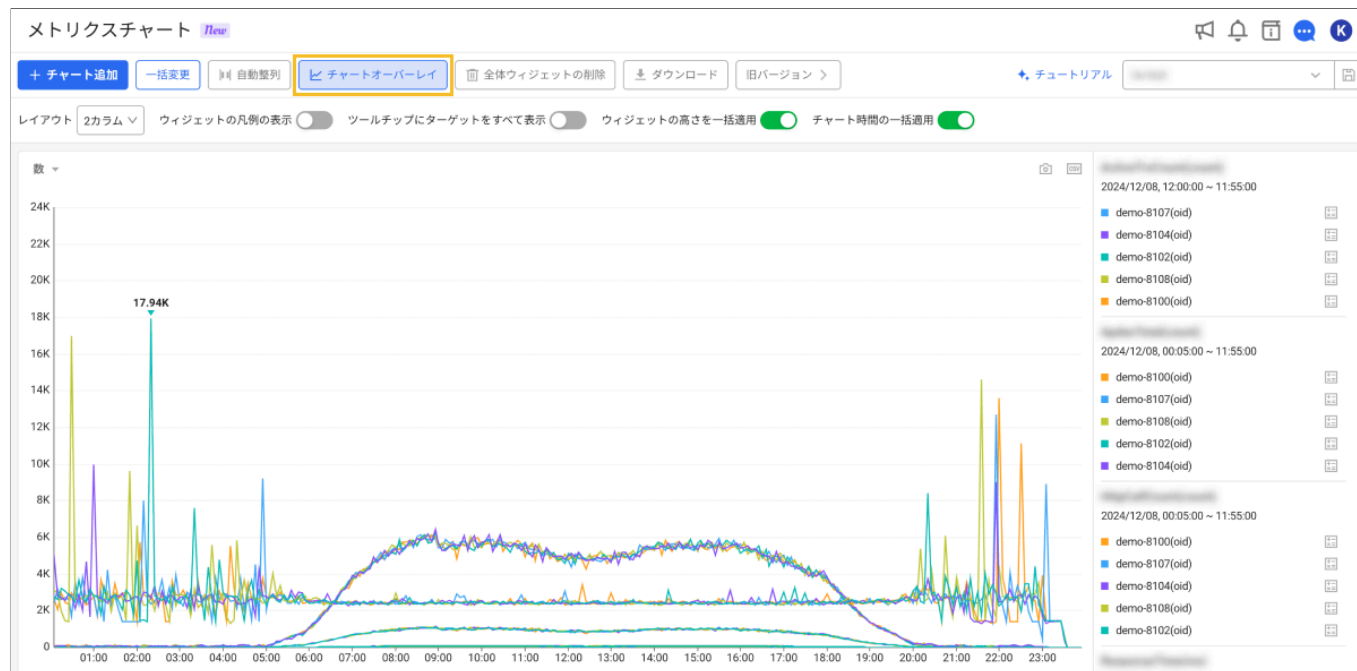
- **エージェント**: 個別エージェントまたはすべてを選択できます。
- **種類別**: エージェント設定で `whatap.okind` に分類されたグループ単位でエージェントをモニタリングします。
- **サーバー別**: エージェント設定で `whatap.onode` に分類されたグループ単位でエージェントをモニタリングします。
- **間隔**: 様々な時間間隔でデータの集計単位を調整できます。
- **適用ウィジェット**: 変更するオプションを適用するチャートウィジェットを選択します。選択したウィジェットに変更したオプションを一度に適用できます。

ダッシュボードを自動ソートする

ダッシュボードに配置したウィジェットの位置を自動的にソートするには**自動整列**ボタンを選択します。ウィジェットの幅、高さが小さい順にウィジェットの配置を自動的にソートします。幅が同じであれば、高さが小さい順にソートします。

複数のチャートを比較する

ダッシュボードに配置した複数のチャートデータを1つのチャートで確認できる機能です。この機能により、同時時間帯の様々なメトリクス情報を他のメトリクスと比較しながら確認できます。**チャートオーバーレイ**ボタンを選択します。画面にオーバーレイチャートを配置できます。



- オーバーレイチャート左上の**数**を選択すると、x軸チャートのデータ基準を変更できる一覧が表示されます。希望する項目を選択して、チャートのデータを比較できます。
- : 照会中のデータに基づいて、チャートを画像形式のファイルとしてダウンロードできます。
- : 照会中のデータに基づいて、チャートをCSV形式のファイルとしてダウンロードできます。
- 右側のセクションでは、メトリクスごとのエージェントの一覧を確認できます。個々のエージェントの色は、チャートのグラフの色と一致します。

配置されたウィジェットを一括削除する

ダッシュボードに配置された複数のウィジェットを一度に削除できます。 **全体ウィジェットの削除**ボタンを選択します。確認メッセージが表示されたら削除ボタンを選択します。

CSVファイルでダウンロードする

ダッシュボードに配置したウィジェットのデータをCSV形式のファイルとしてダウンロードできます。 **ダウンロード > CSV**を選択します。時間帯のメトリクスのデータを確認できます。

イメージファイルでダウンロードする

ダッシュボードに配置されたすべてのウィジェットを1つのイメージファイルでダウンロードできます。 **ダウンロード** > **画像をダウンロード** を選択します。 ダッシュボード画面をそのままイメージで保存できます。

以前のバージョンを利用する

以前のバージョンのメトリクスチャートメニューを利用するには、**旧バージョン**ボタンを選択します。 または **サイトマップ** > **分析** > **メトリクスチャート** **Old** メニューパスで確認できます。

❗ メトリクスチャート **Old** メニューの詳細は、[次の文書](#)を参照してください。

その他のオプションを利用する

次のその他のオプションで、ダッシュボードのレイアウトを調整したり、ウィジェットに適用されたオプションを簡単に一括設定できます。

レイアウト **2カラム** ▼ ウィジェットの凡例の表示 ☒ ツールチップにターゲットをすべて表示 ☐ ウィジェットの高さを一括適用 ☒ チャート時間の一括適用 ☒

- **レイアウト**: ダッシュボードで列の数、つまり1行に配置できるウィジェットの数を設定できます。 最大4列まで設定できます。 **カスタム**を選択すると、列の数に関係なくウィジェットを配置できます。

❗ **カスタム**では、ウィジェットの幅と高さを自由に調整できます。 他のオプションを選択すると、ウィジェットの高さだけを調整できます。

- **ウィジェットの凡例の表示**: オプションをオンまたはオフにすると、ウィジェットのチャートの右側に凡例を表示・非表示できます。 オプションをオフにした状態では**凡例**をクリックしてモニタリング対象を確認して選択できます。 凡例の活用についての詳細は[次の文書](#)を参照してください。
- **ツールチップにターゲットをすべて表示**: チャートにマウスかざすと、そのグラフの位置に該当する値を確認できます。 マウスが位置した時間帯のすべての対象のデータを確認するにはオプションをオンにします。
- **ウィジェットの高さを一括適用**: オプションをオンにすると、すべてのウィジェットの高さを同じく設定できます。

- **チャート時間の一括適用:** チャートウィジェットのデータ照会時間を変更する時、他のウィジェットのチャート時間も一緒に変更するにはオプションをオンにします。

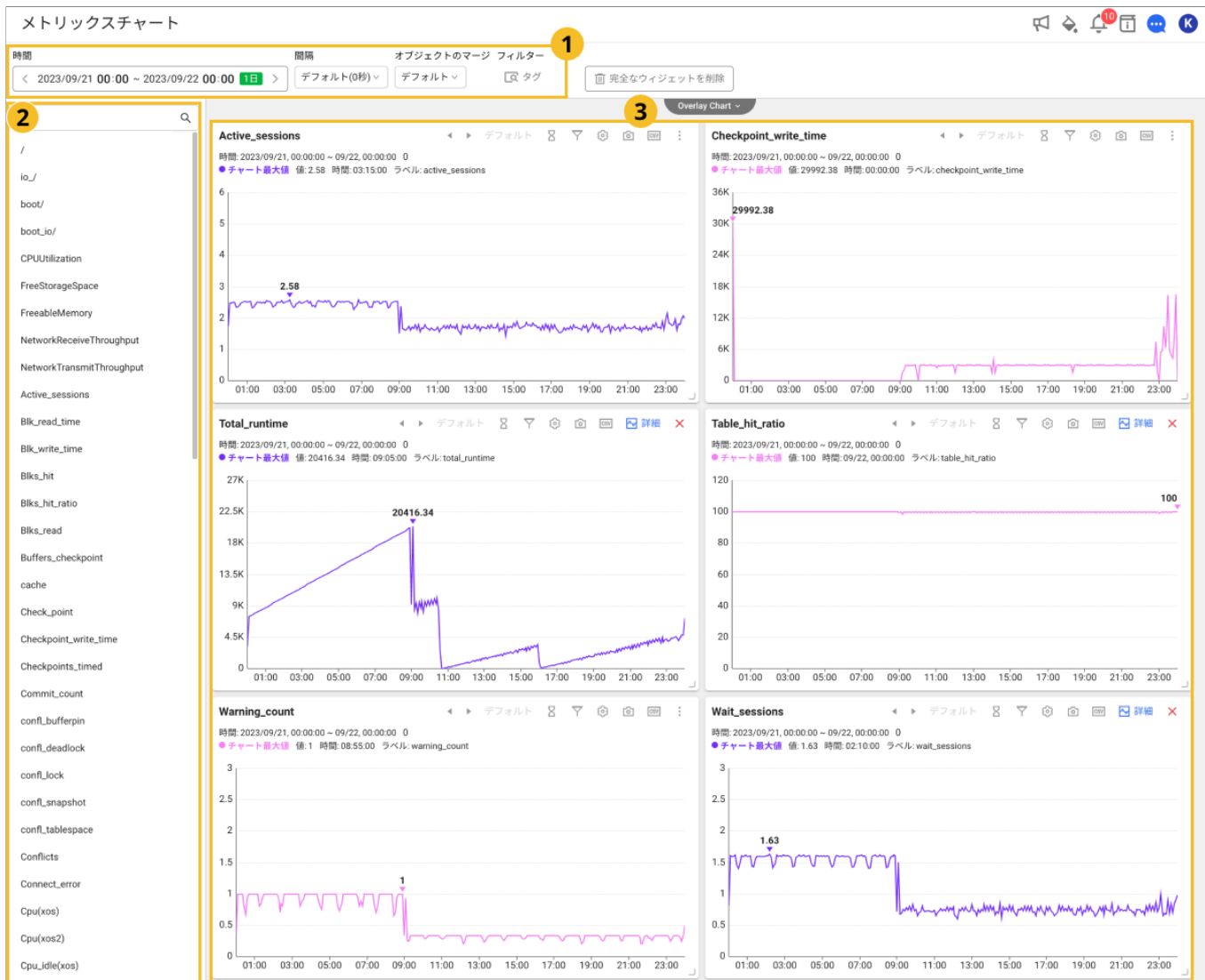
チュートリアルを利用する

画面の右上のチュートリアルボタンを選択すると、[Service 2.8.0](#) アップデート以降の変更を確認できます。

メトリクスチャート

📄 サイトマップ > 分析 > メトリクスチャート Old

メトリクスチャートメニューでは、モニタリング対象から収集されたメトリクスデータを次のようにチャートで照会できます。時間と指標の指定が必須です。



上部オプション

① 領域のメトリクスチャートの上部オプションからチャートの時間範囲とモニタリング対象エージェントを指定できます。

- **時間**：時間の範囲でX軸の開始と終了を指定できます。
- **間隔**：時間間隔でX軸データの間隔を指定できます。
- **時間マージ**：データマージ方式の1つを選んで、指定した時間間隔内のデータをマージできます。

例) 平均は、1時間のデータ平均値を意味します。

-  **エージェント**：照会するエージェントを指定できます。指定しない場合は、すべてを対象に照会します。

指標一覧

② 領域は、オプションを参照する指標の一覧です。指標を検索するか、一覧で目的の指標を選択します。① 領域の上のメニューで指定した時間範囲のデータに基づいて③ 領域でチャートウィジェットを使用してデータを表示することができます。

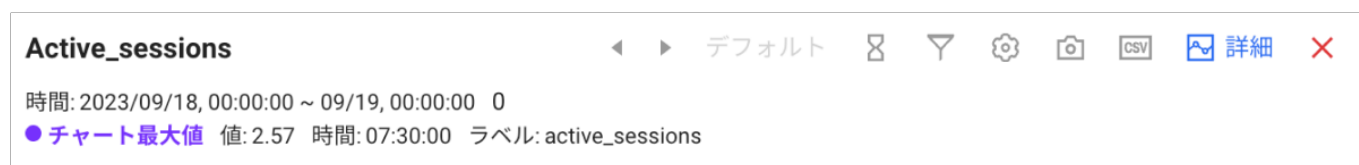
ⓘ データ結合

データのマージは、**オブジェクトのマージ**と**時間マージ**方法を提供します。

- **時間マージ**は、ソースデータからフィールド値が同じデータを一定の間隔でマージします。
- **オブジェクトのマージ**は、異なるフィールド値を持つデータの中からタグが一致する場合、そのデータをマージします。

チャートウィジェット

③ 領域のチャートウィジェットの左上で指標名を確認できます。チャートウィジェットの右上で次のオプションを確認できます。



- タイムシフト：◀ ▶ 左矢印または右矢印ボタンを使用して、選択した時間範囲だけ-1または、+1ずつ移動することができます。

例えば、時間範囲が2月13日00:00~2月14日00:00の場合、◀ 左矢印を選択すると、2月12日00:00~2月13日00:00までのデータが表示できます。

- 間隔/時間のマージ：① トップメニューで指定した間隔と時間のマージを修正できます。
- モニタリング対象：▼ アイコンを選択して、モニタリング対象を指定できます。選択しない場合は、すべて表示されます。
- 時間比較：📊 アイコンを選択すると、同じ指標の以前の時間帯の推移を比較できます。
- スナップショット：📷 アイコンを選択すると、ウィジェットのオプションを除くチャートのスナップショットを取得できます。
- CSV：📄 アイコンを選択して、チャートを描画したデータをCSVファイルとしてダウンロードできます。
- 詳細：🔍 アイコンを選択して詳細を表示します。モニタリング対象が多い場合、モニタリング対象の指標推移を特定して確認できるため便利です。

① メトリクスチャートウィジェットの上部にオプションが表示されない場合は、⋮ アイコンを選択します。

ウィジェットを削除する

画面に配置したすべてのウィジェットを削除するには、🗑️ 全体ウィジェットの削除ボタンを選択します。

メトリクス照会

ホーム画面 > プロジェクト選択 > 分析 > メトリクス照会

メトリクス照会メニューからタグベースで特定のメトリクスを照会できます。

メトリクス照会

時間選択

< 2023/08/17 08:04 ~ 2023/08/17 09:04 1時 >

最大カウント

100

* カテゴリ

app_counter

🔍

タグ 全体

container

containerKey

host_ip

okindName

oname

onodeName

pid

type

8/8

フィルター

▼ タグ値でフィルタリング

フィールド 全体

_clock_delta

active_tx_0

active_tx_3

active_tx_8

active_tx_count

apdex_satisfied

apdex_tolerated

apdex_total

arrival_rate

httpc_count

httpc_error

httpc_time

metering

resp10

resp95

resp_time

sql_count

sql_error

sql_fetch_count

sql_fetch_time

sql_time

tps

tx_count

tx_dbc_time

tx_error

tx_http_time

tx_sql_time

tx_time

tx_time_sqr_sum

5/29

🔍

📄 CSV

	Time	Oid	Tags								Fields				
			container	containerKey	host_ip	okindName	oname	onodeName	pid	type	active_tx_3	arrival_rate	sql_time	tps	tx_error
1	2023-08-17 08:04:30	-877561626	agent.service	1395943798	10.21.1.26	demo-okind-1	demo-8101	node-1	3284967	java	13	19.554815	9.995699	19.008858	1
2	2023-08-17 08:04:30	1387800924	agent.service	1395943798	10.21.1.26	demo-okind-0	demo-8102	node-0	3285018	java	10	21.37008	5.108556	19.723042	1

1 時間とカテゴリの選択

① 領域でメトリクスが収集された時間選択と最大カウントおよびカテゴリを指定できます。時間選択とカテゴリは、必ず指定してください。

- 時間選択：メトリクスが収集された時間を指定して照会できます。既定値は 1時間 です。基本オプションとして提供する照会時間以外に、ユーザーが時間選択タブを選択して日付と時間を指定できます。

① 時間選択オプションの利用方法の詳細については。

- 最大カウント：③ 一覧から参照するメトリクスの最大数を指定できます。10、50、100、200、300、1000、2000、3000 個まで設定できます。
- カテゴリ：関連指標の分類単位です。カテゴリタブを選択し、カテゴリを指定できます。
- 🔄 カテゴリ、タグ、フィールドオプションの更新：カテゴリ、タグおよびフィールドオプションを再読み込みできます。

2 タグとフィールドの選択

2 領域でタグとフィールドを選択します。ユーザーが個別に指定しない場合、デフォルト設定が完全な選択になります。

- **タグ**：収集された対象を区分できる固有情報データです。
- **フィールド**：モニタリング対象から収集された指標です。
- **フィルター**：▼タグ値でフィルタリングボタンを選択し、タグ値を設定してフィルタリングできます。

例) oname の値を demo-8101 に設定すると、フィルタリングしたデータを表示することができます。

- 🔍 **検索**：条件を設定して検索アイコンを選択すると、3 領域でそのメトリクスの原本データを検索できます。
- 📄 **CSV**：メトリクスの原本データをCSVファイルとしてダウンロードできます。

3 メトリクステーブル

収集されるメトリクスを事前に特定できないため、収集されるすべてのメトリクスの原本データを確認することが重要です。上記の条件領域で目的の条件を設定した後 3 領域で、そのメトリクスの原本データをテーブル形式で検索できます。ユーザーがタグとフィールドの各条件を指定することによって、テーブルのカラムが変更されます。

- ❗ • メトリクスの参照時、**時間選択**と**カテゴリ**を指定する必要があります。
- メトリクス参照時、**タグとフィールド**の指定はオプションです。

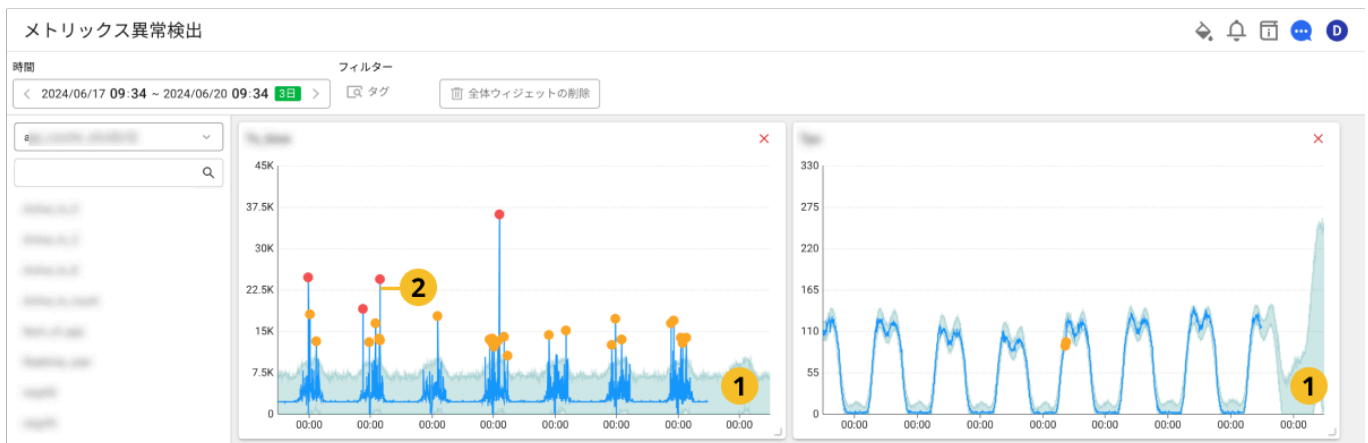
メトリクス異常検出

ホーム画面 > プロジェクト選択 > 分析 > メトリクス異常検出

さまざまなメトリクスのパターンをAIが学習した予想パターンと比較できます。予想パターンを外れた異常探知をグラフチャートで確認できます。過去のデータに基づいて繰り返されるパターンを確認し、今後の指標値の予測に活用できます。

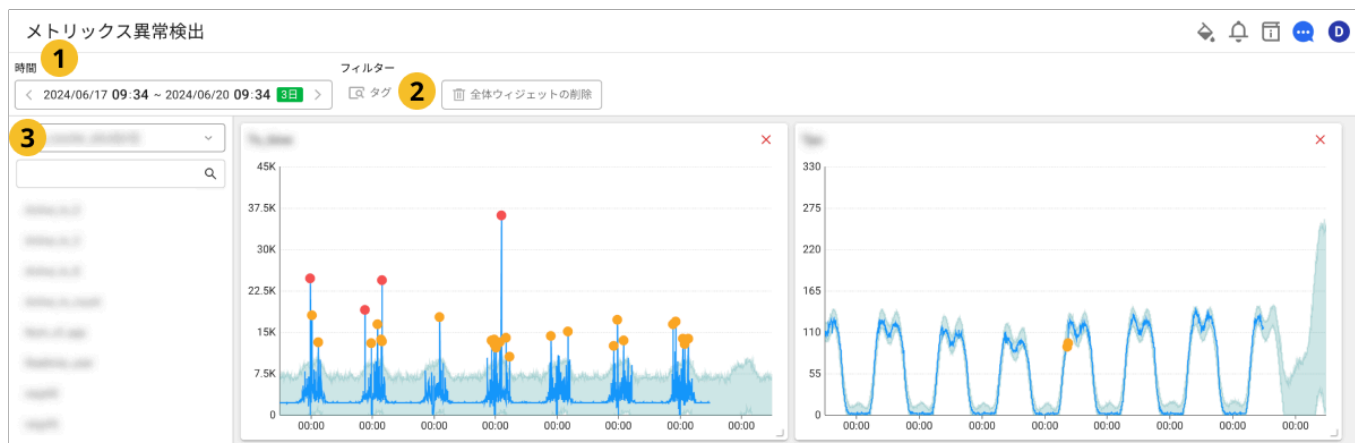
- ❗ 画面に配置したウィジェットは、別のメニューに移動すると保存されずに初期化されます。
- パターン分析と異常値検出機能を除くと分析 > メトリクスチャートメニューに似ています。
- 異常値(Anomaly Detection)アラート通知機能の技術的根拠は、メトリクス異常値です。異常値アラート通知設定の詳細については、[次の文書](#)を参照してください。

ウィジェットを確認する



- ① 明るい色のグラフ領域は、AIによって分析した予想されるパターンです。
- ② 青色グラフは、プロジェクトのメトリクスの推移です。
- AIによって分析された予想パターンから外れるとオレンジ色と、赤色のステップでグラフにドットが表示されます。予想されるパターンの範囲から大幅に外れた値を赤色として表示します。

ウィジェットを配置する



1. ① 時間で、目的の時間間隔を設定してください。最大1か月間隔まで設定できます。
2. ② フィルターでメトリクスの範囲を選択します。
3. 以下の一覧でモニタリングするメトリクスを選択します。

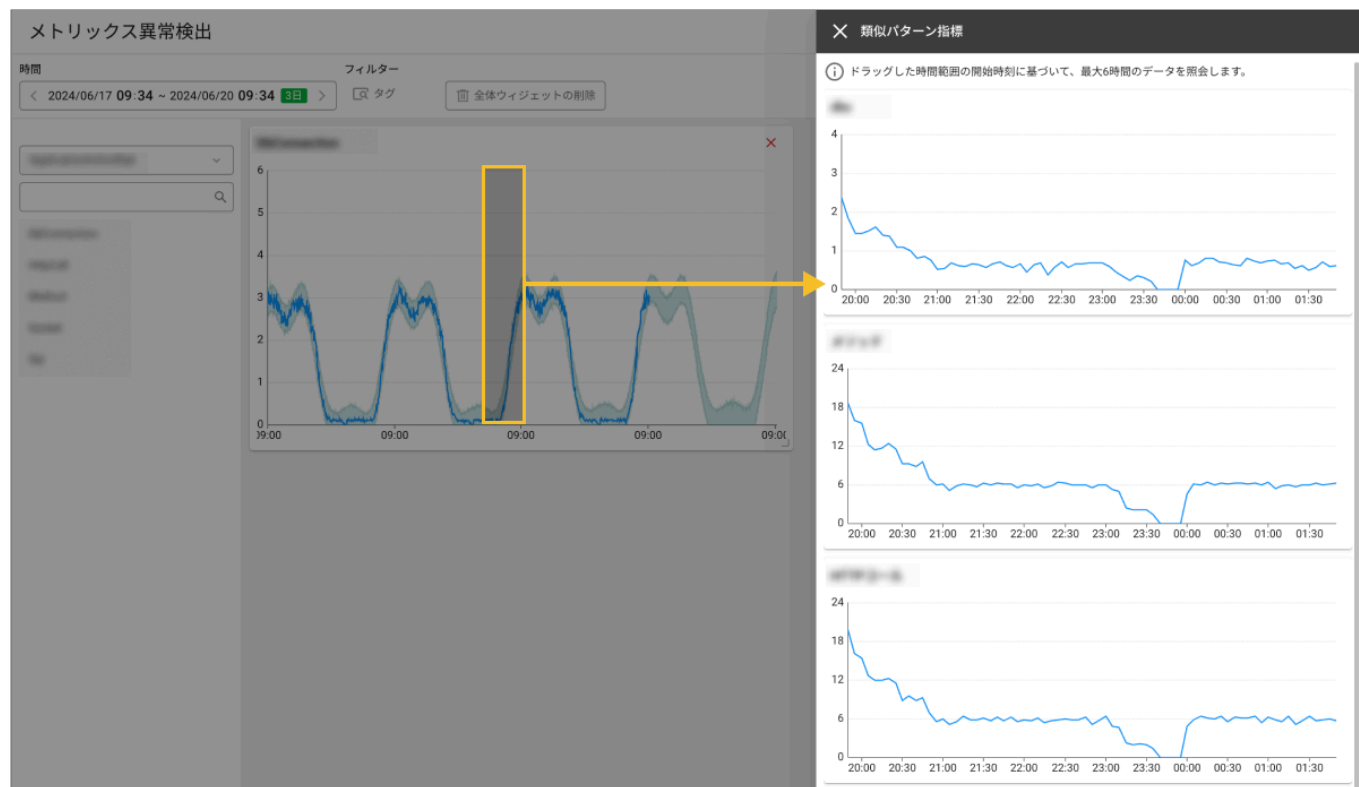
選択したメトリクスを画面の右側に配置します。

- ① モニタリング対象を選択してメトリクスを絞り込んで確認するには、② タグを選択します。画面右側にタグ選択リストが表示されます。目的の項目を選択し、ウィジェットを追加します。
- 画面に配置したウィジェットは、時間またはタグ値を変更してもチャートに反映されません。
- 画面に配置したウィジェットを削除するには、ウィジェット右上の✕ボタンを選択します。
- ウィジェットの左上を選択した状態でドラッグすると、ウィジェットの位置を変更できます。
- ウィジェットの右下を選択した状態でドラッグすると、ウィジェットのサイズを変更できます。

ウィジェットを削除する

画面に配置したすべてのウィジェットを削除するには、 全体ウィジェットの削除ボタンを選択します。

関連メトリクスを確認する



ウィジェットでチャートの一部の領域をドラッグします。画面の右側に**類似パターン指標**画面が表示されます。ドラッグした時間範囲の開始時刻に基づいて、最大6時間の関連メトリクスを照会できます。

メトリクストレンド分析

メトリクス推移分析 は、過去 30 日以内のメトリクス変化を可視化し、トレンドを比較・分析できる機能です。データベースの主要なパフォーマンス指標を期間別または時間別に照会し、特定の時点や時間帯のパターンを迅速に把握できます。

基本画面

メトリクス推移分析 は、上部の **オプションバー** と下部の **チャート領域** で構成されています。オプションバーで照会範囲と表示メトリクスを選択すると、条件に合ったトレンドグラフが自動的に表示されます。

オプションバー

オプションバーでは、時間範囲の設定、メトリクス選択、プリセット管理などの機能を提供します。

Time Window

特定の時間帯のみのデータを表示したい場合に使用します。
トグルを有効にすると時間範囲入力フィールドが有効になります。デフォルトは無効です。
たとえば、業務時間（09:00～18:00）のデータを分析する場合は、Time Window をオンにして該当時間を指定します。

インスタンス

照会するデータベースインスタンスを選択します。
プロジェクトに接続されたインスタンス一覧が表示され、検索ボックスを使用してインスタンスをすばやく見つけることができます。

メトリクス

表示するメトリクスを選択します。
カテゴリ別にフィールドが一覧表示され、必要な項目を選択して照会できます。

プリセット

頻繁に使用するチャート構成を保存または呼び出すことができます。

- 上部の日付別チャートのクリック状態はプリセットに保存されます。
- 下部の日別トレンドチャートのクリック状態は保存されません。

プリセットの保存

ダッシュボードで設定した指標とレイアウトをプリセットとして保存します。

1. **Time Window**、**メトリクス**などを希望の形で設定します。
2. 画面右上の  アイコンをクリックします。
3. **プリセットとして保存** ウィンドウで名前を入力し、**保存** ボタンをクリックします。

プリセットの呼び出し

1. プリセットドロップダウンリストから保存済みプリセットを選択します。
2. 保存された **Time Window**、**メトリクス**、**チャート**などの設定がそのまま適用されます。

✔ プリセットを保存すると、繰り返し設定することなくすぐに照会できます。

チャート

メトリクス推移分析のチャート領域では、選択したメトリクスの変化を可視化します。

特定期間のトレンド、平均値、最大値などを比較でき、必要に応じて新しいチャートを追加したり不要なチャートを削除できます。

チャートの追加

1. （任意）上部オプションバーで **Time Window** を指定します。
2. **メトリクス** から希望するフィールドを選択します。
3. 画面右上の **+ チャート追加** ボタンをクリックします。

チャートの削除

削除するチャートの右上にある  アイコンをクリックします。

ウィジェット

メトリクス推移分析 の下部チャートは **Interval** と **リンクを開く** 機能を提供します。

これにより特定期間のデータをより詳細に分析したり、該当時間帯の詳細モニタリング画面へ直接移動することができます。

Interval

チャートに表示されるデータの集計単位を変更します。

データを1時間単位または5分単位で照会できます。

デフォルトでは1時間単位の集計データが表示され、より詳細なデータを確認したい場合は5分単位表示に切り替えます。

リンクを開く

チャートの特定時間区間のバーをクリックすると、その時間に移動して該当区間の詳細モニタリングページを開きます。

❗ **リンクを開く** > ボタンは、特定の時間区間のバーをクリックしたときのみ有効になります。
選択した時間帯の詳細な指標を確認できます。

Top SQL

Top SQL画面では、照会期間中のTop SQLリストを提供します。エージェントが収集したアクティブセッション内のSQLに基づいて、実行回数、実行時間、待ち時間などの指標をWhaTapが独自に算出したSQL統計です。

- 最小1分～最大30日の範囲で照会でき、長期分析に適しています。
- 推移チャートを通じて、特定の時間帯のSQL発生パターンを視覚的に把握できます。
- リテラル値を除いたSQLを基準に統計を算出します。

✔ SQL統計（DB）との違い

区分	Top SQL	SQL統計（DB）
データソース	エージェントが収集したアクティブセッション	DBの pg_stat_statements ビュー
統計算出	WhaTap独自算出	DBが提供する統計をそのまま収集
別途DB設定	不要	pg_stat_statements 拡張のインストールが必要
主な指標	実行回数、実行時間（avg/max/sum）、待ち時間	実行回数、実行時間、Row処理、ブロックI/Oなど
照会範囲	1分～30日	時間/日単位
推移チャート	提供	非提供

事前設定

Top SQL分析機能を使用するにはエージェント設定が必要です。

エージェント設定

別途設定なしで5分単位の統計がデフォルトで提供されます。より精密な1分単位の分析が必要な場合、[whatap.conf](#)に以下の設定を追加します。

```
sqlstat_send_interval=1
```

設定項目	タイプ	デフォルト値	説明
sqlstat_send_interval	int	5	SQL統計データを生成・送信する周期（分）。1または5に設定可能。

Top SQLを分析する

Top SQL画面は分析基準タブとソート順を選択して、SQLパフォーマンスをさまざまな視点から分析できるように構成されています。ユーザーはアプリケーション視点のグループ分析から個別SQLパフォーマンス分析までdrill-down/drill-up方式で段階的に探索できます。

Top SQL分析は以下の順序で進めます。

1. 分析基準タブで分析基準を選択してください。
 - 分析基準タイプ：ALL、DB、APPLICATION、USER、HOST
2. ソート順でTop選定指標を選択してください。
 - Top選定指標タイプ：Elapsed Time、Wait Elapsed Timeなど
3. チャートで負荷が高いグループを識別してください。
4. 特定のグループを選択し、SQL Listで個別Top SQLを確認してください。
5. 追加分析が必要な場合は、SQL詳細分析画面に移動して分析してください。

基本オプション

上部フィルター領域で時間、対象インスタンス、フィルターなどを設定できます。

- **時間:** 照会時間範囲を選択（1分～30日）
- **インスタンス:** 照会対象DBインスタンスを選択
- **フィルター:** フィールド別条件検索（デフォルト： SQL Text から'WhaTap'を除外）
- **ソート順:** Top SQLを選定する基準指標（Elapsed Time、Wait Elapsed Time、Avg Elapsed Time、Max Elapsed Time、Execute Count）を選択。選択した指標を基準に降順ソート
- **照会件数:** テーブルに表示する件数を設定
- **Top N:** 推移チャートに表示する上位項目数を設定
- **カラム選択:** 表示するカラムを追加/削除し、ドラッグで順序変更可能
- **ダウンロード:** CSV形式でエクスポート

フィールドパネル

画面左側のフィールドパネルでフィルター条件を簡単に追加できます。フィールド名をクリックして条件を入力すると、その条件がフィルターに追加されます。

フィールド	タイプ	説明
SQL Text	文字	SQL文
Elapsed Time	数値	実行時間合計（秒）
Wait Elapsed Time	数値	待ち時間合計（秒）
Avg Elapsed Time	数値	平均実行時間（秒）
Max Elapsed Time	数値	最大実行時間（秒）
Execute Count	数値	実行回数
WSQL Hash	文字	SQLハッシュ値
User	文字	実行ユーザー
Datname	文字	データベース名

フィールド	タイプ	説明
Application Name	文字	アプリケーション名
Client Hostname	文字	クライアントホスト名

分析基準タブ

分析基準タブでは、Top SQLをどの基準で分析するかを選択します。ALL、DB、APPLICATION、USER、HOSTなどさまざまな分析基準を提供し、選択した基準に応じてSQL実行情報をグループ単位またはSQL単位で分析できます。

ALL

ALLを選択するとSQL単位のTop SQLをすぐに確認できます。チャートもSQL単位の基準で表示され、システム全体でパフォーマンス影響度が高いSQLを直接確認する際に使用します。

DB/APPLICATION/USER/HOST

DB / APPLICATION / USER / HOSTはアプリケーション視点でよく使用する分析基準です。このタブを選択すると、該当基準でSQL実行指標が集計されてチャートに表示されます。

タブ	説明
DB	データベース単位でSQL実行負荷を分析します。
APPLICATION	アプリケーション名基準でSQL負荷を分析します。
USER	ユーザーアカウント基準でSQL負荷を分析します。
HOST	クライアントホスト基準でSQL負荷を分析します。

チャートで特定のグループを選択すると、下部に**そのグループに属するSQLリスト**（SQL List）が表示されます。これによりグループ→SQL段階でdrill-downして個別Top SQLまで詳細分析できます。

推移チャート

選択した指標タブ基準でTop N個のSQLの推移を時系列チャートで表示します。どのSQLが該当指標値が高いか、どの時間帯に高いかを一目で確認できます。チャート下部の凡例で各SQLを確認でき、凡例またはチャートをクリックすると該当SQLラインがハイライトされます。

照会時間範囲に応じてチャートのデータ間隔が自動調整されます。

照会範囲	チャートインターバル
3時間以下	1分または5分（ <code>sqlstat_send_interval</code> 値により決定）
1日以下	5分
30日以下	1時間

クエリ別チャート

テーブル項目の**▶ボタンをクリックすると、該当個別クエリの実行回数と平均実行時間**チャートを確認できます。チャートインターバルは上記基準と同じです。

- マウスをオーバーすると時間別の数値を確認できます。
- チャートの棒グラフをクリックすると、該当時間の**インスタンスモニタリング**画面が新しいウィンドウで開きます。該当SQLが実行された時間帯のインスタンス全体とアクティブセッションを一度に確認できます。

データ解釈ガイド

主な指標

カラム	意味	解釈ポイント
sql	SQL文（リテラル値除く）	どのクエリかを確認

カラム	意味	解釈ポイント
instance	実行されたインスタンス名	どのDBで実行されたかを確認
execute count	実行回数	高いほど頻繁に呼び出されるSQL → 最適化効果が大きい
elapsed sum (sec)	総累積実行時間 (秒)	高いほど全体負荷が大きいSQL
wait elapsed sum (sec)	総累積待ち時間 (秒)	高いほど待ちが長いSQL → ロック、I/Oなどのボトルネックを確認
elapsed avg (sec)	平均実行時間 (秒) = elapsed sum / execute count	高いほど個別実行が遅いSQL → チューニング優先対象
elapsed max (sec)	最大実行時間 (秒)	断続的なパフォーマンス低下の検出 → 最悪ケースの把握

チューニング対象SQLの識別

- elapsed sum が高く execute count も高いSQL
頻繁に実行されながら全体負荷が大きい → 最優先チューニング対象
- elapsed avg が高いSQL
個別クエリ自体が遅い → SQLチューニング、インデックス確認
- elapsed avg は低い execute count が高いSQL
個別は速いが頻繁に呼び出されすぎ → キャッシュ、呼び出しロジックの改善
- wait elapsed sum が高いSQL
待ち時間が長く全体実行時間が遅い → ロック競合、I/Oボトルネックなどを確認
- elapsed max と elapsed avg の差が大きいSQL
普段は速いが特定の条件で急激に遅くなる → ロック競合、データ偏りなどを確認

実践活用シナリオ

シナリオ1. 全体負荷が大きいSQLを探す

1. `elapsed sum (sec)` 基準で降順ソート
2. 上位SQLがDB全体の負荷に最も大きな影響を与えるクエリ
3. `execute count` と `elapsed avg` を合わせて確認し判断
 - `elapsed avg`が高い場合: 個別クエリ自体が遅い → SQLチューニング、インデックス確認
 - `elapsed avg`は低い`execute count`が高い場合: 頻繁に呼び出されすぎ → キャッシュ、呼び出しロジックの改善

シナリオ2. 待ち時間が長いSQLを探す

1. `wait elapsed sum (sec)` 基準で降順ソート
2. 待ち時間が長いSQLを識別
3. ロック競合、I/Oボトルネック、コネクション待ちなどの原因を分析

シナリオ3. 遅いSQLを探す

1. `elapsed avg (sec)` 基準で降順ソート
2. 個別実行が遅いSQLを識別
3. 該当SQLのPlan（実行計画）を確認してチューニングポイントを決定

シナリオ4. 断続的なパフォーマンス低下の検出

1. `elapsed max (sec)` と `elapsed avg (sec)` の差が大きいSQLを確認 → 普段は速いが特定の条件で急激に遅くなるSQL
2. ▶ボタンでクエリ別推移チャートを開き遅くなった時間帯を確認
3. 該当時間帯の棒グラフをクリックしてインスタンスモニタリングで原因分析

シナリオ5. 特定時間帯の異常分析

1. 推移チャートで異常に高い区間を確認
2. 該当時間帯をクリックしてインスタンスモニタリング指標を確認
3. どのSQLが該当時間帯に集中的に実行されたかを分析

追加機能

SQL詳細情報の確認と条件ベースのフィルター設定ができます。

SQL詳細表示

SQLカラムをクリックすると詳細情報ウィンドウが開きます。SQL詳細ウィンドウの説明は以下を参照してください。

フィルターを追加する

1. フィルター追加ボタンをクリックしてください。
2. フィルターキーを選択してください。
 - 文字 (String) : `=` (含む)、`!=` (含まない)
 - 数値 (Number) : `==` (等しい)、`>=` (以上)、`<=` (以下)
3. 値を入力した後、適用してください。
4. 複数条件はANDロジックを適用してください。

PG SQL統計

ⓘ PG SQL統計は今後SQL統計（DB）に名称が変更される予定です。

PostgreSQLの `pg_stat_statements` ビューで提供する**DB自体のSQL統計**です。SQL別の実行回数、実行時間、処理Row数、ブロックI/Oなどのデルタ値をDBに直接照会することなく画面で確認できます。

- エージェントが1時間間隔でデルタ値を計算して収集します。
- エージェントオプション `statements_order_by` 基準（デフォルト値 `total_time` ）で `statements_row_limit` まで収集します。（デフォルト値：5,000件）
- DBXエージェント1.6.10バージョン以上が必要です。

ⓘ 設定完了後、約1時間後にデータ収集が開始されます。

事前設定

画面にデータを表示するには、**DB設定とエージェント設定**を必ず設定する必要があります。

DB設定

1. `pg_stat_statements`拡張のインストール

`pg_stat_statements` はPostgreSQLのcontribモジュールに含まれています。インストールされていない場合はパッケージをインストールします。

```
# Red Hat / CentOS
yum install postgresql-contrib

# Debian / Ubuntu
apt-get install postgresql-contrib
```

2. `postgresql.conf`の設定

`shared_preload_libraries` に `pg_stat_statements` を追加します。変更後、PostgreSQLの再起動が必要です。

```
shared_preload_libraries = 'pg_stat_statements'
```

3. 拡張モジュールの作成

モニタリング対象データベースで拡張を作成します。

```
CREATE EXTENSION pg_stat_statements;
```

4. インストール確認

```
SELECT * FROM pg_stat_statements LIMIT 1;
```

エージェント設定

PostgreSQLでは `statements` オプションのデフォルト値が `true` のため、別途設定なしでSQL統計がデフォルト収集されます。収集を無効にするには `statements=false` に設定してください。

設定項目	タイプ	デフォルト値	説明
<code>statements</code>	boolean	<code>true</code>	SQL統計の収集有無を設定します。

詳細オプション

設定項目	タイプ	デフォルト値	説明
<code>statements_interval</code>	int	1	SQL統計収集間隔（時間単位）。 <code>statements_interval=1</code> に指定した場合、1時間周期で収集します。
<code>statements_interval_min</code>	int	0	SQL統計収集間隔（分単位）。



設定項目	タイプ	デフォルト値	説明
			<code>statements_interval_min=10</code> に指定した場合、10分周期で収集します。このオプションを設定すると <code>statements_interval</code> より優先されます。
<code>statements_min_row</code>	int	10000	SQL統計データを収集する基準。下記の収集基準を参照してください。
<code>statements_row_limit</code>	int	5000	収集最大件数。 <code>statements_order_by</code> 基準の上位件数まで収集します。
<code>statements_order_by</code>	String	"total_time"	SQL統計情報を収集する際のソート基準を指定します。（例： <code>statements_order_by=total_time,temp_blks_read</code> ）
<code>statements_schema</code>	String	""	<code>pg_stat_statements</code> 拡張が whatap.conf の db オプションに記入したDBにインストールされていない場合、拡張がインストールされているDB名を指定します。

収集基準（ `statements_min_row` ）

`pg_stat_statements` の `rows`（該当クエリで出力または影響を受ける総Row数）が `statements_min_row` を超える場合に、該当SQLデータを収集します。

ⓘ DBワークロードによってSQL収集量が多すぎるまたは少なすぎる場合、
`statements_min_row` と `statements_row_limit` の値を調整して収集量を制御できます。

SQL統計（DB）を分析する

基本オプション

上部フィルター領域で照会期間、対象インスタンス、フィルター条件などを設定できます。

- **時間:** 照会する日時を選択
- **インスタンス:** 照会対象DBインスタンスを選択
- **フィルター:** 条件別データフィルタリング（複数条件はANDロジックを適用）
- **ソート順:** データをソートする基準カラムを選択（例： `rows`、`total_time`、`calls` など）
- **照会件数:** テーブル表示件数を設定
- **結果内検索:** 照会された結果内でクエリキーワード検索
- **☐☐☐（カラム選択）:** 表示するカラムを追加/削除、ドラッグで順序変更可能
- **📄 CSV（ダウンロード）:** CSV形式でエクスポート

カラムガイド

基本情報

カラム	説明
<code>query</code>	正規化されたSQL文（リテラル値が\$1、\$2などに置換される）
<code>queryid</code>	同じ正規化クエリを識別するハッシュコード
<code>username</code>	クエリを実行したユーザー名
<code>db</code>	クエリが実行されたデータベース名
<code>instance</code>	SQLが実行されたDBインスタンス名

実行および時間

カラム	説明
<code>calls</code>	SQL実行回数
<code>total_time</code>	総実行時間（単位：ミリ秒）
<code>rows</code>	クエリで出力または影響を受けた総Row数

ブロックI/O（共有バッファ）

カラム	説明
<code>shared_blks_hit</code>	共有バッファキャッシュから読み取ったブロック数（キャッシュヒット）
<code>shared_blks_read</code>	ディスクから読み取った共有ブロック数（キャッシュミス）
<code>shared_blks_dirtied</code>	変更された共有ブロック数
<code>shared_blks_written</code>	ディスクに書き込んだ共有ブロック数

ブロックI/O（ローカルバッファ）

カラム	説明
<code>local_blks_hit</code>	ローカルバッファキャッシュから読み取ったブロック数
<code>local_blks_read</code>	ディスクから読み取ったローカルブロック数
<code>local_blks_dirtied</code>	変更されたローカルブロック数
<code>local_blks_written</code>	ディスクに書き込んだローカルブロック数

一時ブロック

カラム	説明
temp_blks_read	読み取った一時ブロック数
temp_blks_written	書き込んだ一時ブロック数

I/O時間

カラム	説明
blk_read_time	ブロック読み取りに費やした時間（単位：ミリ秒）。track_io_timingの有効化が必要。
blk_write_time	ブロック書き込みに費やした時間（単位：ミリ秒）。track_io_timingの有効化が必要。

テーブルFetch

カラム	説明
table fetch by rowid	インデックスを通じてテーブルRowを直接照会した回数
table fetch continued row	Rowが複数のブロックにまたがって保存されている場合に追加ブロックを読み取った回数

データ解釈ガイド

チューニング対象SQLの識別

- total_time が高く calls も高いSQL
頻繁に実行されながら遅い → 最優先チューニング対象
- rows が高いSQL

大量のRowを処理 → 不必要な照会でないか確認

- `total_time / calls` (平均実行時間) が高いSQL

個別実行が遅い → SQLチューニング、実行計画の確認

キャッシュ効率の確認

- キャッシュヒット率 = $\text{shared_blks_hit} / (\text{shared_blks_hit} + \text{shared_blks_read})$
 - 90%以下の場合、`shared_buffers` の増設または非効率なクエリの確認
- `shared_blks_read` が高いSQL: ディスクI/Oを多く引き起こす → インデックス確認、`shared_buffers` 増設を検討
- `shared_blks_dirtied` が高いSQL: データを多く変更 → チェックポイント負荷を引き起こす可能性あり

一時ブロックの確認

- `temp_blks_read`、`temp_blks_written` が高い場合、ソートやハッシュジョイン時にメモリが不足してディスクを使用中
- `work_mem` の増設を検討

I/O時間の確認

- `blk_read_time` が高いSQL: ディスク読み取り待ちが大きい → ストレージパフォーマンスまたはキャッシュ効率を確認
- `blk_read_time` の確認のために `postgresql.conf` に `track_io_timing = on` の設定が必要

実践活用シナリオ

シナリオ1. 遅いSQLを探す

1. `total_time` 基準で降順ソート
2. 上位SQLの `calls` を確認 → 頻繁に実行されながら遅いSQLが最優先チューニング対象

シナリオ2. キャッシュ効率が低いSQLを探す

1. `shared_blks_read` が高いSQLを確認
2. `shared_blks_hit` と比較してキャッシュヒット率を計算
3. キャッシュヒット率が低い場合、該当SQLのPlan（実行計画）を確認してインデックス追加の有無を決定

シナリオ3. ディスクI/O過多のSQLを探す

1. `blk_read_time` が高いSQLを確認（`track_io_timing = on` が必要）
2. → ディスク読み取りに時間を多く消費するSQL → インデックス確認、不必要なシーケンシャルスキャンを除去

追加機能

SQL詳細表示

SQLカラムをクリックすると詳細情報ウィンドウが開きます。SQL詳細ウィンドウの説明は以下を参照してください。

DB参考情報

`pg_stat_statements`

この画面のデータはDBの `pg_stat_statements` ビューから取得するため、DB設定変更や統計初期化が収集データに直接影響します。

`pg_stat_statements.max`

このパラメータは、異なるSQLパターン（queryid）を最大何個まで保存するかを決定します。デフォルト値は `5000` です。

```
-- 例：以下の2つのSQLは同じqueryidに正規化されます。
```

```
SELECT * FROM users WHERE id = 10;
```

```
SELECT * FROM users WHERE id = 20;
```

```
→ SELECT * FROM users WHERE id = $1
```

保存可能なSQLパターン数がlimitを超えると、最も実行回数が少ないSQLから削除されます。この場合、一部のSQLの統計が欠落する可能性があります。

```
-- 現在のlimitを確認
```

```
SHOW pg_stat_statements.max;
```

```
-- 現在保存されているSQLパターン数を確認
```

```
SELECT COUNT(*) FROM pg_stat_statements;
```

SQLパターン数がlimitに近づいたら `pg_stat_statements.max` の値を増やしてください。変更後、PostgreSQLの再起動が必要です。

`pg_stat_statements_reset()` 初期化

`pg_stat_statements_reset()` 関数で統計を初期化すると、既存データがすべて削除され、新しいデータが記録されます。エージェントが収集するデルタ値も初期化されるため注意してください。

```
SELECT pg_stat_statements_reset();
```

データベースのサイズ

ホーム画面 > プロジェクト選択 > 統計/レポート > DBサイズ

データベースごとのテーブルのサイズを集計し、推移情報をチャートで提供します。データベースの容量が増加する傾向を識別して管理できます。容量が急速に増加する場合を事前に把握して備えられます。

データベースのサイズは、性能にも大きな影響を与える可能性があります。サイズの増加により、クエリの実行時間が長くなる可能性があるため、性能問題の解決を準備できます。間違ったクエリやデータベース構造によりサイズが増加する可能性のある問題を事前に把握して予防することも必要です。

データベースサイズの推移により、使用量のパターンとデータの成長が理解できます。将来の容量要件を予測し、対策を講じられます。これらの変化をモニタリングすることは、リソースとコストを最適化するのに役立ちます。

使用する前に

権限を割り当てる

全データベースサイズは、`pg_database_size` 関数で照会するため、モニタリングアカウントにconnect権限または`pg_read_all_stats` 権限が必要です。

個別テーブルのサイズは、`pg_relation_size`、`pg_indexes_size`、`pg_total_relation_size` 関数で照会するため、その照会権限が必要です。

```
grant usage on schema {schema_name} to whatap;  
grant select on all tables in schema {schema_name} to whatap;
```

- ❗ エージェント設定で `skip_schema`、`skip_db` のオプションに該当する場合は、データベースのサイズ情報を収集しません。
- 収集サイクルは1時間に1回収集します。

エージェントオプションを設定する

データベースサイズの情報を収集するために、次のエージェントオプションを設定できます。

- **tables** Boolean

既定値 true

テーブルサイズの情報を収集します。統計/レポート > DBサイズ メニューでそのデータベースを選択すると、テーブルサイズ情報が確認できます。

- **tables_hour** Int

既定値 5

テーブルサイズの情報を保存するために情報を収集する時刻を設定します。既定値は 5 であり、午前5時に収集を始めます。

- **tables_min_size** Int

既定値 10

テーブル情報を収集する場合、設定した値より小さいサイズのテーブル情報は収集しません。テーブルデータを収集する既定値の最小単位は pg_total_relation_size 基準10MBです。

- **skip_schema** String

既定値：pg_catalog、sys、information_schema、pg_toast

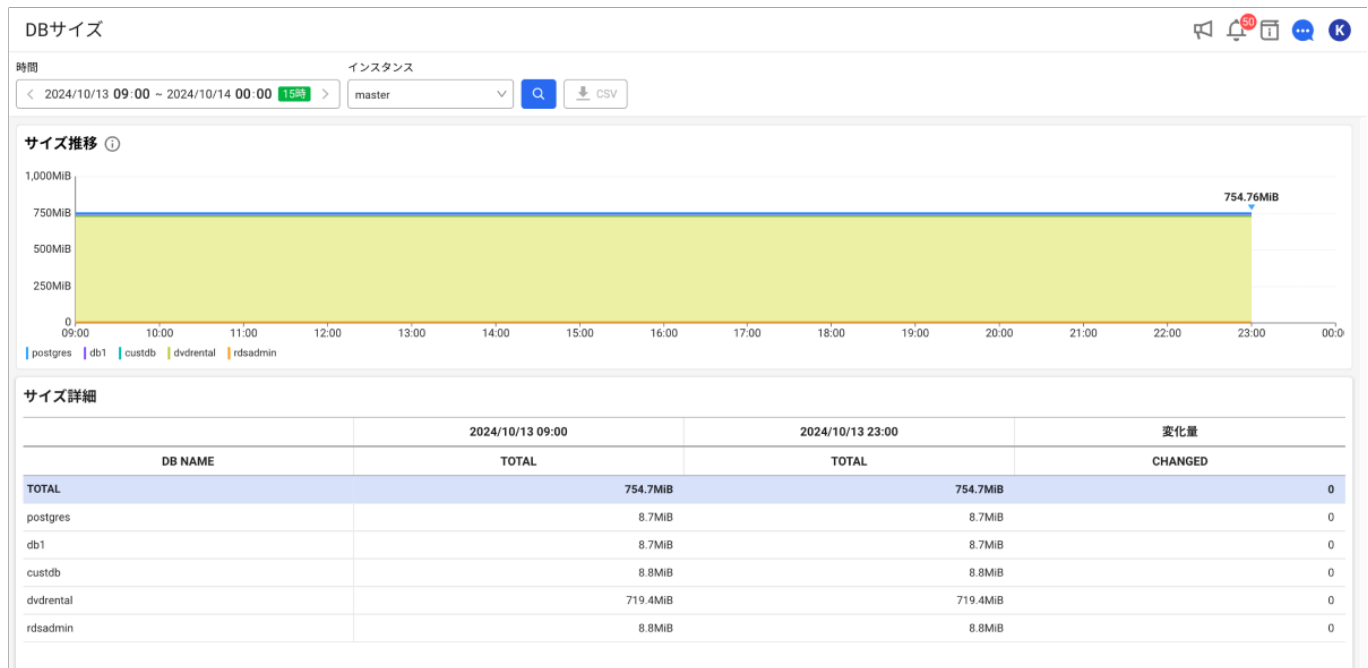
オプション値に設定したスキーマのテーブルは収集しません。

- **skip_db** String

既定値：edb、postgres

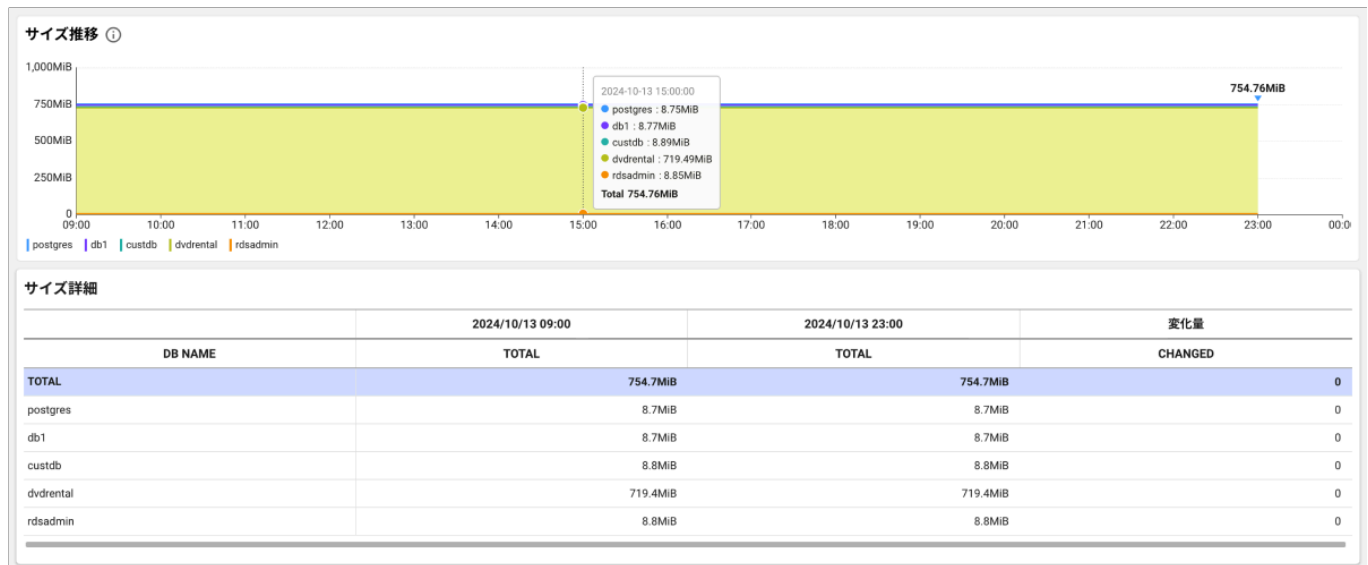
オプション値に設定したデータベースのテーブルは収集しません。

基本画面ガイド



1. 時間で照会したい時間を設定します。緑のボタンを選択して照会時間を選択することもできます。
2. インスタンスで照会したい対象を選択します。
3. 🔍 ボタンを選択します。

時間別変化量を確認する



・ サイズ推移

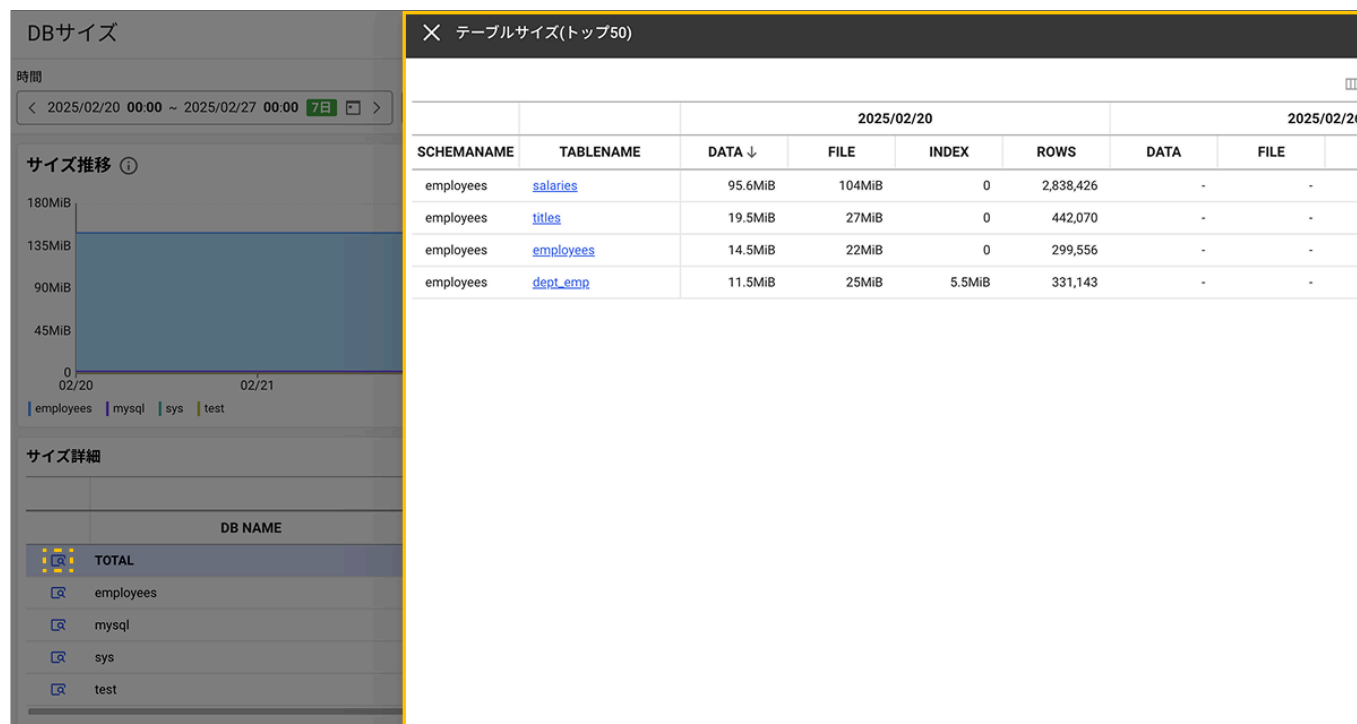
時間帯ごとに、データベースの全体的なサイズと使用量の変化を視覚的に確認できます。チャートで各データベースのサイズを色分けして表示し、マウスをチャートの上にオーバーした状態で左右に移動します。特定の時点の各データベースサイズをツールチップで詳細に確認できます。

・ サイズ詳細

データベースごとの容量の詳細をテーブル形式で提供します。各データベースの名前(DB NAME)および合計容量(TOTAL)、変更量(CHANGED)を確認できます。特定のデータベースを選択すると、テーブルサイズ(トップ50)セクションで詳細情報を確認できます。

テーブル別のサイズを確認する

サイズ詳細セクションでデータベースを選択すると、そのデータベース内のサイズTOP50に該当するテーブルごとのサイズを一覧で確認できます。



データベース内のテーブルごとにスキーマ名(SCHEMANAME)およびテーブル名(TABLENAME)、テーブルサイズ(TABLE SIZE)、インデックスサイズ(INDEX SIZE)、合計容量(TOTAL SIZE)を照会できます。照会開始時点と終了時点の変化量を確認できます。これにより、容量の大きいテーブルを簡単に把握し、容量最適化のための管理に役立ちます。

テーブルサイズ(トップ50)セクションのテーブル一覧でTABLENAMEカラム項目を選択すると、columnおよびindex構成が確認できるObject Detail画面が表示されます。

- ⓘ この機能は、DBXエージェント1.6.15バージョン以降に対応しています。データベース権限に関する設定は、[次の文書](#)を参考にしてください。
- テーブルサイズ(トップ50)セクションでテーブルカラムの順番を変更したり、特定のカラムを非表示にするには、右上に☐ボタンを選択します。カラム設定の詳細については、[次の文書](#)を参照してください。

照会データのダウンロード

照会したデータをCSV形式のファイルでダウンロードできます。 [📄 CSV](#) ボタンを選択した後**サイズ詳細**または**テーブル**

サイズ(トップ50)を選択します。

レポート

ホーム画面 > プロジェクト選択 > 統計/レポート > レポート

画面の上に**レポート**で▽ボタンをクリックすると、日刊、週刊、月刊に分類された各種分析レポートを確認できます。希望するレポート様式を選択し、照会したい日付と時間、フィルターオプションを設定してください。 ボタンを選択すると、選択したレポートを確認できます。

レポートは、個々のプロジェクトのサービスの使用状況と障害の発生履歴を報告する文書です。サービスモニタリング担当者は、レポートを通じて関連部署の担当者とステータスを共有します。

モニタリングデータ分析は、サービスの改善方向を定める指標となるため、極めて重要です。ただし、複数のダッシュボードからデータを収集して文書化するには煩雑な場合があります。

WhaTapの**レポートメニュー**は、**レポート作成業務の自動化**をサポートします。毎週報告書を作成すること、決まった時間に報告書を共有すること、様々な書式を管理することなど、すべて**レポートメニュー**から解決できます。

✔ レポートダウンロード

レポートのダウンロード、印刷またはメール送信の予約を希望する場合は ダウンロードボタンをクリックしてください。html形式でダウンロードできます。

レポートの種類

基本レポートは、大手企業、公共機関やITサービス企業などで実際に使用しているフォームです。ご要望のフォームを作成したい場合は、support@whatap.ioまでお問い合わせください。要望されたレポートは、**統計/レポート > レポート一覧**に追加されます。

メール送信予約

定期的にレポートをメールで受信する場合は、**レポートのメール送信予約**を選択します。出社直後行っていたサービスのチェック作業をメール確認で代わりにできます。

レポートのメール送信予約

×

① 登録した翌日の04~06時にレポートが送信されます。

② 最大5件までのレポート送信が予約できます。

レポートタイプ

Report on Daily Report

▼

レポート時刻

00:00

▼

~

23:59

▼

送信曜日

☒ Mon

☒ Tue

☒ Wed

☒ Thu

☒ Fri

☐ Sat

☐ Sun

メール

メールを入力してからEnterする

👤

×

+ 保存

目次レポート：データベース

日次データベースレポートは、1日あたりのデータベースのパフォーマンスを数値とチャートで確認できるレポートです。レポート作成時に必要に応じて開始日、開始時間、終了時間を設定することができます。特定のインスタンスのみ選択して表示することができます。すべてを選択すると、すべてのインスタンスを一度に表示します。

性能推移チャート

データベースインスタンスごとに指標チャートを確認できます。各指標の最大値を円形で表示し、最大値が発生した時間をまとめて表示します。

- ① チャートで表現される指標は、Config設定によって異なる場合があります。
- 各チャートの右上にあるCSVボタンを使用して、データをエクセルファイルとしてダウンロードすることができます。

SQL TOP 10

検索期間中に全体または選択したインスタンスが実行したSQLクエリ統計を確認できます。クエリ実行時間の最大値を降順にソートし、上位10項目のみ表示します。

週次レポート：データベース

週次データベースレポートは、1週間のデータベースのパフォーマンスを数値とチャートで確認できるレポートです。開始日を直接設定することができ、参照日の7日前から表示できます。

性能推移チャート

データベースインスタンスごとに、週間指標を数値およびチャートで比較できます。検索期間の最大値を円形で表示し、その数値が発生した時間も表示されます。

- ① • チャートで表現される指標は、Config設定によって異なる場合があります。
- 各チャートの右上にある[CSV](#)ボタンを使用して、データをエクセルファイルとしてダウンロードすることができます。

SQL TOP 10

検索期間に全体または選択したインスタンスが実行したSQLクエリ統計を確認できます。クエリ実行時間の最大値(elapse max)を降順にソートし、上位10項目のみを表示します。

月次レポート：データベース

月間のデータベースの性能を数値とチャートで確認できるレポートです。開始日を設定でき、開始日から30日まで表示できます。月間の各指標別の変化量をインスタンス別に確認することができ、検索期間中に実行した上位10項目のSQLクエリ統計も確認できます。

ログ

WhaTapのログモニタリングは、システム統合を通じて、優れた利便性とログデータへのスムーズなアクセスを提供します。独自の技術力による確実なデータ収集を強みとし、ライブテール、ログトレンド、ログ検索、イベント通知、パーサーなどの主要機能をより効率的に活用いただけます。

ログメニューは、照会およびクエリの分析とオプション設定などの機能を提供します。イベント設定メニューでは、ログ関連のイベント通知が設定できます。

- ホーム画面 > プロジェクト選択 > **ログ**

WhaTapモニタリングサービスのメイン画面からプロジェクトを選択し、プロジェクトメニューにある**ログメニュー**を選択します。次の機能を活用すると、複雑なログにより簡単にアクセスができ、様々な条件でリアルタイムに確認および分析ができます。

- **ログTail**
- **ログトレンド**
- **ログ検索**

- ホーム画面 > プロジェクト選択 > **アラート通知** > **イベント設定**

WhaTapモニタリングサービスのメイン画面でプロジェクトを選択し、プロジェクトメニューにある**アラート通知**メニューを選択し、**イベント設定**メニューを開きます。イベント条件を設定し、メール、SMS、メッセージャー、App Pushなどの様々な方法で通知を受信できます。

WhaTapログモニタリングサービスの主要なメニューガイドを次のように提供します。

適用方法

データベースとAWS RDSのログを収集する方法をご案内します。

 ライブテール

ログモニタリングのライブテールについて説明します。

 ログエクスプローラー

ログモニタリングのログエクスプローラーメニューについて案内します。

 ログトレンド

ログモニタリングのログトレンドメニューについて案内します。

 ログの検索

ログモニタリングのログ検索メニューについて説明します。

 ログの設定

ログモニタリングの設定方法について説明します。

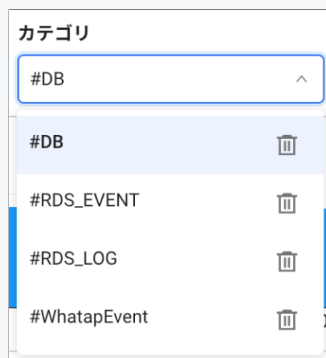
 ログパーシング

WhaTapが提供するログパーサについて案内します。

適用方法

WhaTapは、データベースのログモニタリングを提供します。サーバーインストール型データベースと、クラウドサービスであるAWS RDSのログを収集できます。これにより、WhaTapのログモニタリング機能である**ログTail**、**ログトレンド**メニューからデータベースの散在したログをひと目で確認できます。

- ✔ ログモニタリングを適用した後、各メニューの**カテゴリ**項目とログ一覧からインストール型DBは **#DB**、AWS RDSは **#RDS_LOG** および **#RDS_EVENT** と表示されます。



サーバーインストール型DBに適用する

サーバーにインストールされたデータベースのログをモニタリングするには、WhaTapの**xos.conf**ファイルに次のオプションを適用します。

xos.conf

```
file=/opt/postgresql/logs/postgresql.log
file1=/opt/postgresql/data/logs/pg10/postgresql-Sun.log
file2=/opt/postgresql/data/logs/pg10/postgresql-Mon.log
file3=/opt/postgresql/data/logs/pg10/postgresql-Tue.log
```

- ❗ 適用する前に、XOSエージェントをインストールしてエージェント設定を行う必要があります。詳細については、[次の文書](#)を参照してください。

ログモニタリングを開始する

すべての設定を完了した後、ログ>ログ設定メニューにアクセスし、ログモニタリングを活性化します。

エージェントの設定とログモニタリング活性化

ログTail、ログトレンド、ログ検索機能で、アプリケーションに点在するログをまとめて確認することができます。

ログモニタリングの設定方法

[インストール型DB](#) AWS RDS

1. XOSエージェントのインストール

LinuxファイルシステムにインストールされたDBのログは、WaタブのXOSエージェントで収集します。
下記のガイドリンクからインストールを進めてください。

[エージェントインストールガイド >](#)

2. xos.confファイルにオプション適用

xos.confファイルに次のオプションを適用してください。

```
log_monitoring: true
log_monitoring_options:
  log_monitoring: true
  log_monitoring_options: true
```

📋 コピー

3. ログモニタリング機能の活性化

設定完了後、ログ監視を有効にするトグルボタンをクリックしてください。



ライブテール

❗ ログ参照権限がない場合は、そのメニューにアクセスできません。

ホーム画面 > プロジェクト選択 > ログ > ログTail

ログTail メニューでは、サーバーコンソールに接続せずに、ログデータストリームをモニタリング画面上で直接確認できます。複雑で大量のログも、フィルター と ハイライト 機能を活用して絞り込み、素早く把握できます。ログデータの取得周期は2秒です。

主な用語は次のとおりです。

- **Category**：ログの収集および照会単位
- **Content**：ログメッセージ
- **Search Key**：ログパーサーの設定によって生成される検索キー
- **Tag**：収集されたログを検索できるタグ

❗ ログテーブルカラムの端をドラッグしてカラムの幅を修正できます。

エージェントオプション

エージェントオプションが設定されている場合、ログレベルを収集し、各ログの左側に色付きの縦線が次のように表示されます。

2023-12-18 14:31:02.563	level	INFO	pcode	2277	agenttime	1702877462042	oid	778873916	category	AppLog	loggerName	io.home.test.logback02starter.base.web.LogbackControllerGreeting
												INFO log in our greeting method.
2023-12-18 14:31:02.563	level	WARN	pcode	2277	agenttime	1702877462043	oid	778873916	category	AppLog	loggerName	io.home.test.logback02starter.base.web.LogbackControllerError
											threadName	http-nio-19090-exec-2
												io.home.test.logback02starter.base.errors.exception.ApiException: [2204] Process failure. Please try again.
2023-12-18 14:31:02.586	level	error	pcode	2277	agenttime	1702877462043	oid	778873916	category	AppServer	event_status	error
											event_status_literal_name	level
												Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.error
2023-12-18 14:31:02.586	level	error	pcode	2277	agenttime	1702877462057	oid	778873916	category	AppServer	event_status	error
											event_status_literal_name	level
												Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.error
2023-12-18 14:31:02.586	level	error	pcode	2277	agenttime	1702877462081	oid	778873916	category	AppServer	event_status	error
											event_status_literal_name	level
												Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.error
2023-12-18 14:31:02.586	level	error	pcode	2277	agenttime	1702877462087	oid	778873916	category	AppServer	event_status	error
											event_status_literal_name	level
												Servlet.service() for servlet [dispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.error

❗ エージェントのオプション設定

```
# whatap.conf
weaving=log4j-2.17
weaving=logback-1.2.8
```

- Javaエージェント2.2.22バージョンからウィービング設定にlog4j-2.17またはlogback-1.2.8設定時に使用できます。エージェントの再起動が必要です。
- ログレベルは、パーシングされたキーワードの中で level 、 type 基準で判別します。 level 、 type にパーシングされたキーが存在し、パーシングの値がFATAL、CRITICAL、ERROR、WARN、WARNING、INFOを含む場合は、ログレベルの色を表示します。

1 フィルター領域

フィルター適用

フィルターを適用すると、入力した条件に合ったログをフィルタリングします。複数のフィルターを入力できます。フィルターのタグが同じ場合、OR(||)、そうでない場合はAND(&&)が適用されます。

入力欄に文字を直接入力するか、**フィルター**入力欄をクリックしてフィルターを指定します。フィルタータグは **検索キー**、**オペレーター**、**検索値** の順に入力します。🔍 **検索** ボタンを選択すると、フィルターが適用されたデータをログ一覧で確認できます。

❗ ガイドUI

次のように入力ウィンドウの下に**ガイドUI**を提供します。入力画面またはフィルタータグにマウスカースールがある場合、**ガイドUI**を閉じる時はESCキーで操作できます。



検索キー、オペレーター、検索値の入力

- **検索キー** 入力時、一般インデックス、予約語インデックス、数字のみ入力できるインデックスを区分して推奨値を提供します。
- **オペレーター** 入力時、一般インデックス検索キーの場合は、`==`、`!=` オプションが下に表示されます。数字のみ入力できるインデックスの場合、`>`、`<`、`<=`、`>=`、`==`、`!=` オプションを提供します。
- **検索値** 入力時の一致検索(`>`、`<`、`<=`、`>=`、`==`)の時は青色で、除外検索(`!=`)の時は赤色でハイライトします。**検索値** 入力時に大文字と小文字の区分オプションを活用して検索することができます。

- ① フィルタータグが2行以上長くなる場合へ **折りたたみ**アイコンを選択して折りたたむことができます。
- フィルタータグを入力した後、入力画面でShiftとTabキーを同時に使用して、以前のフィルタータグに移動できます。
- フィルタータグを入力した後、入力画面でTABキーを通じて次のフィルタータグに移動できます。

フィルタータグ追加

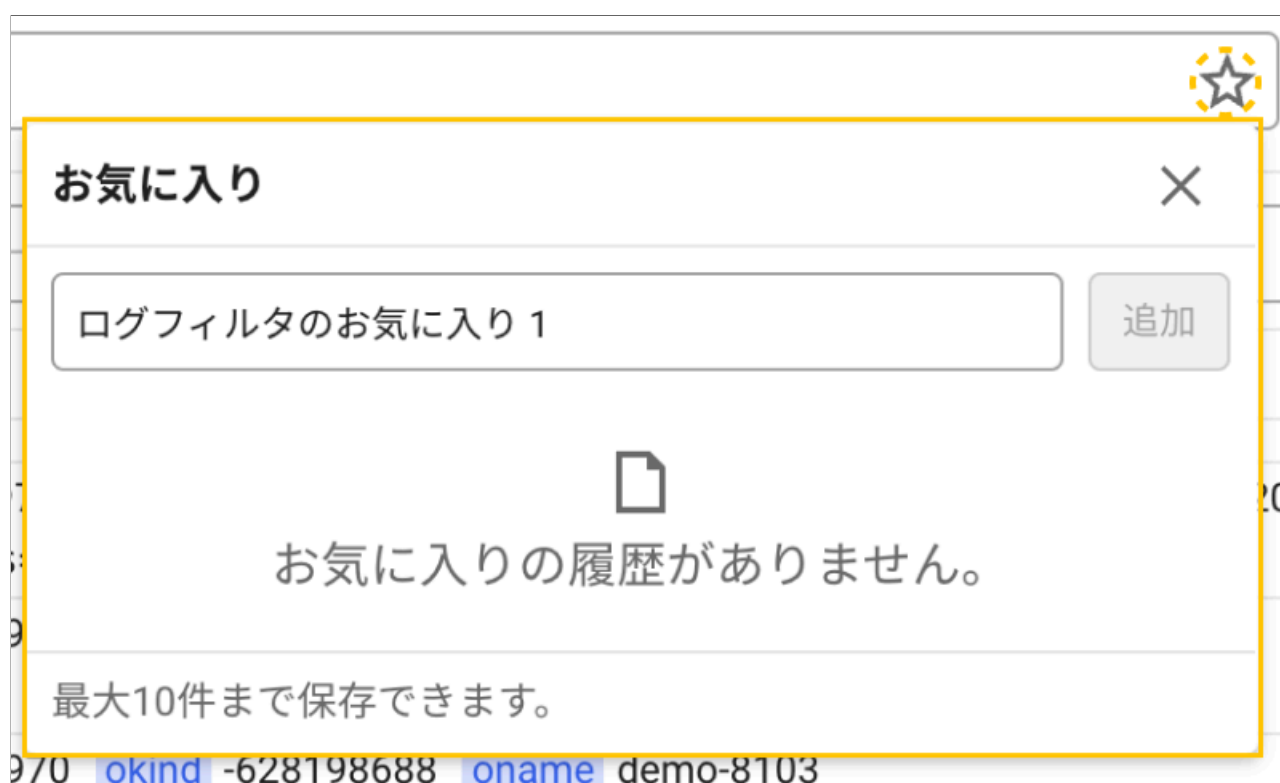
入力フィールドにテキストを入力し、Enter または Tab キーを押すか、入力欄下のガイド UI で推奨値をクリックまたは矢印キーで選択して追加できます。

フィルタータグの削除

単一のタグを削除するには、**Backspace** キーを押すか、タグ上の **X** アイコンをクリックします。すべてのタグを一度に削除するには、入力フィールド右側の **X** アイコンをクリックします。

フィルターのお気に入り

フィルターお気に入り機能を提供します。入力画面の右側にある ☆ お気に入り アイコンを選択して、目的のフィルター検索条件をお気に入りに追加、削除および既存のお気に入りを選択できます。お気に入りは最大50個まで保存されます。



フィルター適用の例外状況

- 数字のみ入力できるインデックス(.n で終わる 検索キー)を入力したタグで、 検索値 は数字のみ入力できます。
- 重複した 検索キー 、 検索値 は入力できません。
- 検索キー 、 検索値 いずれかのタグが存在しない場合は、検索できません。有効でないタグの場合、灰色と表示します。

❗ **ログTail** 検索キー として category は入力できません。

入力フィルター値の下にある数式(expression)は、ログデータを表示するときに適用されるフィルター式のプレビューです。

パースングされていないキーワードのフィルター適用

ログからパースングされていない、つまりインデックスが作成されていないキーワードを含むログを照会できます。この場合、指定範囲内のすべてのログをFull Scanします。そのため、インデックスが生成されたキーに比べて検索速度が多少落ちる可能性があります。定型化されたログデータの場合、**ログパーサの設定**を通じてインデックスキーを活用した検索をお勧めします。

フィルター ⓘ

▼ content == *select* ×

フィルターを入力してから「Enter」キーを押して追加

content == *select*

1. カテゴリを選択します。
2. フィルター入力欄に content を選択後、検索キーワードを入力します。
例) content *select*
3. 🔍 検索ボタンをクリックしてログを照会します。

- ❗ **ログTail**の場合、すべてのログ検索が可能であるため、カテゴリを指定する必要はありません。
- パーサ設定の詳細については、[次の文書](#)を参考にしてください。

フィルター修正

フィルターに値を入力し、入力した値をクリックすると、その値を修正できます。

フィルター ⓘ

🔍 oname == ote-account × key == aws.base:index × appender != fileAppender × フィルターを入力してから「Enter」キーを押して追加 ×

☐ 大小文字を区別する

fileAppender

searchValue*	検索値で始まるログを検索します。 ex key!=searchValue*
*searchValue	検索値で終わるログを検索します。 ex key!=*searchValue
searchValue	検索値を含むログを検索します。 ex key!=*searchValue*
*search*Value*	検索値を含むログを検索します。 ex key!=*search*Value*
re:searchValue	正規表現に一致するログを検索します。 ex key!=re:searchValue
**	検索語に該当するすべてのログを検索します。 ex key!=**

key == value 一致 key != value 除外 key == value 無効なタグです。 ⓘ ガイド表示

- フィルター入力欄にテキストを書き換えて修正できます。
- フィルター入力欄の下に表示されるガイドUIから推奨値を選択して修正できます。

検索キー(Search Key)

次の画像で青いボックス部分はパーシング(parsing)された検索キーです。検索キーはログ設定のログパーサの設定タブでパーシングロジックを登録し、設定できます。

検索キー

フィルター入力の文法

タグは検索キーと検索値で構成されています。次の例で、検索キーは `exception`、検索値は `UnknownHostException` となります。この例は、収集したログデータのうち、IPアドレスとドメインアドレスが一致せず、サーバーをホストに接続できない場合に発生する例外(`UnknownHostException`)が含まれたログデータを照会

します。

exception == UnknownHostException

×

検索キーの種類

検索キーの種類	検索キーフォーマット	意味	検索キーと検索値の例	検索例
文字列キーワード	keyword	ファイル名	- キー: fileName - 値: /data/whatap/logs/yard.log	fileName:/data/whatap/logs/yard.log
数字キーワード	keyword.n	レスポンス時間	- キー: response_time.n - 値: 2945	response_time.n>=2945
予約語キーワード (辞書定義キーワード)	@keyword	TXID	- キー: @txid - 値: 85459614215434144	-
ログ本文キーワード	content	ログ本文	- キー: content - 値: ユーザー入力値	content: *ERROR*

ⓘ Content検索キー

- Content検索キーは、インデックスされていないログの本文を対象に検索します。例えば、`content *ERROR*` のように入力する場合は、ログ本文中で ERROR を含めたログを検索します。
- どのキーワードでインデックスをかけるべきかわからない場合は、Content検索キーを活用して問題となるキーワードを含むログを識別します。その後、**ログ設定メニュー**のログパーサ設定を通じて、該当キーワードでパーサを設定し、インデックスを生成する方式で、検索速度を早くできます。

共通文法

文法の種類	説明	例
<code>==searchValue</code>	検索値と一致するログを検索します。	<code>exception==RuntimeExceptionexception</code>
<code>!=searchValue</code>	検索値を除いたログを検索します。	<code>exception!=RuntimeException</code>
<code>*searchValue</code>	検索値で終わるログを検索します。	<code>word==*hello</code>
<code>searchValue*</code>	検索値で始まるログを検索します。	<code>word==hello*</code>
<code>*searchValue*</code>	検索値として中間に含まれるログを検索します。	<code>word==*hello*</code>
<code>*search*Value*</code>	検索値として含まれるログを検索します。	<code>word==*he*llo*</code>
<code>re:{正規表現}</code>	正規表現にマッチングされるログを検索します。	<code>caller==re:^(i\.w\.a\.w\.s\.v\.r\.</code>
<code>**</code>	検索語に該当するすべてのログを検索します。	

検索キーが数字キーワード(keyword.n)の場合の文法

次の文法は検索キーが、`keyword.n` 形式の場合のみに対応します。

- 検索値は数字のみ使えます。
- `.n` キーワードの値にはPrefixを付けません。`.n` 以外のキーワードは、すべてPrefixを付ける必要があります。

例) `+>searchValue` は有効ではありません。

文法の種類	説明	例
<code>>searchValue</code>	検索値よりも大きな値を含むログを照会します。	<code>response_time.n>3000</code>
<code>>=searchValue</code>	検索値よりも大きい、同じ値を含むログを照会します。	<code>response_time.n>=3000</code>
<code>==searchValue</code>	検索値と同じ値を含むログを照会します。	<code>response_time.n==3000</code>

文法の種類	説明	例
<code>!=searchValue</code>	検索値とは異なる値を含むログを照会します。	<code>response_time.n!=3000</code>
<code><searchValue</code>	検索値より小さい値を含むログを照会します。	<code>response_time.n<3000</code>
<code><=searchValue</code>	検索値より小さいか、同じ値を含むログを照会します。	<code>response_time.n<=3000</code>

個人情報の照会

ログの個人情報照会権限がある場合個人情報の表示  トグルボタンの活性化を通じて非識別化処理した個人情報を表示することができます。

- ① ログの個人情報照会権限についての詳細は[次の文書](#)を参照してください。
- ログ個人情報の非識別化機能についての詳細は[次の文書](#)を参照してください。

2 コンテンツハイライト領域

ログのコンテンツ内で目的のキーワードを簡単に識別できるよう、ハイライト機能を提供します。単一または複数のキーワードでフィルタリングできます。

ログTail

カテゴリー

AppLog

フィルター

content

+=select*

×

フィルターを入力してから「Enter」キーを押して追加します。

content == +=select*

act

🔍

⏸

🔄

🔍

⚙

	oname	タイムスタンプ	ログ
▶	demo-8102	2023-11-24 14:06:43.648	@txid 4785826165275137961 pcode 5490 oname demo-8102 onodeName node-0 oid 1387800924 okindName demo-okind-0 okind -1383513482 select distinct ename, deptno, sal, job from emp
▶	demo-8102	2023-11-24 14:06:43.654	@txid 4785826165275137961 pcode 5490 oname demo-8102 onodeName node-0 oid 1387800924 okindName demo-okind-0 okind -1383513482 select ename, 1000 from emp
▶	demo-8102	2023-11-24 14:06:43.659	@txid -240363117644553212 pcode 5490 oname demo-8102 onodeName node-0 oid 1387800924 okindName demo-okind-0 okind -1383513482 select distinct pp.lastname, pp.firstname
▶	demo-8102	2023-11-24 14:06:43.659	@txid 4785826165275137961 pcode 5490 oname demo-8102 onodeName node-0 oid 1387800924 okindName demo-okind-0 okind -1383513482 select 'total income is', ((orderqty * unitprice) * (1.0 - unitpricediscount)), ' for '

1. ハイライトしたいキーワードをキーワード入力欄に入力します。

- 例: `select`

2. 検索したキーワードがログリストのContent内でハイライト表示されます。

-  **拡大表示** アイコンをクリックすると、**ログ** と **タイムスタンプ** を全画面で確認できます。
- ハイライトされたキーワード間の移動はショートカットで操作できます。
 - `Enter` : 次の行へ移動
 - `Shift+Enter` : 前の行へ移動
 - `↑ / ↓` アイコン: 行の移動
 - `Cmd + F` (Mac) / `Ctrl + F` (Windows): キーワード検索

複数キーワード条件

複数キーワードでハイライトする場合は、次のように作成します。

入力文字列	説明	結果
<code>a b c</code>	各キーワードをスペースで区切ります。	a, b, c
<code>"Whatap is good."</code>	キーワードにスペースを含める場合は、 <code>"</code> または <code>"</code> で囲みます。	Whatap is good.
<code>"Whatap\\ is good."</code>	<code>"</code> で囲まれたキーワードに、 <code>\</code> を含める場合は、 <code>\\</code> と入力する必要があります。	Whatap\ is good.

ハイライトカラー設定

ハイライトするキーワードおよび色を設定できます。

- デフォルトでは、ログレベルに応じたハイライト (**FATAL**、**ERROR**、**WARN**、**SEVERE**、**CRITICAL**) が適用されています。
- 設定した内容は**プロジェクト単位**で保存されます。

1.  アイコンをクリックします。

2. **ハイライト** ウィンドウで、色を設定するキーワードを入力欄に追加してください。
3. 入力欄の左側の色をクリックすると、色を選択できます。

テーブル設定

カラム設定

2 領域の右で  **カラム設定** ボタンを選択します。カラムを追加したり、順序を設定できます。

- **カラム追加**

タグを選択してテーブルにカラムを追加できます。**log**カラム選択を解除する場合**ログ表示詳細設定**が確認できません。必ず1つ以上のカラムを選択します。

- **カラム順序設定**

カラムを追加すると**表示項目**に該当カラムが追加されます。希望のカラムをドラッグしてカラムの順序を変更します。

ログ表示詳細設定

2 領域右で  **ログ表示詳細設定** ボタンを選択します。デフォルトで**content**、**タグ**すべてチェックされており、両方の項目を表示します。**content**と**タグ**どちらかは必ず選択します。

チェックが解除されたアイテムはテーブルに表示されません。次のように**タグ**を解除する場合、テーブルでログの**タグ**は表示されません。

ログテーブルコンテンツタグ非表示

タグ管理一覧にタグを追加すると、追加した順にログのタグがソートされます。タグの順序はドラッグして変更できます。追加したタグを無効にすると、無効にしたタグはログのタグに表示されません。

❗ **カラムおよびログ表示設定の適用範囲**

- **カラム設定**と**ログ表示詳細設定**は、**ログTail**、**ログ検索**、**ログトレンド**で使用できます。
- 同一プロジェクト内の**ログTail**、**ログ検索**、**ログトレンド**は、**カラム設定**と**ログ表示詳細設定**を共有します。

ログファイルのダウンロード

検索したログの結果を **CSV** または **TXT** ファイル形式でダウンロードできます。

1. 画面中央右側の **↓ ダウンロード** ボタンをクリックします。
2. **CSVダウンロード** または **Logダウンロード** をクリックして、希望の形式のファイルをダウンロードします。

ログエクスプローラー

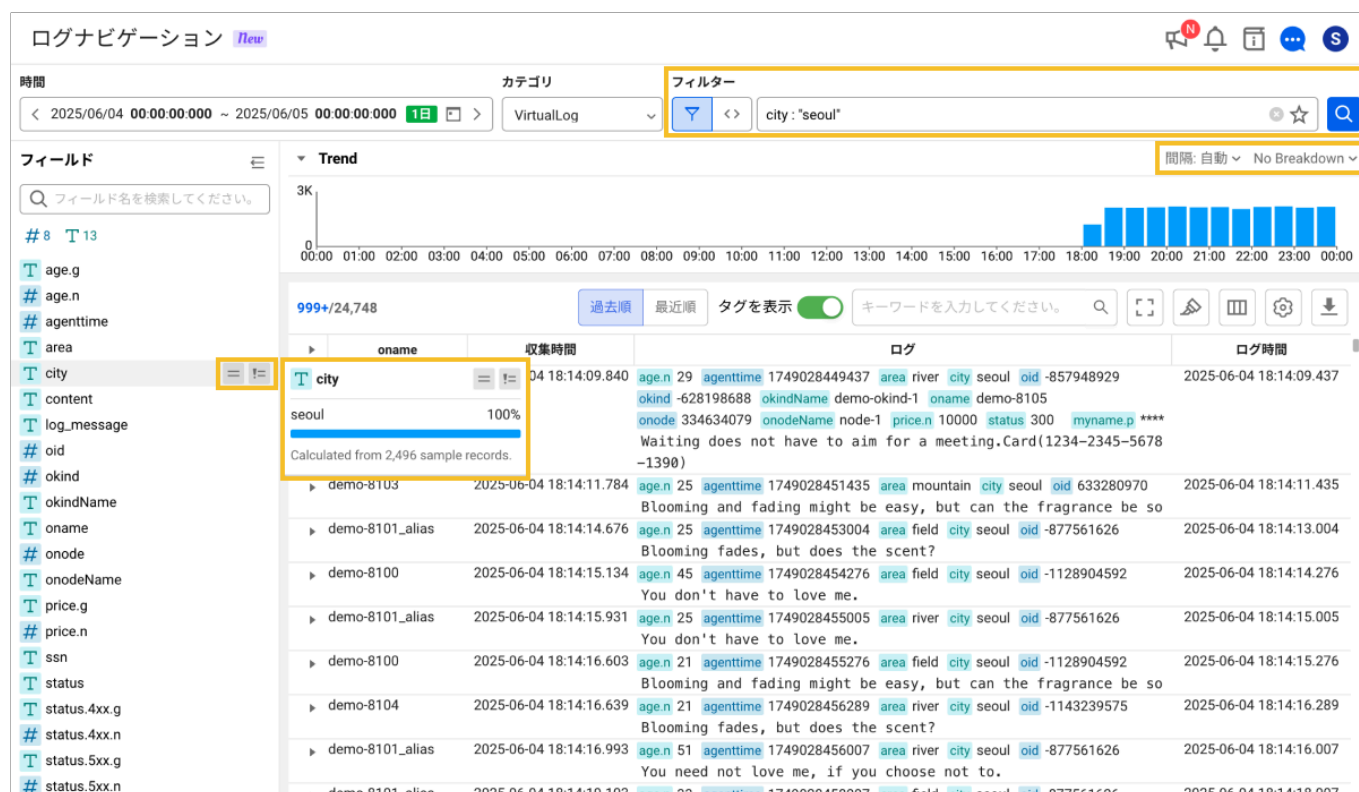
ⓘ ログ参照権限がない場合は、そのメニューにアクセスできません。

ホーム画面 > プロジェクト選択 > ログ > ログエクスプローラー *New*

ログエクスプローラー (Log Explorer) 機能は、ログデータを迅速に検索、フィルタリング、分析します。フィールド構造に関する情報を把握し、クエリベースのフィルタリングによって必要なログを効率的に検索し、結果をデータの可視化を通じて確認できます。本機能はLuceneインデックスをベースに動作し、主な特徴は以下のとおりです。

1. **ログトレンド分析および検索:** ログのトレンドと検索機能を同時にサポートし、時間経過によるログの変化傾向を把握できます。
2. **高速検索およびフィルタリング:** ユーザーは簡単なキーワード入力だけで、問題のあるログを迅速に検索できます。WhaTapログ検索クエリおよびLuceneクエリ言語に対応しており、さまざまな検索条件を設定できます。
3. **オートコンプリート機能:** クエリ入力時にオートコンプリート機能が提供され、ユーザーが簡単にクエリを作成し、習得できるようサポートします。
4. **データ可視化:** スタックグラフを活用して時系列のログ件数データを視覚的に表示し、ドリルダウン機能を通じて特定のログを深く分析できます。
5. **お気に入りと共有:** よく使うフィルターはお気に入りとして保存でき、URLを通じて他のユーザーと簡単に共有できます。

基本画面ガイド



時間設定

上部のオプションバーにある**タイムセレクター**を使用して、ログ検索の時間範囲を設定できます。

カテゴリ

上部のオプションバーにある**カテゴリ選択**で、探索したいログの種類を選択します。（例：AppLogなど）

フィルター

上部オプションバーの**フィルター**でクエリ言語を選択し、検索条件を入力してフィルターを適用します。
検索クエリ構文は、WhaTapログ検索クエリとLuceneクエリの両方をサポートしています。

-  : WhaTapログ検索クエリ
-  : Luceneクエリ

検索クエリ構文

WhaTap log search query

WhaTapのログ検索クエリはオートコンプリート機能を提供しており、簡単なキーワード入力だけで問題のログを迅速に検索できます。and、orなどのさまざまな構文を活用して、さまざまな検索条件を入力できます。

- 例、検索フィルターに `error` と入力するだけで、content（ログ本文）を検索できます。
 1. 上部オプションバーでフィルターアイコンをクリックした後、クエリ検索ウィンドウでフィールド（選択した時間とカテゴリに応じたリストが表示されます）を選択してください。
 2. 演算子を選択してください。
 - a. 基本: `:`、`*`、`and`、`or`
 - b. 選択したフィールドが `Number` 型の場合（基本 + `<=`、`>=`、`<`、`>`）
 - c. 選択したフィールドが `String` の場合（基本のみ）
 3. 値（選択した時間、カテゴリ、フィールドに一致するリストが表示されます）を選択してください。

検索インデックスフィールドタイプ

サポートされているフィールドタイプは、`Number`、`String`、`Boolean` の3種類のみです。

- `Number` : Long、double、float 型の数値
- `String` : メール本文のような全文テキスト
- `Boolean` : True または False の値

 WhaTapログ検索クエリの構文に関する詳細は[次の文書](#)を参照してください。

Lucene

Apache Lucene 8.11.1バージョンのオープンソースの構文がそのまま利用可能です。ただし、自動補完機能は提供していません。

❗ LuceneのQueryParserクラシックパッケージに関する詳細は[公式ドキュメント](#)を参照してください。

フィールド一覧

ドリルダウン分析に活用できます。

フィールド情報のフィルターを反映する

フィールド一覧でフィールド名を確認し、`=` または `!=` ボタンをクリックして条件をフィルターに適用してください。

- フィールドにマウスオーバーすると、`=`、`!=` ボタンが表示され、クリックするとフィルターに反映されます。
- フィールド検索機能を使って、目的のフィールドを素早く見つけることができます。

Trendチャート

Trendチャートでインターバルやブレイクダウンを適用して、ログをグラフで確認できます。

- グラフィインタラクション:** マウสดラッグや  ボタンで時間のズームイン、バークリックで該当領域のログを確認可能
- インターバル:** 自動、1秒、5秒、30秒、1分、5分、10分、30分、1時間から選択可能
- Breakdown:** 特定のフィールドを基準にTop 3 + Othersの値を分布として可視化し、ドリルダウンフィルターの適用が可能

breakdown

Breakdownは、ログ件数の推移を特定のフィールドで分析する可視化機能です。

- 特定のフィールドのTop 3の値とOthers（その他の値）を区別して、スタックグラフで分布を表示します。
- ドリルダウン分析に活用でき、凡例やスタックグラフの特定領域（部分グラフ）をクリックして、特定の値のみを

強調したり、フィルター条件を追加することができます。

ログの確認

検索条件に一致するログの一覧を確認でき、**フィールド一覧**ではフィールド情報とログタグが同じ色で表示されます。

コンテンツハイライト

ログのコンテンツ内で目的のキーワードを簡単に識別できるよう、ハイライト機能を提供します。単一または複数のキーワードでフィルタリングできます。

1. ハイライトしたいキーワードをキーワード入力欄に入力します。
 - 例: `select`
2. 検索したキーワードがログリストのContent内でハイライト表示されます。
 -  **拡大表示** アイコンをクリックすると、**ログ** と **タイムスタンプ** を全画面で確認できます。
 - ハイライトされたキーワード間の移動はショートカットで操作できます。
 - `Enter` : 次の行へ移動
 - `Shift+Enter` : 前の行へ移動
 - `↑ / ↓` アイコン: 行の移動
 - `Cmd + F (Mac) / Ctrl + F (Windows)`: キーワード検索

複数キーワード条件

複数キーワードでハイライトする場合は、次のように作成します。

入力文字列	説明	結果
<code>a b c</code>	各キーワードをスペースで区切ります。	a, b, c
<code>"Whatap is good."</code>	キーワードにスペースを含める場合は、 <code>"</code> または <code>"</code> で囲みます。	Whatap is good.

入力文字列	説明	結果
"Whatap\\ is good."	""で囲まれたキーワードに、\を含める場合は、\\と入力する必要があります。	Whatap\ is good.

ハイライトカラー設定

ハイライトするキーワードおよび色を設定できます。

- デフォルトでは、ログレベルに応じたハイライト（FATAL、ERROR、WARN、SEVERE、CRITICAL）が適用されています。
- 設定した内容はプロジェクト単位で保存されます。
 - 🔗アイコンをクリックします。
 - ハイライト ウィンドウで、色を設定するキーワードを入力欄に追加してください。
 - 入力欄の左側の色をクリックすると、色を選択できます。

カラム設定

☐ ボタンをクリックして、カラムを追加したり順序を設定することができます。

カラム追加

カラム追加：タグを選択して、テーブルにカラムを追加できます。

⚠ ログカラムの選択を解除すると、ログ表示詳細設定を確認できません。必ず1つ以上のカラムを選択します。

カラム順序設定

カラム設定でカラムを追加すると、表示項目に該当カラムが追加されます。希望するカラムをドラッグして、カラムの順序を変更してください。

ログ表示詳細設定

⚙ ボタンをクリックして、ログの表示を詳細に設定します。デフォルトでは **content**、**タグ** の両方がチェックされており、両方の項目が表示されます。

- **content**と**タグ**のいずれかを必ず選択してください。
- **タグ管理**リストにタグを追加すると、追加した順にログのタグが表示されます。タグの順序はドラッグで変更できます。追加したタグを無効にすると、無効にしたタグはログのタグに表示されません。
- チェックが解除された項目は、テーブルに表示されません。次のように**タグ**を解除する場合、テーブルからログの**タグ**は表示されません。

log table contents remove tag sc

ⓘ カラムおよびログ表示設定の適用範囲

- **カラム設定**と**ログ表示詳細設定**は、**ログTail**、**ログ検索**、**ログトレンド**で使用できます。
- 同一プロジェクト内の**ログTail**、**ログ検索**、**ログトレンド**は、**カラム設定**と**ログ表示詳細設定**を共有します。

ログファイルのダウンロード

検索したログの結果を **CSV** または **TXT** ファイル形式でダウンロードできます。

1. 画面中央右側の **↓ ダウンロード** ボタンをクリックします。
2. **CSVダウンロード** または **Logダウンロード** をクリックして、希望の形式のファイルをダウンロードします。

ログトレンド

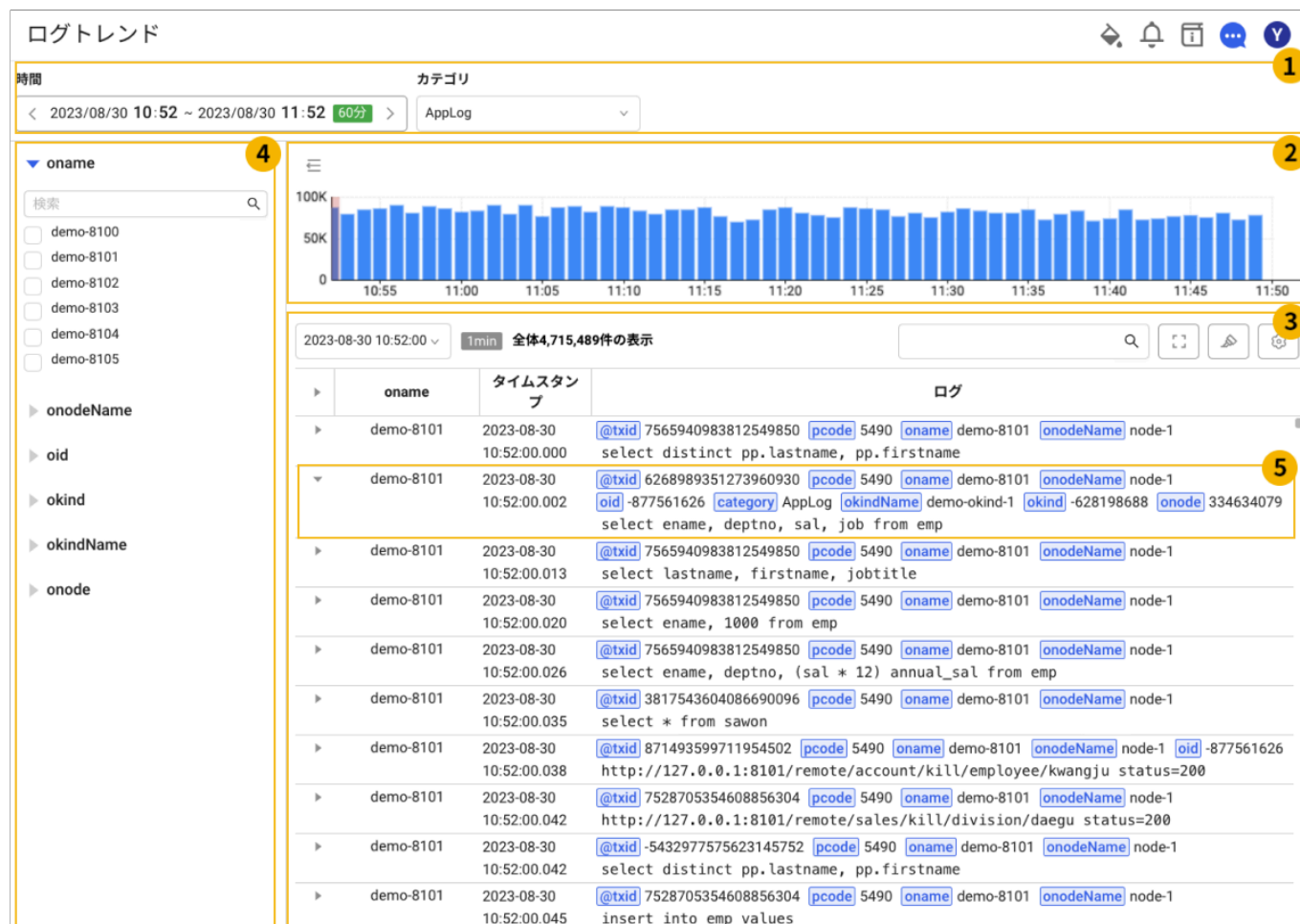
ⓘ ログ参照権限がない場合は、そのメニューにアクセスできません。

ホーム画面 > プロジェクト選択 > ログ > ログトレンド

ログトレンドでは、タイプ別に分類されたログの発生件数の推移を通じて、特定エラータイプの発生パターンを確認し、時間別の詳細ログデータを確認できます。ハイライト機能により、目的のログをすばやく識別できます。カテゴリ別に収集されたログの推移を照会できます。

主な用語は次のとおりです。

- **Category**：ログの収集および照会単位
- **Content**：ログメッセージ
- **Search Key**：ログパーサーの設定によって生成される検索キー
- **Tag**：収集されたログを検索できるタグ



データ参照

- 下部までスクロールすると、次のデータを表示します。
- 1で検索するログデータの時間と収集単位としてカテゴリを指定します。
- カテゴリを変更すると、選択されたカテゴリに対応するログを表示します。2バーチャートと3ログテーブルで確認できます。
- 2棒グラフのバーをクリックすると、バーの時間範囲に対応するログを検索します。
- 3ログテーブルの左上から照会されたログの総数を確認できます。
- 3 ログテーブル上段の右の「全画面」アイコンを選択すると、ログと収集時間を全画面で確認できます。
- 4 サイドメニューからタグでフィルタリングすることでログを確認できます。検索キーは最大2つまで選択でき、

検索値は複数選択できます。

- エージェントオプションが設定されている場合、ログレベルを収集し、各ログの左側に色付きの縦線が次のように表示されます。

2023-12-18 14:31:02.563	level	INFO	pcode	2277	agenttime	1702877462042	oid	778873916	category	AppLog	loggerName	io.home.test.logback02starter.base.web.LogbackControllerGreeting
												INFO log in our greeting method.
2023-12-18 14:31:02.563	level	WARN	pcode	2277	agenttime	1702877462043	oid	778873916	category	AppLog	loggerName	io.home.test.logback02starter.base.web.LogbackControllerError
											threadName	http-nio-19090-exec-2
												io.home.test.logback02starter.base.errors.exception.ApiException: [2204] Process failure. Please try again.
2023-12-18 14:31:02.586	level	error	pcode	2277	agenttime	1702877462043	oid	778873916	category	AppServer	event_status	error
											event_status_literal_name	level
												Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
2023-12-18 14:31:02.586	level	error	pcode	2277	agenttime	1702877462057	oid	778873916	category	AppServer	event_status	error
											event_status_literal_name	level
												Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
2023-12-18 14:31:02.586	level	error	pcode	2277	agenttime	1702877462081	oid	778873916	category	AppServer	event_status	error
											event_status_literal_name	level
												Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
2023-12-18 14:31:02.586	level	error	pcode	2277	agenttime	1702877462087	oid	778873916	category	AppServer	event_status	error
											event_status_literal_name	level
												Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err

❗ エージェントのオプション設定

```
# whatap.conf
weaving=log4j-2.17
weaving=logback-1.2.8
```

- Javaエージェント2.2.22バージョンからウィービング設定にlog4j-2.17またはlogback-1.2.8設定時に使用できます。エージェントの再起動が必要です。
- ログレベルは、パースされたキーワードの中で level 、 type 基準で判別します。 level 、 type にパースされたキーが存在し、パースの値がFATAL、CRITICAL、ERROR、WARN、WARNING、INFOを含む場合は、ログレベルの色を表示します。

- ❗ ログテーブルカラムの端をドラッグしてカラムの幅を修正できます。

個人情報の照会

ログの個人情報照会権限がある場合個人情報の表示  トグルボタンの活性化を通じて非識別化処理した個人情報を表示することができます。

- ❗ ログの個人情報照会権限についての詳細は次の文書を参照してください。
- ログ個人情報の非識別化機能についての詳細は次の文書を参照してください。

ログContentの確認

Content はログメッセージです。ログカラムの1行目には、パースされたキーと値が表示され、2行目にはContent（ログメッセージ）が表示されます。

- **3** ログテーブルの行(ログ)ごとに ▶ 詳細ボタンがあります。 ▶ 詳細ボタンを選択すると **5** のように、ログの全体 contentを確認できます。

ハイライト設定

目的のキーワードを簡単に識別できるように、ハイライト機能を提供しています。🔍 アイコンをクリックし、キーワードを入力してからEnterキーを押して追加してください。

コンテンツハイライト

ログのコンテンツ内で目的のキーワードを簡単に識別できるよう、ハイライト機能を提供します。単一または複数のキーワードでフィルタリングできます。

1. ハイライトしたいキーワードをキーワード入力欄に入力します。
 - 例: `select`
2. 検索したキーワードがログリストのContent内でハイライト表示されます。
 - **[] 拡大表示** アイコンをクリックすると、**ログ** と **タイムスタンプ** を全画面で確認できます。
 - ハイライトされたキーワード間の移動はショートカットで操作できます。
 - `Enter` : 次の行へ移動
 - `Shift+Enter` : 前の行へ移動
 - `^ / v` アイコン: 行の移動
 - `Cmd + F (Mac) / Ctrl + F (Windows)`: キーワード検索

複数キーワード条件

複数キーワードでハイライトする場合は、次のように作成します。

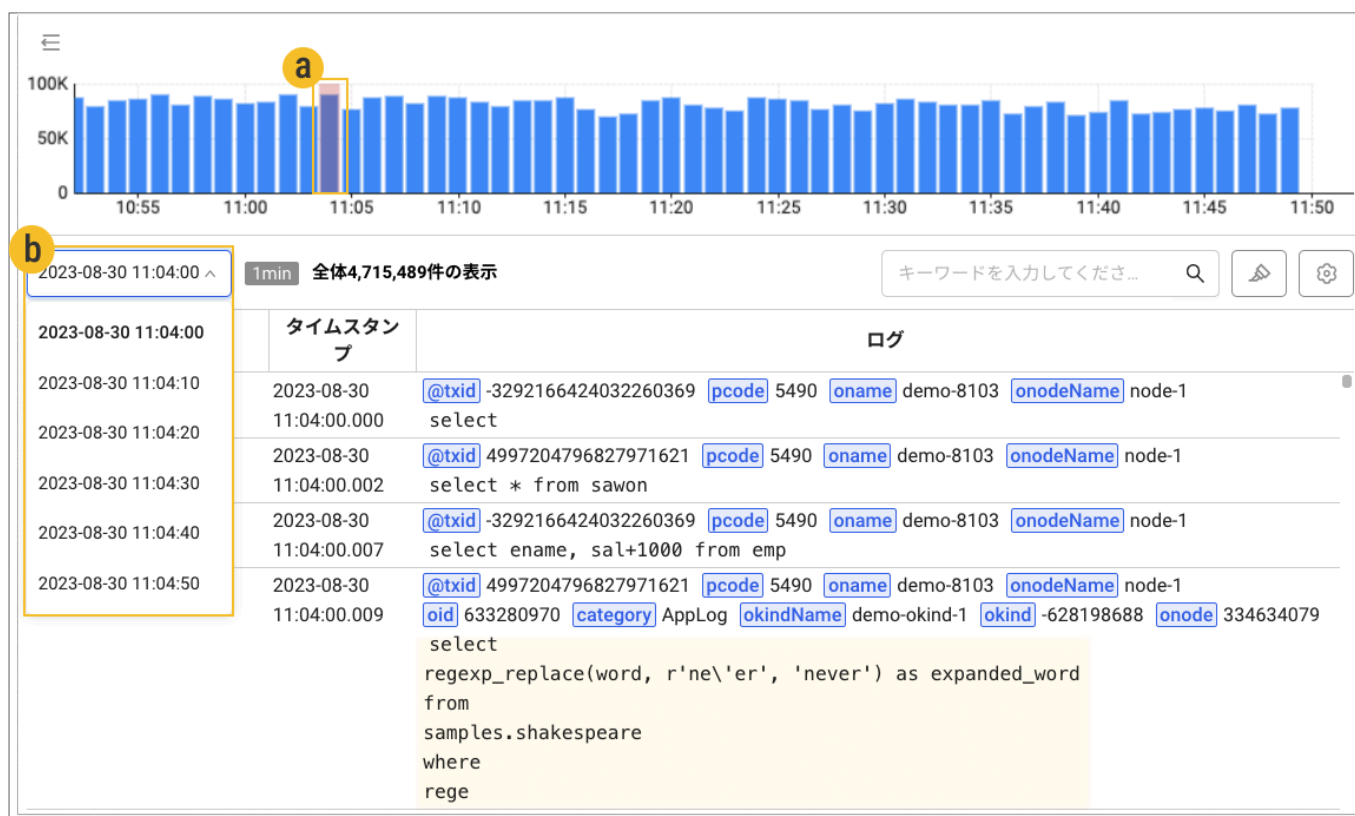
入力文字列	説明	結果
a b c	各キーワードをスペースで区切ります。	a, b, c
"Whatap is good."	キーワードにスペースを含める場合は、" または "" で囲みます。	Whatap is good.
"Whatap\\ is good."	""で囲まれたキーワードに、\ を含める場合は、\\ と入力する必要があります。	Whatap\\ is good.

ハイライトカラー設定

ハイライトするキーワードおよび色を設定できます。

- デフォルトでは、ログレベルに応じたハイライト（FATAL、ERROR、WARN、SEVERE、CRITICAL）が適用されています。
- 設定した内容はプロジェクト単位で保存されます。
 -  アイコンをクリックします。
 - ハイライト ウィンドウで、色を設定するキーワードを入力欄に追加してください。
 - 入力欄の左側の色をクリックすると、色を選択できます。

ログをチャートで検索



- バーチャートで **a** 目的の時間領域バーをクリックまたはドラッグして、その時間範囲のログを確認できます。
- 棒チャートの下ログテーブル上部左側の **b** 時間選択オプションを使用して、次のように選択した時間帯でより詳細なログ検索が可能です。
 - 1min : interval (チャート内の棒の間隔)
 - 時間選択オプション : 選択された時間範囲を6つの間隔に分割した時間帯

テーブル設定

カラム設定

3 領域の右で  **カラム設定** ボタンを選択します。カラムを追加したり、順序を設定できます。

- **カラム追加**

タグを選択してテーブルにカラムを追加できます。**log**カラム選択を解除する場合**ログ表示詳細設定**が確認できません。必ず1つ以上のカラムを選択します。

- **カラム順序設定**

カラムを追加すると**表示項目**に該当カラムが追加されます。希望のカラムをドラッグしてカラムの順序を変更します。

ログ表示詳細設定

3 領域右で  **ログ表示詳細設定** ボタンを選択します。デフォルトで**content**、**タグ**すべてチェックされており、両方の項目を表示します。**content**と**タグ**どちらかは必ず選択します。

チェックが解除されたアイテムはテーブルに表示されません。次のように**タグ**を解除する場合、テーブルでログの**タグ**は表示されません。

ログテーブルコンテンツタグ非表示

タグ管理一覧にタグを追加すると、追加した順にログのタグがソートされます。タグの順序はドラッグして変更できます。追加したタグを無効にすると、無効にしたタグはログのタグに表示されません。

ⓘ カラムおよびログ表示設定の適用範囲

- **カラム設定**と**ログ表示詳細設定**は、**ログTail**、**ログ検索**、**ログトレンド**で使用できます。
- 同一プロジェクト内の**ログTail**、**ログ検索**、**ログトレンド**は、**カラム設定**と**ログ表示詳細設定**を共有します。

ログファイルのダウンロード

検索したログの結果を **CSV** または **TXT** ファイル形式でダウンロードできます。

1. 画面中央右側の **↓ ダウンロード** ボタンをクリックします。
2. **CSVダウンロード** または **Logダウンロード** をクリックして、希望の形式のファイルをダウンロードします。

ログの検索

ⓘ ログ参照権限がない場合は、そのメニューにアクセスできません。

ホーム画面 > プロジェクト選択 > ログ > ログ検索

ログ検索メニューから収集された大量のログをさまざまな条件で検索し、ユーザーが必要なログを指定できます。複数の検索条件をパーシングされたキーとバリューで指定できるため、目的の条件に一致するログデータのみを抽出します。

動的ページで取得したログデータは、設定された行単位で取得され、スクロールで画面の下まで移動すると自動的に次のデータが取得されて表示されます。

主な用語は次のとおりです。

- **Category**：ログの収集および照会単位
- **Content**：ログメッセージ
- **Search Key**：ログパーサーの設定によって生成される検索キー
- **Tag**：収集されたログを検索できるタグ

ログ検索

時間: 2023/08/24 15:10 ~ 2023/08/24 16:10 60分

フィルター: フィルターを入力してから「Enter」キーを押して追加します。

DB (4,623)

- #DB (4,623)
- #RDS_LOG (85)
- #WhatapEvent (5)

全体4,713件の表示

Timestamp: 過去 最近

キーワードを入力してください

oname	タイムスタンプ	ログ
DBX-1-203-5432	2023-08-24 15:10:00.954	2023-08-24 15:09:57 KST [8053]: [1-1] user=kwlee,db=db1,app=,client=[local] LOG: duration: 17814.873 ms statement: s
DBX-1-203-5432	2023-08-24 15:10:06.990	2023-08-24 15:10:02 KST [8201]: [1-1] user=whatap-f,db=selong.sql,app=[unknown],client=[local] FATAL: role "whatap-f" does not exist
DBX-1-203-5432	2023-08-24 15:10:10.001	2023-08-24 15:10:08 KST [8055]: [9-1] user=kwlee,db=db1,app=,client=[local] LOG: duration: 15016.880 ms statement: select (15);
DBX-1-203-5432	2023-08-24 15:10:10.001	2023-08-24 15:10:08 KST [7986]: [11-1] user=kwlee,db=db1,app=,client=[local] CONTEXT: while locking tuple (1,129) in relation "kwlee"
DBX-1-203-5432	2023-08-24 15:10:10.001	2023-08-24 15:10:08 KST [7986]: [10-1] user=kwlee,db=db1,app=,client=[local] LOG: process 7986 acquired ShareLock on
DBX-1-203-5432	2023-08-24 15:10:10.001	2023-08-24 15:10:08 KST [7986]: [12-1] user=kwlee,db=db1,app=,client=[local] STATEMENT: update kwlee set b='67890' wh
DBX-1-203-5432	2023-08-24 15:10:10.001	2023-08-24 15:10:08 KST [8127]: [5-1] user=kwlee,db=db1,app=,client=[local] STATEMENT: update kwlee set b='67890' whe
DBX-1-203-5432	2023-08-24 15:10:10.001	2023-08-24 15:10:08 KST [8127]: [4-1] user=kwlee,db=db1,app=,client=[local] LOG: process 8127 acquired ExclusiveLock
DBX-1-203-5432	2023-08-24 15:10:10.001	2023-08-24 15:10:08 KST [7986]: [13-1] user=kwlee,db=db1,app=,client=[local] LOG: duration: 44040.104 ms statement:
DBX-1-203-5432	2023-08-24 15:10:10.001	2023-08-24 15:10:08 KST [8129]: [4-1] user=kwlee,db=db1,app=,client=[local] LOG: process 8129 acquired AccessExclusiv
DBX-1-203-5432	2023-08-24 15:10:10.001	2023-08-24 15:10:08 KST [8129]: [5-1] user=kwlee,db=db1,app=,client=[local] STATEMENT: select /* this is test */ "a",

データ参照

- 画面の下までスクロールすると、次のデータを検索します。一度に、10,000個のログを参照できます。
- ③ ログテーブルの左上に検索されたログの総数を確認することができます。
- ログデータを時間の順と逆順で照会できます。③ ログテーブルの右上の古い順と新しい順の中から照会方式を選択します。
- 時間範囲を指定した後適用ボタンを選択して照会時間を設定し、🔍 検索ボタンを選択してデータを照会します。
- ③ ログテーブルの右上の🔍 拡大表示アイコンを選択すると、ログとタイムスタンプを全画面で確認できます。
- エージェントオプションが設定されている場合、ログレベルを収集し、各ログの左側に色付きの縦線が次のように表示されます。

2023-12-18 14:31:02.563	INFO	1702877462042	1702877462042	AppLog	io.home.test.logback02starter.base.web.LogbackControllerGreeting
2023-12-18 14:31:02.563	WARN	1702877462043	1702877462043	AppLog	io.home.test.logback02starter.base.web.LogbackControllerError
2023-12-18 14:31:02.586	error	1702877462043	1702877462043	AppServer	io.home.test.logback02starter.base.errors.exception.ApiException: [2204] Process failure. Please try again.
2023-12-18 14:31:02.586	error	1702877462057	1702877462057	AppServer	Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
2023-12-18 14:31:02.586	error	1702877462081	1702877462081	AppServer	Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err
2023-12-18 14:31:02.586	error	1702877462087	1702877462087	AppServer	Servlet.service() for servlet [DispatcherServlet] in context with path [] threw exception [Request processing failed: io.home.test.logback02starter.base.err

① エージェントのオプション設定

```
# whatap.conf
weaving=log4j-2.17
weaving=logback-1.2.8
```

- Javaエージェント2.2.22バージョンからウィーピング設定にlog4j-2.17またはlogback-1.2.8設定時に使用できます。エージェントの再起動が必要です。
- ログレベルは、パーシングされたキーワードの中で level 、 type 基準で判別します。 level 、 type にパーシングされたキーが存在し、パーシングの値がFATAL、CRITICAL、ERROR、WARN、WARNING、INFOを含む場合は、ログレベルの色を表示します。

- ① ログテーブルカラムの端をドラッグしてカラムの幅を修正できます。

ログContentの確認

Content はログメッセージです。ログカラムの1行目には、パースされたキーと値が表示され、2行目にはContent（ログメッセージ）が表示されます。

- ③ ログテーブルの行(ログ)ごとに ▶ 詳細ボタンがあります。 ▶ 詳細ボタンを選択すると④のように、ログの全体Contentを確認できます。
- ログのタグを選択すると、コピー, 検索, 除外検索, 隣接ログを検索できるドロップダウンメニューが表示されます。

フィルター

フィルター適用

① 左側の時間選択バーで時間範囲を指定できます。右側から、フィルターを適用すると、入力した条件に合ったログをフィルタリングします。複数のフィルターを入力できます。フィルターのタグが同じ場合、OR(||)、そうでない場合はAND(&&)が適用されます。

入力欄に文字を直接入力するか、**フィルター**入力欄をクリックしてフィルターを指定します。フィルタータグは **検索キー**、**オペレーター**、**検索値** の順に入力します。 **検索** ボタンを選択すると、フィルターが適用されたデータをログ一覧で確認できます。

① ガイドUI

次のように入力ウィンドウの下に**ガイドUI**を提供します。入力画面またはフィルタータグにマウスカーソルがある場合、**ガイドUI**を閉じる時はESCキーで操作できます。



検索キー、オペレーター、検索値の入力

- **検索キー** 入力時、一般インデックス、予約語インデックス、数字のみ入力できるインデックスを区分して推奨値を提供します。
- **オペレーター** 入力時、一般インデックス検索キーの場合は、**==**、**!=** オプションが下に表示されます。数字のみ入力できるインデックスの場合、**>**、**<**、**<=**、**>=**、**==**、**!=** オプションを提供します。
- **検索値** 入力時の一致検索(**>**、**<**、**<=**、**>=**、**==**)の時は青色で、除外検索(**!=**)の時は赤色でハイライトしま

す。 **検索値** 入力時に大文字と小文字の区分オプションを活用して検索することができます。

- ❗ • フィルタータグが2行以上長くなる場合へ **折りたたみ**アイコンを選択して折りたたむことができます。
- フィルタータグを入力した後、入力画面でShiftとTabキーを同時に使用して、以前のフィルタータグに移動できます。
- フィルタータグを入力した後、入力画面でTABキーを通じて次のフィルタータグに移動できます。

フィルタータグ追加

入力フィールドにテキストを入力し、Enter または Tab キーを押すか、入力欄下のガイド UI で推奨値をクリックまたは矢印キーで選択して追加できます。

フィルタータグの削除

単一のタグを削除するには、**Backspace** キーを押すか、タグ上の **X** アイコンをクリックします。すべてのタグを一度に削除するには、入力フィールド右側の **X** アイコンをクリックします。

フィルターのお気に入り

フィルターお気に入り機能を提供します。入力画面の右側にある **☆ お気に入り**アイコンを選択して、目的のフィルター検索条件をお気に入りに**追加**、**削除**および既存のお気に入りを**選択**できます。**お気に入り**は最大50個まで保存されます。



フィルター適用の例外状況

- 数字のみ入力できるインデックス(.n で終わる 検索キー)を入力したタグで、 検索値 は数字のみ入力できます。
- 重複した 検索キー 、 検索値 は入力できません。
- 検索キー 、 検索値 いずれかのタグが存在しない場合は、検索できません。有効でないタグの場合、灰色と表示します。

構文解析されていないキーワードのフィルター適用

ログからパーシングされていない、つまりインデックスが作成されていないキーワードを含むログを照会できます。この場合、指定範囲内のすべてのログをFull Scanします。そのため、インデックスが生成されたキーに比べて検索速度が多少落ちる可能性があります。定型化されたログデータの場合、**ログパーサの設定**を通じてインデックスキーを活用した検索をお勧めします。

フィルター ⓘ

content == *select* × フィルターを入力してから「Enter」キーを押して追加

content == *select*

1. カテゴリを選択します。カテゴリ指定が必須です。
2. フィルター入力欄に content を選択後、検索キーワードを入力します。
例) content *select*
3. 🔍 検索ボタンをクリックしてログを照会します。ログ全体の一部を先に照会します。1回あたりの検索結果は最大1万件です。
4. スクロールを下げて下段の追加照会ボタンを選択すると、つづけて照会できます。

ログ検索

時間 2024/01/15 14:50 ~ 2024/01/15 15:50

category == AppLog && oid == 397157872 && content == *select*

AppLog (41,513)

全体の41,513件の内、10,500件を照会し、検索された結果は9999+件です。

AppStdErr (2,355)

AppStdOut (56,731)

ログをもっと照会しますか？
ログパーサーを追加することで、追加の照会なしで検索ができます。
ログパーサーの設定 追加照会

- ① 全体ログのうち、サーバー照会範囲当たり1万件ずつ照会します。サーバー照会範囲の場合、基本20万件ですが、ログ全体の量によって比率が変わることがあります。
- パーサ設定の詳細については、[次の文書](#)を参考にしてください。

フィルター修正

フィルターに値を入力し、入力した値をクリックすると、その値を修正できます。

フィルター ⓘ

🔍 oname == ote-account × key == aws.base:index × appender != fileAppender × フィルターを入力してから「Enter」キーを押して追加 ×

☐ 大小文字を区別する

fileAppender

searchValue*	検索値で始まるログを検索します。 ex key!=searchValue*
*searchValue	検索値で終わるログを検索します。 ex key!=*searchValue
searchValue	検索値を含むログを検索します。 ex key!=*searchValue*
*search*Value*	検索値を含むログを検索します。 ex key!=*search*Value*
re:searchValue	正規表現に一致するログを検索します。 ex key!=re:searchValue
**	検索語に該当するすべてのログを検索します。 ex key!=**

key == value 一致 key != value 除外 key == value 無効なタグです。 ⓘ ガイド表示

- ・ フィルター入力欄にテキストを書き換えて修正できます。
- ・ フィルター入力欄の下に表示されるガイドUIから推奨値を選択して修正できます。

検索キー(Search Key)

検索キーは、ログデータ内で希望する特定の値にアクセスするための識別子を意味します。検索キーに該当する実際のデータが検索値です。左側の②領域のタグはカテゴリ別にパーシング(parsing)された検索キーです。タグを選択してフィルターを入力します。オレンジ色タグはカテゴリで、青色タグは検索キーです。

例えば②領域のAppLogとAppStdOutはカテゴリであり、その下のoidなどのタグは、パーシング(parsing)された検索キーです。検索キーはログ>ログ設定のログパーサの設定タブでパーシングロジックを登録し、設定できます。

フィルター入力文法

タグは検索キーと検索値で構成されています。次の例で、検索キーは `exception`、検索値は `UnknownHostException` となります。この例は、収集したログデータのうち、IPアドレスとドメインアドレスが一致せず、サーバーをホストに接続できない場合に発生する例外(`UnknownHostException`)が含まれたログデータを照会します。

exception == UnknownHostException

×

検索キーの種類

検索キーの種類	検索キーフォーマット	意味	検索キーと検索値の例	検索例
文字列キーワード	keyword	ファイル名	- キー: fileName - 値: /data/whatap/logs/yard.log	fileName:/data/whatap/logs/yard.log
数字キーワード	keyword.n	レスポンス時間	- キー: response_time.n - 値: 2945	response_time.n>=2945
予約語キーワード (辞書定義キーワード)	@keyword	TXID	- キー: @txid - 値: 85459614215434144	-

共通文法

文法の種類	説明	例
<code>==searchValue</code>	検索値と一致するログを検索します。	<code>exception==RuntimeExceptionexception</code>
<code>!=searchValue</code>	検索値を除いたログを検索します。	<code>exception!=RuntimeException</code>
<code>*searchValue</code>	検索値で終わるログを検索します。	<code>word==*hello</code>
<code>searchValue*</code>	検索値で始まるログを検索します。	<code>word==hello*</code>
<code>*searchValue*</code>	検索値として中間に含まれるログを検索します。	<code>word==*hello*</code>
<code>*search*Value*</code>	検索値として含まれるログを検索します。	<code>word==*he*llo*</code>

文法の種類	説明	例
re:{正規表現}	正規表現にマッチングされるログを検索します。	caller==re:^i\.w\.a\.w\.s\.v\.r\.
**	検索語に該当するすべてのログを検索します。	

検索キーが数字キーワード(keyword.n)の場合の文法

次の文法は検索キーが、 keyword.n 形式の場合のみに対応します。

- 検索値は数字のみ使えます。
- .n キーワードの値にはPrefixを付けません。 .n 以外のキーワードは、すべてPrefixを付ける必要があります。

例) +>searchValue は有効ではありません。

文法の種類	説明	例
>searchValue	検索値よりも大きな値を含むログを照会します。	response_time.n>3000
>=searchValue	検索値よりも大きいか、同じ値を含むログを照会します。	response_time.n>=3000
==searchValue	検索値と同じ値を含むログを照会します。	response_time.n==3000
!=searchValue	検索値とは異なる値を含むログを照会します。	response_time.n!=3000
<searchValue	検索値より小さい値を含むログを照会します。	response_time.n<3000
<=searchValue	検索値より小さいか、同じ値を含むログを照会します。	response_time.n<=3000

個人情報の照会

ログの個人情報照会権限がある場合**個人情報の表示**  トグルボタンの活性化を通じて非識別化処理した個人情報を表示することができます。

- ❗ ・ ログの個人情報照会権限についての詳細は[次の文書](#)を参照してください。

- ・ ログ個人情報の非識別化機能についての詳細は[次の文書](#)を参照してください。

ログのタグオプション

ログのタグを選択すると、ドロップダウンメニューが表示されます。検索、除外検索、隣接ログ、トランザクション情報オプションが確認できます。

全体90,699件の表示

過去順

最近順

キーワードを入力してください。

🔍

🔗

📄

🗑️

⚙️

▶	oname	タイムスタンプ	ログ
▶	java-demo3	2025-04-17 09:17:13.018	@txid 7759974649447934516 pcode 8 onodeName node-0 oid -278185929 okind 1152715164 oname java-demo3 eException: random exception
▶	java-demo3	2025-04-17 09:17:13.018	demo3 onodeName node-0 agentime 1744849031927 oid -278185929 category AppStdErr okindName dev-okind-0 rtual.web.Simula.exec(Simula.java:79)
▶	java-demo3	2025-04-17 09:17:13.018	demo3 onodeName node-0 agentime 1744849031927 oid -278185929 category AppStdErr okindName dev-okind-0 ion: java.lang.RuntimeException: random exception
▶	java-demo3	2025-04-17 09:17:13.018	demo3 onodeName node-0 agentime 1744849031928 oid -278185929 category AppStdErr okindName dev-okind-0 rtual.web.VExec.doFilter(VExec.java:16)

ログタグ オプション メニュー	説明
コピー	該当するタグのログをコピーします。
検索	フィルターには、該当のタグを含む ('==') 条件で入力されます。
除外検索	フィルターには、該当のタグを除外する ('!=') 条件で入力されます。
隣接ログ	隣接ログの詳細画面では、選択したログのサーバーを対象に、選択したログと隣接する時間帯のログを照会できます。時間を選択して、周辺の時間帯に発生したログを確認することも可能です。基準となるログは青い背景で表示されます。
トランザクション 情報	ログに @txid や @mtid タグが含まれている場合は、トランザクション情報をクリックすると、トランザクション情報の詳細画面で内容を確認できます。

コンテンツハイライト

ログのコンテンツ内で目的のキーワードを簡単に識別できるよう、ハイライト機能を提供します。単一または複数のキーワードでフィルタリングできます。

1. ハイライトしたいキーワードをキーワード入力欄に入力します。
 - 例: `select`
2. 検索したキーワードがログリストのContent内でハイライト表示されます。
 -  **拡大表示** アイコンをクリックすると、**ログ** と **タイムスタンプ** を全画面で確認できます。
 - ハイライトされたキーワード間の移動はショートカットで操作できます。
 - `Enter` : 次の行へ移動
 - `Shift+Enter` : 前の行へ移動
 - `↑ / ↓` アイコン: 行の移動
 - `Cmd + F` (Mac) / `Ctrl + F` (Windows): キーワード検索

複数キーワード条件

複数キーワードでハイライトする場合は、次のように作成します。

入力文字列	説明	結果
<code>a b c</code>	各キーワードをスペースで区切ります。	a, b, c
<code>"Whatap is good."</code>	キーワードにスペースを含める場合は、 <code>"</code> または <code>"</code> で囲みます。	Whatap is good.
<code>"Whatap\\ is good."</code>	<code>"</code> で囲まれたキーワードに、 <code>\</code> を含める場合は、 <code>\\</code> と入力する必要があります。	Whatap\ is good.

ハイライトカラー設定

ハイライトするキーワードおよび色を設定できます。

- デフォルトでは、ログレベルに応じたハイライト（**FATAL**、**ERROR**、**WARN**、**SEVERE**、**CRITICAL**）が適用されています。
- 設定した内容はプロジェクト単位で保存されます。
 -  アイコンをクリックします。
 - ハイライト ウィンドウで、色を設定するキーワードを入力欄に追加してください。
 - 入力欄の左側の色をクリックすると、色を選択できます。

テーブル設定

カラム設定

3 領域の右で  **カラム設定** ボタンを選択します。カラムを追加したり、順序を設定できます。

- カラム追加

タグを選択してテーブルにカラムを追加できます。logカラム選択を解除する場合**ログ表示詳細設定**が確認できません。必ず1つ以上のカラムを選択します。

- カラム順序設定

カラムを追加すると**表示項目**に該当カラムが追加されます。希望のカラムをドラッグしてカラムの順序を変更します。

ログ表示詳細設定

3 領域右で  **ログ表示詳細設定** ボタンを選択します。デフォルトで**content**、**タグ**すべてチェックされており、両方の項目を表示します。**content**と**タグ**どちらかは必ず選択します。

チェックが解除されたアイテムはテーブルに表示されません。次のように**タグ**を解除する場合、テーブルでログの**タグ**は表示されません。

ログテーブルコンテンツタグ非表示

タグ管理一覧にタグを追加すると、追加した順にログのタグがソートされます。タグの順序はドラッグして変更できます。追加したタグを無効にすると、無効にしたタグはログのタグに表示されません。

ⓘ カラムおよびログ表示設定の適用範囲

- カラム設定とログ表示詳細設定は、ログTail、ログ検索、ログトレンドで使用できます。
- 同一プロジェクト内のログTail、ログ検索、ログトレンドは、カラム設定とログ表示詳細設定を共有します。

ログファイルのダウンロード

検索したログの結果を CSV または TXT ファイル形式でダウンロードできます。

1. 画面中央右側の  ダウンロード ボタンをクリックします。
2. CSVダウンロード または Logダウンロード をクリックして、希望の形式のファイルをダウンロードします。

ログの設定

ホーム画面 > プロジェクト選択 > ログ > ログ設定

ログ設定 の上部タブからログモニタリングを設定できます。 エージェント設定の確認、ログモニタリングの有効化設定、ログデータの保持期間および照会パスワードの設定、ログパーサーの登録、クイックインデックスの設定、個人情報のマスキング管理などを行うことができます。

ⓘ 権限

- **ログモニタリング機能の活性化** を使用するには、**プロジェクト修正** 権限が必要です。
- **ログエディター** 権限があれば、**ログモニタリング機能の活性化** のほか、**ログ設定** メニューを編集できます。

ログモニタリングを開始する

ログ設定

ログモニタリングの開始

ログプライマリパーサ設定

ログセカンダリーパーサの設定

高速インデックス設定

ログ長期保存統計

1

エージェントの設定とログモニタリング活性化

ログTail、ログトレンド、ログ検索機能で、アプリケーションに点在するログをまとめて確認することができます。

ログモニタリングの設定方法

[インストール型DB](#)[AWS RDS](#)

- #### XOSエージェントのインストール

LinuxファイルシステムにインストールされたDBのログは、WaタブのXOSエージェントで収集します。
下記のガイドリンクからインストールを進めてください。

[エージェントインストールガイド](#)
- #### xos.confファイルにオプション適用

xos.confファイルに次のオプションを適用してください。

```
tail -f /var/log/audit/audit.log
tail -f /var/log/audit/audit.log
tail -f /var/log/audit/audit.log
```

コピー
- #### ログモニタリング機能の活性化

設定完了後、ログ監視を有効にするトグルボタンをクリックしてください。

☒

2

ログモニタリングデータの設定

初期化

保存

ログ使用量

0ライン

ログ照会パスワード

☐ ログ照会の際にパスワードを入力するするには、オンにしてください。

データ保存期間

1日

カテゴリ別のデータ保存期間

カテゴリ	データ保存日	日間ログ数 ①	昨日のログ数 ①	週間ログ数 ①	月間ログ数 ①	予想ログ数 ①
WhatapEvent.whatap	1日	▼	4	0	4	4
DB.whatap	1日	▼	0	0	0	0

上段でログモニタリングの開始タブを選択します。  ガイド表示アイコンを選択すると、関連の案内画面に移動します。

エージェント設定とログモニタリングの有効化

- ① 領域でエージェント設定を確認し、ログモニタリングの有効化または無効化を設定できます。

エージェントの設定確認

ログモニタリングを開始するために、エージェントのバージョンと設定情報を確認してください。 エージェント設定

に関する詳細は[次の文書](#)を参照してください。

ログモニタリングの有効化

ログモニタリングを有効化または無効化できます。**2. ログモニタリング機能の活性化**でログモニタリングの有効／無効を設定してください。

エージェントの設定とログモニタリング活性化

ログTail、ログトレンド、ログ検索機能で、アプリケーションに点在するログをまとめて確認することができます。

ログモニタリングの設定方法

[インストール型DB](#) AWS RDS

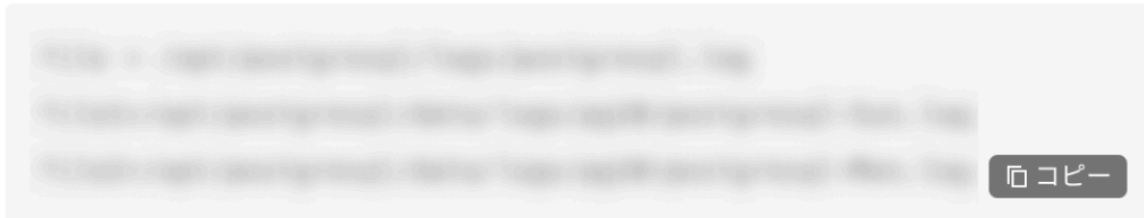
1. XOSエージェントのインストール

LinuxファイルシステムにインストールされたDBのログは、WaタブのXOSエージェントで収集します。
下記のガイドリンクからインストールを進めてください。

[エージェントインストールガイド >](#)

2. xos.confファイルにオプション適用

xos.confファイルに次のオプションを適用してください。



3. ログモニタリング機能の活性化

設定完了後、ログ監視を有効にするトグルボタンをクリックしてください。



-  ログモニタリングの有効化：トグルボタンをオンにするとログモニタリングが有効になり、有効化した日から

15日間は無料でお試しいただけます。

- ☐ ログモニタリングの無効化：トグルボタンをオフにするとログモニタリングが無効になり、以降ログは保存されません。

ログモニタリングのデータ設定

2 領域でログ使用量が確認できます。またログ保存期間とログ照会パスワード設定を変更できます。

ログ参照パスワード

セキュリティを強化するため、**ログ照会パスワード**を設定してください。ログ検索パスワードの指定はオプションです。ログ検索パスワードを使用中の場合は、ログ画面に入るときは必ずパスワードを入力します。

❗ パスワードを忘れた場合

ログ編集権限を持つユーザーは、**ログ設定** で新しいパスワードに変更できます。

ログ保存期間

共通に適用する基本(default) データ保持期間です。

- 最小1日から最大40日まで入力できます。未指定の場合、既定値は1日です。ログ保存期間の選択オプションの他に、ユーザーが希望する期間を入力できます。
- **ログ使用量**一覧で別に設定しない場合、このログデータの保存期間がデフォルトで適用されます。**ログ使用量** リストでカテゴリごとのデータ保持期間を設定し、**初期化** ボタンをクリックすると、デフォルトのデータ保持期間に初期化されます。

ログ使用量

ログ使用量一覧でカテゴリ別のログデータの保存期間を指定できます。ログ数は、当該期間中に蓄積されたログの件数を意味します。例えば、**日間ログ数**は、1日にたまったログ件数、**予想ログ数**は、データ保持日に今日のログ数を掛けたログ件数を意味します。

ログデータの保存期間を次のように指定できます。期間指定により、古いデータを削除してスペースを確保できます。

- **トライアルプロジェクト**

データ保存期間として1日、2日、3日を選択できます。

- **有料プロジェクト**

データの保存期間として1日、2日、3日、4日、5日、6日、7日、10日、30日、40日を選択できます。

- **保存量基準課金**

データの保存期間によってコストが異なります。

例、1日平均200万ログ件数が蓄積され、データ保存期間を3日に指定した場合、平均600万ログ件数が収集サーバーに保持され、課金の対象となります。

ログプライマリパーサ設定

ログ設定 上部の **ログプライマリパーサ設定** タブを選択して、収集されたログに対するパーサーを登録できます。

ログプライマリパーサは**GROK**と**JSON**パーサーを提供し、タイプ別ログ発生数の集計や特定ログの検索に必須のパーサーです。パターン条件に一致する検索キーと検索値を抽出し、パーシングされたキーはログのタイプ分類と検索インデックスに活用します。

- **GROK**：基本的には、正規表現ベースのパーシングに対応します。予約キーワードベースのパーシングを提供します。
- **JSON**：ログからJSONで出力された部分に対して一括パーシング機能を提供します。

✔ **パーシングロジックが未登録時に検索可能なkey**

category , oid , oname , okind , okindName , @txid , @login , httphost

パーサーで登録できない予約語

timestamp , message , pcode , category , content , logContent

- 次の予約語の場合、パーサーを登録してもインデックスは作成されません。

❗ ログパーサーに関する詳細は[こちらの文書](#)をご覧ください。

設定項目

設定値	説明	その他
カテゴリ	パターンを適用する対象のカテゴリです。	required
ログ検出条件	フィルターとして使用する検索キーと検索値を入力します。 - ログ検出条件に一致するログデータにのみパターンが適用されます。検出条件を入力しない場合、すべてのログにパターンが適用されます。	optional
パターン	ログをパースするためのパターンです。 - 作成したパターンに従ってログをパースし、インデックスを生成します。 GROKおよび正規表現の構文をサポートしています。	required

パーサー一覧

登録済みのパーサーを確認でき、追加や編集が可能です。適用順にパーサーが評価され、最初に一致したパーサーのみが適用されます。

- 右上の*****追加する**ボタンをクリックすると、**パーサ 追加画面**が表示されます。
- **優先順位**カラムの📌アイコンをドラッグしてパーサの設定手順を変更できます。
-  **活性化**トグルでパーサの活性化有無を指定できます。
-  **修正**及び  **削除**アイコンで登録したパーサを修正または削除できます。

パーサー登録

以下はパーサーを登録する共通の手順です。

1. **ログプライマリパーサ設定** で **+ 追加する** ボタンをクリックしてください。
2. **パーサ 追加画面** でパーサを選択してください。

❗ パーサーおよびパターンの登録方法

- [GROK パーサおよびパターン登録](#)
- [JSON パーサおよびパターン登録](#)

3. パーサーを選択すると、**パターン登録**ボタンが有効になります。**パターン登録**ボタンをクリックした後、**パーサーシミュレーション**ウィンドウでパターンとログを入力します。

a. **パターン**を入力してください。

b. **ログのカテゴリ**と**シミュレーション用のパースログ**を入力してください。**ログ検索**ボタンをクリックした後、検索結果の一覧から目的のログの右側にある**選択**ボタンをクリックすると、ログが自動的に入力されます。

❗ **ログ検索**を使用して、収集されたログの中からサンプルログを選択できます。選択したログはシミュレーション入力欄に自動的に反映され、コピーや貼り付けなしですぐにテストできます。

c. 登録するパターンが正しいかどうかを確認するには、**シミュレーション**ボタンをクリックして、シミュレーションとパターンのパフォーマンスを測定してください。

❗ シミュレーションと性能測定についての詳細は、[次の文書](#)を参照してください。

4. 正常にパースされるかどうかシミュレーション結果を確認した後、**パターン適用**ボタンをクリックしてください。

5. **追加**ボタンをクリックすると、パーサーが追加されます。

❗ 個人情報の非識別化

パーサー登録時に検索キー名を **検索キー.p** に指定すると非識別化の対象になります（例：**myname.p** **）。WhaTapは**個人情報の非識別化**タブで非識別化を管理する方式を勧めます。詳細は**個人情報の非識別化**を参照してください。

GROKパーサパターン登録

基本文法は、`%{SYNTAX:SEMANTIC}` です。GROKパーサの詳細は、[次の文書](#)を参照してください。パターン登録とシミュレーションが必須です。

- SYNTAX

GROK定義パターンです。

- SEMANTIC

パーシングされたデータに割り当てるキーです。

❗ SEMANTICでは、予約語ではなく、組み合わせでの使用を推奨します。

JSONフォーマットのパーサパターン登録

ログ全体または一部がJSON形式で出力される場合、JSONフォーマットのパーサを使用して、JSONに出力された部分をパーシングできます。ログのうちJSON形式として出力された部分を検出するために、**Prefix**、**Postfix**オプションを組み合わせでログのどの部分をJSONとして認識してパーシングするかを指定します。JSONパーサの詳細は、[次の文書](#)を参照してください。シミュレーションが必須です。

オプション	説明
Prefix	JSON文字列の先頭の前にある文字列を指定します。未指定時、ログ出力文の先頭からJSON文字列として識別します。
Postfix	JSON文字列の最後の部分の文字列を指定します。未指定時、ログ出力文の最後までをJSON文字列として識別します。
Ignore	JSON出力部分のうちキー抽出を除外するフィールドを指定します。

- 登録例

Log

```
[2022-10-25 10:15:34:145]...(改行)
Request: {"key1": "value1", "key2": "value2", ...}(改行)
Response: {"key3": "value3", "key4": "value4", ...}
```

例のように流入されるログがRequest JSON、Response JSONをすべてパーシングする場合、次の2つのパターンを登録します。

- Requestパーシング用のパターン

"Request : "と"Response"間の文字列 {"key1": "value1", "key2": "value2", ...} 対象

- Responseパーシング用のパターン

"Response : "からログの最後まで文字列 {"key3": "value3", "key4": "value4", ...} 対象

• JSONカスタムパターンの登録

ログの一部がJSON形式で出力される場合、JSONで出力された部分をカスタムパーサを使用してパーシングできます。パターンを次のように入力します。

```
io.whatap.logsink.parser.JsonFormatParser{}
```

ログのうちJSON形式として出力された部分を検出するために、**Prefix**、**Postfix**オプションを組み合わせでログのどの部分をJSONとして認識してパーシングするかを指定します。

JsonFormatParser{} の {} にオプションを指定します。

- 登録例

Log

```
[2022-10-25 10:15:34:145]...(改行)
Request: {"key1": "value1", "key2": "value2", ...}(改行)
Response: {"key3": "value3", "key4": "value4", ...}
```

例のように流入されるログがRequest JSON、Response JSONをすべてパーシングする場合、次の2つのパターンを登録します。

- Requestパーシング用のパターン

"Request : "と"Response"間の文字列 {"key1": "value1", "key2": "value2", ...} 対象

```
io.whatap.logsink.parser.JsonFormatParser {prefix:"Request :",postfix:"Response"}
```

- Responseパースング用のパターン

"Response："からログの最後までまでの文字列 {"key3":"value3","key4":"value4",...} 対象

```
io.whatap.logsink.parser.JsonFormatParser {prefix:"Response:"}
```

パーサーシミュレーションと性能測定

パーサーシミュレーション後、パターンを登録できます。パフォーマンス測定は、シミュレーション実行対象文字列に対してパーサーの反復パースング所要時間を測定します。

1. パターンとログを入力します。ログ検索ボタンをクリックした後、検索結果の一覧から目的のログの右側にある選択ボタンをクリックすると、ログが自動的に入力されます。
2. シミュレーションボタンをクリックして、登録しようとするパターンでパースングに成功することを確認します。
3. シミュレーションの成功時シミュレーション結果と性能測定の結果を照会できます。
4. シミュレーション後、パターン適用ボタンをクリックすると、選択したパーサーに入力したパターンが適用されます。

パースングの成功

パースング ロジックを登録し、キー(key)が生成されたら、ログを照会する際に該当キーでパースングされた値が追加されます。次のログTailメニュー例のように、パースングされたキーと値が追加されます。

	Timestamp	ログ
▶	2022-08-17 14:28:00.612	oname dev949400-8093 onodeName node-1 oid 413390913 category AppStdOut okindName dev-okind-1 load 52

パースングされたキーはログTail、ログ検索、ログトレンドで確認できます。

ログセカンダリーパーサ設定

ログ設定メニューの上部にあるログセカンダリーパーサの設定タブを選択してログパーサを登録できます。

ログのセカンダリーパーサは、プライマリーパーサ(GROKまたはJSON)で抽出した値を加工して統計データや新しいフィールドを生成します。HTTPステータスコードベースの統計パーサとともに、キーワードフィルタ、フィールド加工、応答時間分類、GeoIP変換など、さまざまなプロセッサを提供します。

❗ ログのセカンダリーパーサは、1次でパースされた結果に対して、追加での2次パース機能を提供します。セカンダリーパーサを使用するには、**プライマリーパーサ**を登録する必要があります。

パーサー一覧

ログ設定

ログモニタリングの開始 ログプライマリーパーサ設定 **ログセカンダリーパーサの設定** 高速インデックス設定 ログ長期保存統計 1時間の統計ウィジェットのデータ設定のログ

ログセカンダリーパーサの設定 Grok、またはJSONパーサがすでに解析された場合に使用できるパーサです。

+ 追加する

保存

優先順位	パーサ	カテゴリ	ログ検出条件	パターン	活性化	
0	4xx、5xx状態コードパーサ	AppStdErr	すべてのログ		☒	
1	状態コード成功率パーサ	AppLog	すべてのログ		☒	

登録したパーサを確認でき、追加および編集が可能です。

パーサの追加、順序変更、活性化トグル、修正および削除などの操作方法は[プライマリーパーサー一覧](#)と同一です。

パーサ登録手順

次に、パーサ登録の共通の手順を説明します。

× パーサ 追加

パーサ*

WhaTap Monitoringには、サービスの信頼性に影響を与える可能性があるパーサーを無効にできる権限があります。

パーサを選択してください。



除外する状態コード

パーサを選択してください。

カテゴリ*

同じカテゴリに複数のパーサーが登録されている場合、最初に一致するパーサーのみが適用されます。

カテゴリを選択してください。

ログ検出条件

フィルタを入力しなかった場合、または検索キー／値を1つだけ入力した場合、そのパターンはすべてのログに適用される。

検索キー

検索値

追加

1. **+追加する**ボタンを選択すると**パーサ 追加**画面が表示されます。
2. **パーサ**選択画面からパーサを選択します。各パーサの設定項目の詳細については、次の文書を参照してください。
 - [4xx、5xx状態コードパーサ 設定項目](#)
 - [状態コード成功率パーサ 設定項目](#)
3. **除外する状態コード**を入力します。
4. **カテゴリ**選択画面からカテゴリを選択するか、直接入力します。
5. **ログ検出条件**を選択するか、直接入力します。
6. **追加**ボタンを選択してパーサを登録します。

4xx、5xxステータスコードパーサ設定項目

4xx、5xx状態コードパーサ はプライマリーパーサでパーシングされたstatus値を基に4xx、5xx件数データを生成します。特定のステータスコードを除外する場合は、除外するステータスコードを入力してください。

設定項目

設定値	説明	その他
カテゴリ	4xx、5xxの件数データを生成するカテゴリです。	required
ログ検出条件	フィルターとして適用する検索キー、検索値を入力します。ログ検出条件を満たすログデータに対して、4xx、5xxのデータを生成します。ログ検出条件を入力しない場合は、すべてのログのデータを生成します。	optional
除外するステータスコード	統計データの作成時に除外するステータスコードです。入力がない場合は、4xx・5xxに該当するすべてのエラーステータスコードを対象に4xx、5xxの件数データを生成します。	optional

statusパーサの登録例

ログプライマリパーサ設定 収集したログに対してパーサを登録することができます。パーサは、順番通り適用され、最初にマッチしたパーサのみが適用されます。 + 追加する 保存						
優先順位	パーサ	カテゴリ	ログ検出条件	パターン	活性化	
0					☐	 
1					☒	 
2	GROK	AppLog	すべてのログ	%{NUMBER:status}	☒	 

流入されるログが `{"msg":"message","status":404}` で、例のようにGROKパーサでstatusをパーシングする場合、`status: 404` のようにパーシングされます。statusが正しくパーシングされることを確認したら、4xx、5xxのステータスコードのパーサから除外するステータスコードを登録します。

データ参照

パーサをすべて登録すると、統合Flexボードに移動してログ 4XX、5XX件数ウィジェットを作成します。

ウィジェットテンプレート

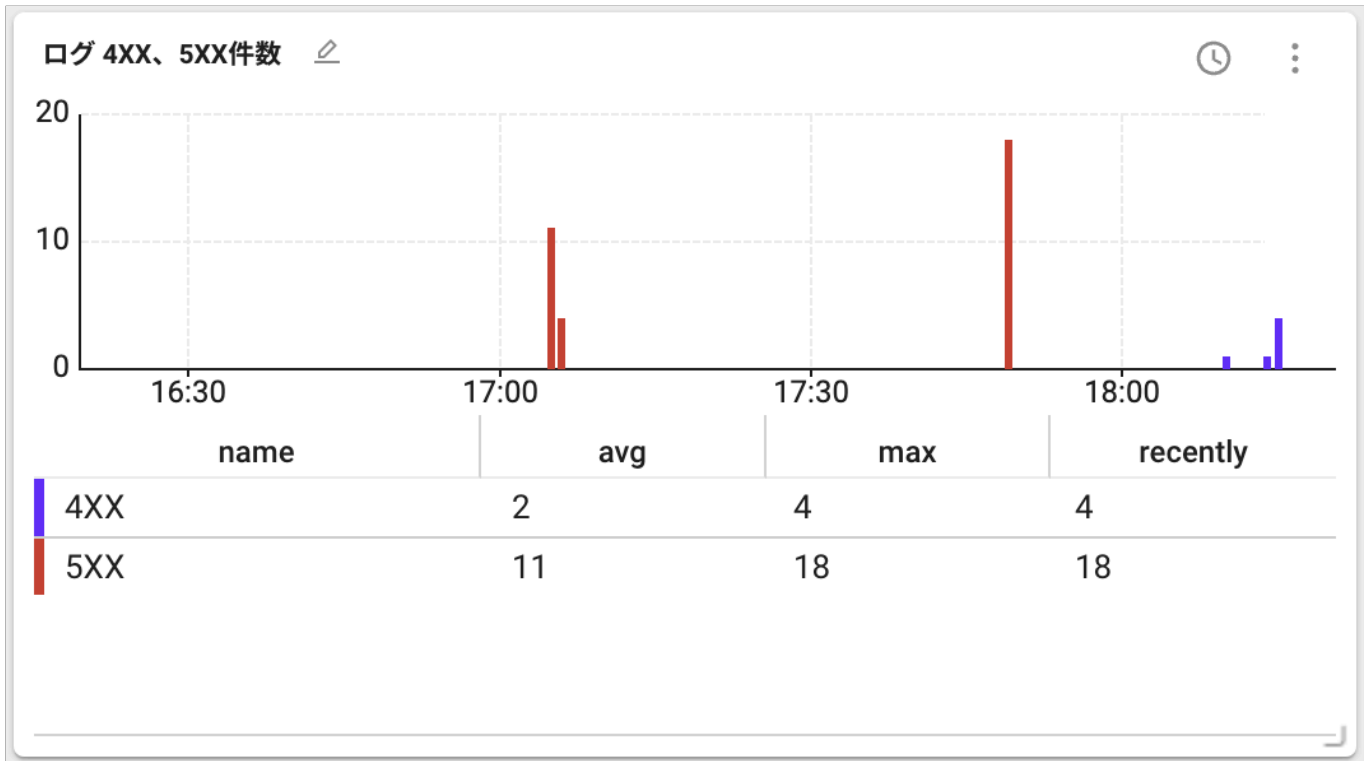
⇒ すべてのメトリックス

4xx

🔍

📊 ログ 4XX、5XX件数

ウィジェットを作成すると、次のようにデータを確認できます。



- **avg**：照会期間のデータで平均値です。
- **max**：照会期間のデータで最大値です。
- **recently**：照会期間のデータで最後の値です。

ステータスコード成功率パーサ設定項目

状態コード成功率パーサ はプライマリーパーサでパースされたstatus値を基にHTTPリクエスト成功率データを生成します。特定のステータスコードを除外する場合は、除外するステータスコードを入力してください。statusパー

シングに関する内容は、[4xx、5xxステータスコードパーサ設定項目](#)を参照してください。

設定項目

設定値	説明	その他
カテゴリ	要請成功率データを生成するカテゴリです。	required
ログ検出条件	フィルターとして適用する検索キー、検索値を入力します。ログの検出条件を満たすログデータのみで要請成功率データを生成します。ログ検出条件を入力しない場合は、すべてのログのデータを生成します。	optional
除外するステータスコード	要請成功率データを生成する際に除外するステータスコードです。入力がない場合は、2xx・3xxに該当するすべての成功ステータスコードを対象に要請成功率のデータを生成します。	optional

データ参照

パーサをすべて登録したら、**統合Flexボード**に移動し、ログ要請の成功率ウィジェットを作成します。

ウィジェットテンプレート

⇌ すべてのメトリックス

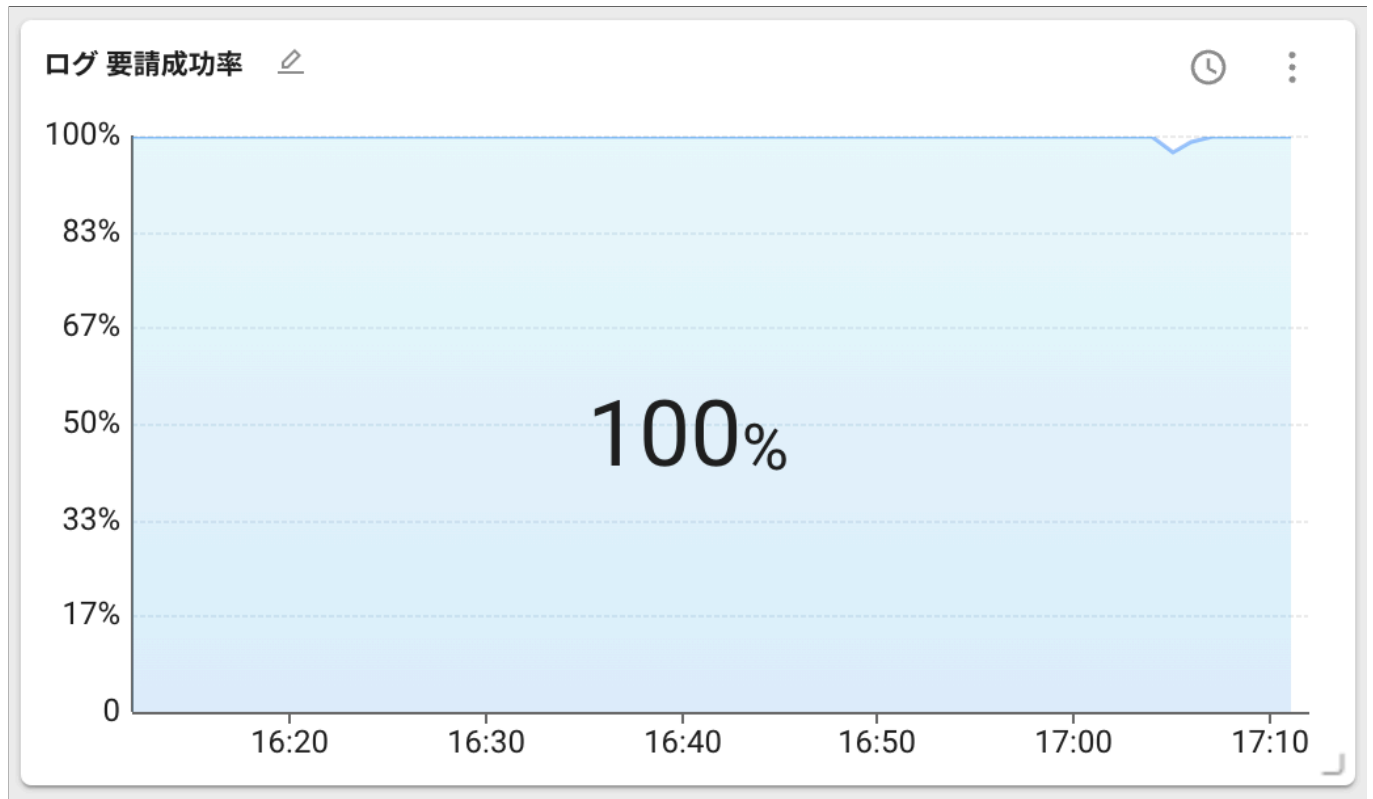
成功率

🔍

📊

ログ 要請成功率

ウィジェットを作成すると、次のようにデータを確認できます。



チャート上のデータは、照会期間に関する統計を示します。統計方法としては最近値、最大値、平均値などを選択できます。最近使った値がデフォルトで選択されています。

セカンダリーパーサプロセッサ

ログのセカンダリーパーサは、4xx、5xx状態コードパーサ と状態コード成功率パーサ のほかにも、次のようなフィールド加工プロセッサを提供します。

ログのセカンダリーパーサプロセッサ一覧

プロセッサ	説明
すべてのキーワード含有フィルタ	指定したすべてのキーワードが含まれるログにフラグフィールドを追加します。
一部キーワード含有フィルタ	指定したキーワードのうち1つでも含まれるログにフラグフィールドを追加します。

プロセッサ	説明
フィールド名変更	既存のフィールド名を別の名前に変更します。
ステータスコードグループ分類	HTTPステータスコードを2xx/3xx/4xx/5xxグループに自動分類します。
URL静的/動的分類	URLに拡張子(.)があれば静的コンテンツ、なければ動的サブリットに分類します。
URLパラメータ除去	URLから?以降のクエリパラメータを除去した値を保存します。
応答時間分類	応答時間(ms)をfast/normal/slow/critical区間に自動分類します。
JSONフィールド抽出	JSON文字列フィールドから特定のキーを抽出して新しいフィールドとして保存します。
GeoIP変換	IPアドレスを国/都市情報に変換します。
直接入力	FQNクラス名でプロセッサを直接登録します。

すべてのキーワード含有フィルタ

指定したすべてのキーワードが含まれるログにフラグフィールドを追加します。

設定値	説明	デフォルト値
入力フィールド	キーワードを検索するフィールド名	content
保存フィールド	結果を保存するフィールド名	has_str_all
keyword	検索するキーワード一覧(カンマで区切って入力)	required

活用例: 入力フィールドを content 、キーワードを error, timeout, database に設定すると、3つのキーワードがすべて含まれるログにのみ has_str_all フィールドが追加されます。

一部キーワード含有フィルタ

指定したキーワードのうち1つでも含まれるログにフラグフィールドを追加します。

設定値	説明	デフォルト値
入力フィールド	キーワードを検索するフィールド名	content
保存フィールド	結果を保存するフィールド名	has_str_any
keyword	検索するキーワード一覧(カンマで区切って入力)	required

活用例: 入力フィールドを `content`、キーワードを `reboot, shutdown, restart` に設定すると、キーワードのうち1つでも含まれるログに `has_str_any` フィールドが追加されます。

フィールド名変更

既存のフィールド名を別の名前に変更します。対象のフィールド名がすでに存在する場合は上書きしません。

設定値	説明	デフォルト値
元のフィールド	変更する既存のフィールド名	required
新しいフィールド名	変更後のフィールド名	required

ステータスコードグループ分類

HTTPステータスコードを `status_2xx`、`status_3xx`、`status_4xx`、`status_5xx` グループに自動分類します。

設定値	説明	デフォルト値
入力フィールド	ステータスコードが入ったフィールド名	status
保存フィールド	グループ分類結果を保存するフィールド名	status_nxx

活用例: `status` 値が `404` の場合、`status_nxx` フィールドに `status_4xx` が保存されます。

URL静的/動的分類

URLに拡張子(.)が含まれていれば静的コンテンツ、含まれていなければ動的サブレットに分類します。

設定値	説明	デフォルト値
入力フィールド	URLが入ったフィールド名	url
保存フィールド	分類結果を保存するフィールド名	url_type

URLパラメータ除去

URLから ? 以降のクエリパラメータを除去したパスのみを保存します。

設定値	説明	デフォルト値
入力フィールド	URLが入ったフィールド名	url
保存フィールド	パラメータ除去結果を保存するフィールド名	url_without_params

活用例: /api/v1/users?page=1&size=20 → url_without_params フィールドに /api/v1/users が保存されます。

応答時間分類

応答時間(ms)を区間別に自動分類します。区間のしきい値を直接設定できます。

設定値	説明	デフォルト値
入力フィールド	応答時間(ms)が入ったフィールド名	elapsed
保存フィールド	分類結果を保存するフィールド名	response_class
Fast上限(ms)	fast区間の上限値	100
Normal上限(ms)	normal区間の上限値	500

設定値	説明	デフォルト値
Slow上限(ms)	slow区間の上限値	2000

デフォルト値基準の分類結果は次のとおりです。

応答時間	分類
0 ~ 100ms	fast
101 ~ 500ms	normal
501 ~ 2000ms	slow
2001ms以上	critical

活用例: ログ探索で `response_class: critical` でフィルタリングすると、応答時間が2秒を超えるログのみを照会できます。

JSONフィールド抽出

プライマリーパーサで抽出したフィールドにJSON文字列が含まれる場合、該当のJSONから特定のキーを抽出して新しいフィールドとして保存します。

設定値	説明	デフォルト値
入力フィールド	JSON文字列が入ったフィールド名	payload
抽出するキー	抽出するJSONキー一覧(カンマで区切って入力)	required
保存プレフィックス	抽出されたキー名の前に付けるプレフィックス(例: <code>json_</code>)	optional

- ① ログ全体がJSONの場合、1次JSONパーサが自動的にパーシングします。このプロセッサは、**GROK 1次パーサ**使用時にテキストログ内に含まれるJSON部分を追加でパーシングする際に便利です。
- パフォーマンス保護のため、JSONフィールドサイズが10KBを超えるとパーシングをスキップします。

活用例

1次GROKパーシング結果

```
payload = {"user": "alice", "action": "purchase", "product": "laptop", "amount": 45000}
```

入力フィールド `payload`、抽出するキー `user, action`、保存プレフィックス `pay_` を設定すると、次のように抽出されます。

JSONフィールド抽出結果

```
pay_user = alice
pay_action = purchase
```

GeoIP変換

IPアドレスを国および都市情報に変換します。IP2Locationデータベースを活用して、アクセスログの地域別分析が可能です。

設定値	説明	デフォルト値
入力フィールド	IPアドレスが入ったフィールド名	<code>client_ip</code>
保存プレフィックス	結果フィールド名のプレフィックス	<code>geo</code>

結果として `{プレフィックス}_country`、`{プレフィックス}_city` の2つのフィールドが生成されます。

活用例: 入力フィールド `client_ip` の値が `223.130.195.200` で、保存プレフィックスが `geo` の場合:

GeoIP変換結果

```
geo_country = KR
geo_city = Seongnam
```

直接入力

一覧にないプロセッサをFQN(Fully Qualified Name)クラス名で直接登録できます。

直接入力形式

```
io.whatap.logsink.processor.ClassName{"param1":"value1","param2":"value2"}
```

- ❗ クラス名とパラメータ形式が正確である必要があります。誤った入力の場合、プロセッサが動作しないことがあります。

セカンダリーパーサのチェイニング

同一カテゴリに複数のセカンダリーパーサを登録してチェイニングできます。各プロセッサは登録順に実行されます。

元のログ

```
2026-04-23T10:30:00.123+0900|INFO|api-01|192.168.1.10|/api/v1/orders?detail=true|200|1234|{"user":"alice","action":"purchase"}|Request processed
```

順序	パーサ	入力 → 出力
1次	GROK	timestamp, level, host, client_ip, url, status, elapsed, payload, content
2次-1	JSONフィールド抽出	payload → user, action
2次-2	応答時間分類	elapsed → response_class (slow)
2次-3	ステータスコードグループ分類	status → status_nxx (status_2xx)
2次-4	URLパラメータ除去	url → url_without_params (/api/v1/orders)
2次-5	GeoIP変換	client_ip → geo_country (KR), geo_city (Seoul)

- ❗ セカンダリーパーサの**カテゴリ**は、プライマリーパーサと同一に設定する必要があります。異なるカテゴリに設定すると、該当のログにセカンダリーパーサが適用されません。

高速インデックス設定

ログ設定メニューの上段から高速インデックス設定タブを選択します。大量のログを収集すると、ログ検索性能が低下する場合があります。よく使う検索条件は、インデックス(index)で生成すると、ログ検索がより速くなります。設定項目は次のとおりです。

設定値	必須の有無	説明
カテゴリ	必須	高速インデックスを設定するカテゴリ
検索キー	必須	高速インデックスを設定する検索キー
大文字と小文字を区別しない	オプション	大文字と小文字の区別するかどうか
ルール	必須	* を少なくとも 1 つ以上を含める必要あり
活性化有無	必須	活性化または非活性化の設定(既定値 true)

ログ設定のインポート・エクスポート

パーサ設定、高速インデックス設定、フィルター設定の内容をJSONファイル形式で保存し、他のプロジェクトからJSONファイルをインポートして適用できます。プロジェクトごとに何度も繰り返し設定を作成する手間を減らします。

- 1つのプロジェクトにパーサ設定、高速インデックス設定、フィルター設定を追加します。
- 各設定タブで画面右上にJSON  ボタンを選択します。
- JSONエクスポート画面が表示されると、画面右上にある抽出ボタンを選択します。
- JSON設定ファイルがユーザーPCに保存されます。
- 他のプロジェクトに移動してから、ログ > ログ設定メニューに移動してください。
- まず、JSON設定ファイルをエクスポートした設定タブを選択し、 ボタンを選択します。
- ファイル選択画面が表示されたら、ユーザーのPCに保存したJSON設定ファイルを選択します。

8. JSONインポート画面が表示されたら、設定内容を確認し、**リストに追加する**または**上書きボタン**を選択します。
9. 画面右上の**保存ボタン**を選択します。

❗ JSON設定ファイルをインポートして、**保存ボタン**を選択しないと、インポートされた設定は保存されません。

ログ長期保存統計の設定

ログ設定メニューの上段から**ログ長期保存統計**タブを選択します。ログデータは、非常に大容量のため、長期間保存することは難しいです。したがって、ログ統計データの設定機能を使用して**特定の条件を満たすログデータが5分ごとに何件収集されたかに関する情報を保存**できます。長期間にあたって、実際のログデータが削除されても、条件を満たすログがどれだけ収集されているかの推移は確認できます。

ログ長期保存統計の追加

×

ログ長期保存統計 追加

統計キー*

統計キーを入力してください。

データ保存期間
(ディスク使用量)

15日 (約 3 MB) ▾

カテゴリ*

カテゴリを選択してください。

ログ検出条件*

まず、カテゴリを選択してください。 ▾

まず、カテゴリを選択してください。

除外 ☐

大小文字を区別する ☒

+ 追加する

追加

ログ長期保存統計タブで+ 追加するボタンを選択するとログ長期保存統計 追加画面が表示されます。+ 追加するボタンでルールを追加したり、作成したルールを**-**アイコンで削除できます。

設定項目

フィールド	説明
カテゴリ	ルールが適用されるカテゴリです。
統計キー(key)	ルールを満たすログが発生した場合、保存するキーとして同じキーを重複で設定できません。
ログ検出条件	ログ統計データを生成する条件です。この条件を満たすログがどのくらい収集されたかで統計データを生成します。
除外	除外をチェックすると、入力した条件を満たさない値で統計データを作成します。
大小文字を区別する	入力したログ検出条件の値に対して大文字と小文字の区別有無を指定します。
活性化有無	活性化または非活性化の設定(既定値 true)

例

次のように設定を追加した場合、収集されたログのうち、statusが 200 , 300 のログを対象にTotalCountのキーで統計データを作成します。

ログ長期保存統計 指定した条件を満たすログがどれ程収集されたかの統計データが生成できます。 					
カテゴリ	統計キー	ログ検出条件	データ保存期間 (ディスク使用量)	活性化	
AppLog	TotalCount	status 200, 300	15日 (約 3 MB)	☐	 
AppLog	TotalCount	status 200, 300	15日 (約 3 MB)	☐	 
AppLog	TotalCount	status 200, 300	15日 (約 3 MB)	☒	 
AppLog	TotalCount	status 200, 300	15日 (約 3 MB)	☒	 
AppLog	TotalCount	status 200, 300	15日 (約 3 MB)	☐	 

データ参照

1. 統合Flexボードのウィジェットテンプレートでログ長期保存統計を検索し、ウィジェットを作成します。

ウィジェットテンプレート

⇒ すべてのメトリックス

ログ長期保存統計



📈 ログ長期保存統計

2. データを照会するカテゴリと統計キーを指定して適用ボタンを選択します。

ログ長期保存統計

カテゴリ変更

AppLog

統計キー

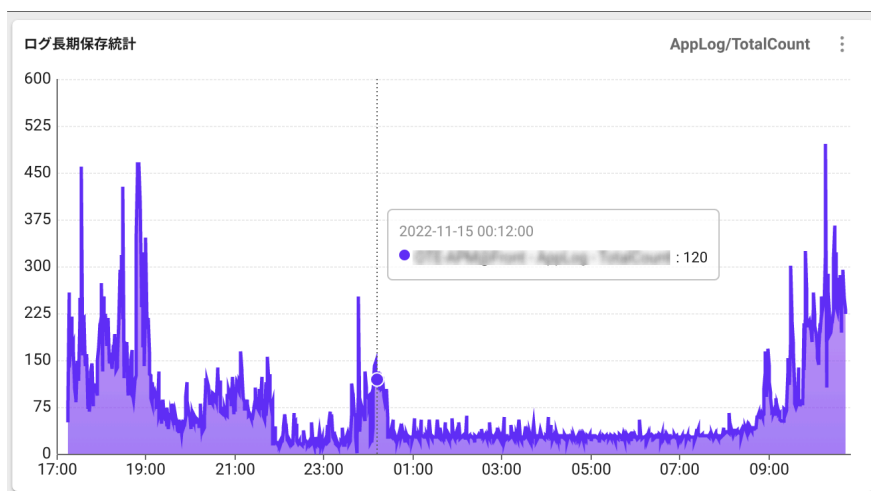
TotalCount

キャンセル 適用

カテゴリ、キーを選択してください。

該当ウィジェットに表示するデータ収集のためにはログ長期保存統計 設定が必要です。
当該設定をしていない場合は、
[ガイド](#)を参考してログ長期保存統計 設定をしてください。
[ログ設定](#)

3. 追加した設定でログ長期保存統計データを次のように確認できます。



個人情報の非識別化

個人情報の非識別化ではログデータ内の個人情報をマスキング処理したり、安全な値に置換できます。

- 暗号化の保存：指定された機密な情報は暗号化され、データベースに保存されます。
- マスキング処理：ログ画面に表示される時、機密な情報は基本的にマスキング(*******)処理されて表示されます。
- 置換処理：機密な情報をユーザーが指定した安全な値に変更します。

個人情報の分類

一般的に個人情報として分類する項目は次のとおりです。一般の個人情報以外にも機密な情報として処理すべき項目があれば**個人情報の非識別化**機能を通じて、ログ内で必要な項目をマスキング処理または置換処理してください。

- 個人識別番号
- パスポート番号
- 運転免許証番号
- 外国人登録番号
- クレジットカード番号
- 銀行口座番号

- 生体情報

要求権限

非識別化された個人情報はマスキング処理されて表示されます。ログの個人情報照会権限が付与されたユーザーのみ個人情報の表示トグルを通じて元のデータを確認できます。非識別化の対象検索キーの追加及び修正は、ログ編集権限が必要です。

❗ ログの個人情報照会権限及びログ編集権限の詳細は次の文書を確認してください。

個人情報の非識別化設定

新規のログデータにマスキング処理を適用して機密な情報を保護できます。

The screenshot displays the 'Log Settings' (ログ設定) interface. The left sidebar contains navigation links: 'Log Monitoring Start' (ログモニタリングの開始), 'Log Primary Parser Settings' (ログプライマリパーサ設定), and 'Log Secondary Parser Settings' (ログセカンダリーパーサの設定). The main content area is titled 'Personal Information Non-identification' (個人情報の非識別化) and includes a description: 'Personal information contained in the search key is added, and the corresponding data is non-identified.' (個人情報が含まれた検索キーを追加すると、当該データを非識別化します。)

Below the description is a table with columns 'Activation' (活性化) and 'Category' (カテゴリ). The 'AppLog' entry is shown with a green toggle switch and a list of search keys: @txid, agenttime, and @mt. Below the table is a 'Log Personal Information Non-identification Check' (ログ個人情報の非識別化チェック) section, which includes a time range selector (2025/01/21 16:48 ~ 2025/01/21 17:48), a category selector, a dropdown for 'Personal Information Item' (個人情報項目), a text input for 'Personal Information Pattern' (個人情報パターン), a toggle for 'Apply Masking' (置換を適用するかどうか), and a 'Masking Value' (置換値) input field. At the bottom of the left pane are buttons for 'Start Check' (点検開始) and 'Stop Check' (チェックを中止).

The right pane is a modal titled 'Personal Information Non-identification Addition' (個人情報の非識別化 追加). It contains a 'Category' (カテゴリ) dropdown menu with the prompt 'Please select a category.' (カテゴリを選択してください。), a 'Search Key' (検索キー) dropdown menu with the prompt 'Please select a search key to be masked and masked.' (暗号化保存およびマスキング処理する検索キーを選択します。), and a text input field for 'Please enter a search key to be masked.' (暗号化する検索キーを入力してください。). An 'Add' (追加) button is located at the bottom right of the modal.

1. 個人情報の非識別化タブで追加するボタンを選択します。
2. 右側の個人情報の非識別化 追加画面でカテゴリを選択し、マスキング処理する検索キーを入力します。

3. 追加ボタンをクリックして設定を追加します。
4. 個人情報の非識別化一覧で非識別化項目を確認した後、右上の保存ボタンを押して現在の設定を保存してください。

個人情報の非識別化 個人情報が含まれた検索キーを追加すると、当該データを暗号化して保存し、マスキングします。 + 追加する 保存			
活性化	カテゴリ	検索キー	修正
☐	VirtualLog	age.n	編集 削除
☒	VirtualLog	status	編集 削除
☒	AppLog	agenttime	編集 削除

- 非識別化処理された検索キーの場合、ログコンテンツ内の **検索キー.pii** 形式で出力されます。
- 個人情報の非識別化一覧内の活性化カラムで非識別化項目を ☒ 活性化にできます。
- 個人情報の非識別化一覧内の編集カラムで [編集](#) 及び [削除](#) できます。

✔ **ログプライマリパーサ設定**タブで検索キー名を **検索キー.p** に指定すると、非識別化対象として自動登録されます。自動登録された項目は**個人情報の非識別化**タブで**カテゴリ**選択後、**指定された検索キー**で確認または解除できます。

パーサーで .p 接尾辞を指定する方式よりも、**個人情報の非識別化**タブで直接設定することを推奨します。

個人情報の表示

非識別化されたデータは**ログTail**、**ログトレンド**、**ログ検索メニュー**でマスキング状態で表示されます。**ログの個人情報照会権限**のあるユーザーはログ一覧内の**個人情報の表示** ☒ ボタンをクリックすると元のデータを確認できます。

▼	java-demo2	2024-12-16 16:06:49.988	area river lprivacy true status.pii *** pcode 8 city seoul onodeName node-0 myname.p *** oid -1737750367 okind 1152715164 price.n 12000 oname java-demo2 age.n 33 agenttime 1734332809555 category VirtualLog okindName dev-okind-0 onode 1693789385 You need not love me, if you choose not to.
▶	java-demo1	2024-12-16 16:06:49.993	area mountain lprivacy true status.pii *** pcode 8 city junju onodeName node-1 myname.p **** Waiting does not have to aim for a meeting.
▼	java-demo2	2024-12-16 16:06:49.988	area river lprivacy true status.pii 500 pcode 8 city seoul onodeName node-0 myname.p lee oid -1737750367 okind 1152715164 price.n 12000 oname java-demo2 age.n 33 agenttime 1734332809555 category VirtualLog okindName dev-okind-0 onode 1693789385 You need not love me, if you choose not to.
▶	java-demo1	2024-12-16 16:06:49.993	area mountain lprivacy true status.pii 200 pcode 8 city junju onodeName node-1 myname.p hong Waiting does not have to aim for a meeting.
▶	java-demo6	2024-12-16 16:06:50.064	area sea lprivacy true status.pii 200 pcode 8 city daegu onodeName node-1 myname.p song Waiting does not have to aim for a meeting.

❗ **個人情報の非識別化**タブで指定した検索キーは、ログメニュー内のログコンテンツから **検索キー.pii** 形式で表示されます。

例: **status.pii** 例 ***

ログの個人情報の非識別化点検

❗ **ログ個人情報の非識別チェック**機能を通じて既存のデータを置き換える場合、そのデータは元に戻すことができません。

ログ個人情報の非識別チェックは既存のログデータ内の個人情報を識別し、ユーザーが指定した**置換値**に置き換えます。

対象キー指定による置換

対象キーを指定すると、該当キーに保存された値のみを置換し、残りのログ内容はそのまま維持します。対象キーはログの **field** 値と **tag** 値の両方に適用されます。対象キーを指定しない場合、従来と同様にログコンテンツ全体から個人情報を識別して置換します。

❗ 置換は保存されたログデータにのみ適用されます。検索インデックスには、置換後も元の値が残っている場合があります。

対象キーは、ログ個人情報の非識別化点検フォームの**対象キー**入力欄で指定し、画面では **field** と **tag** を区別せず、1つの対象キーとして入力します。

ログ個人情報の非識別チェック

収集されてあるログを非識別化します。

📅

✓ チェック履歴の確認

時間

< 2025/03/21 12:26 ~ 2025/03/21 13:26 1時 📅 >

カテゴリ

VirtualLog

個人情報項目

個人識別番号

個人情報パターン

(?<!\w)\d(2)(0[1-9]|1[0-2])\{0-2\}[0-9]{3[0-1]}-[1-4]\d(6)(?\w)

置換の適用有無

☐

置換値 ①

*****_*****

カテゴリ

VirtualLog

検索開始時刻

2025/03/21 12:26

検索終了時刻

2025/03/21 13:26

個人情報項目

個人識別番号

個人情報パターン

(?<!\w)\d(2)(0[1-9]|1[0-2])\{0-2\}[0-9]{3[0-1]}-[1-4]\d(6)(?\w)

置換値

*****_*****

パターンのマッチング結果

0 ✓ 結果の確認

チェック作業の開始時刻

2025/03/21 13:28

チェック作業の終了時刻

2025/03/21 13:28

置換の適用有無

No

チェック作業の結果

SUCCESS

1. 開始時間と終了時間を設定してください。
2. カテゴリを設定します。
3. 個人情報項目を例の中から選択または**直接入力**を選択します。

① 直接入力選択時に個人情報を識別するための正規表現式パターンを直接入力します。

- 置換の適用有無トグルで個人情報が識別された場合、該当情報の置換有無を選択します。
- 既存のログで正規表現パターンに識別された文字列を変更する置換値を入力します。

- ❗ 置換値は元の値より長くしてはなりません。
- 置換の適用有無トグルの活性化時に、必ず置換値を入力する必要があります。

6. 条件を入力した後、**チェック開始ボタン**を選択します。

① 点検中の場合はいつでも**チェック中止**ボタンで点検過程を中止できます。

7. 置き換えを適用する時個人情報の非識別チェックの開始確認画面で置き換え入力後確認ボタンをクリックします。

個人情報の非識別チェックの開始



選択した個人情報パターンに一致する値を置き換えます。

①置き換え後はデータの復元ができません。

置き換えが必要な場合は "**置き換え**" を直接入力してください。

8. 点検作業の完了後、例の画像のように右側から**チェック作業の結果**を確認できます。

- **カテゴリ**：点検されたカテゴリ情報を表示します。
- **検索開始時刻及び検索終了時刻**：点検が実行された時間範囲を表示します。
- **個人情報項目**：点検された個人情報の項目を表示します。
- **個人情報パターン**：使用された正規表現式のパターンを表示します。
- **置換値**：適用された置換値を表示します。
- **パターンのマッチング結果**：識別された個人情報の数を表示します。 **結果の確認**をクリックするとサンプルログを確認できます。
- **チェック作業の結果**：個人情報が識別された結果を表示します。

SUCCESS (点検成功)、 FAILED (点検失敗)、 STOP (点検中止)、 TIMED_OUT (時間超過による点検終了)

チェック履歴の確認

ログ個人情報の非識別化点検を行った後、過去に行われた点検結果を照会するには、上段右側で**チェック履歴の確認**ボタンをクリックします。置き換え履歴、点検時点、適用された個人情報項目などを追跡できます。また**チェック履歴の確認**画面の上段右側の **CSV** ボタンをクリックすると_CSV_ファイル形式でそのリスト情報をダウンロードできます。

 CSV

	パターンのマッチング...	チェック作業の開始時刻	チェック作業の終了時刻	置換の適用有無	チェック作業の結果
***	0	2025/03/25 09:16	2025/03/25 09:16	No	SUCCESS
***	0	2025/03/25 09:16	2025/03/25 09:16	No	SUCCESS
***	0	2025/03/25 09:16	2025/03/25 09:16	No	SUCCESS

個人情報のパターン

既存のログデータから個人情報を識別する際に使用される正規表現式のパターンの例は以下の通りです。

個人情報 の パターン	個人情報の例	正規表現パターン	置換される 値
個人 識別 番号	123456-1234567	(?!w)d{2}(0[1-9] 1[0-2])([0-2][0-9] 3[0-1])-[1-4]d{6}(?!w)	*****_ *****
パ	M1234567	([MSRODTC][0-9]{7}\b)	*****

個人情報のパターン	個人情報の例	正規表現式パターン	置換される値
スポーツ番号			
運転免許証番号	01-23-456789-00	(?<!w)d{2}-d{2}-d{6}-d{2}(?!w)	***_**_ *****_**
外国人登録番号	987654-1234567	(?<!w)d{6}-?[5-8]d{6}(?!w)	*****_ *****
クレジット	1234-5678-9012-3456	(?<!w)d{4}-d{4}-d{4}-d{4}(?!w)	****_****_ ****_****

個人情報のパターン	個人情報の例	正規表現パターン	置換される値
カード番号			
携帯電話番号	010-1234-5678	(?<!w)(+82-?1[016789]-?d{3,4}-?d{4} 01[016789]-?d{3,4}-?d{4})(?!w)	***_***_ ****
メールアドレス	example@domain.com	([a-zA-Z0-9][a-zA-Z0-9._%+~]*@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,})	***@***.***
銀行口座番号	123456-01-654321	(?<!w)d{6}-d{2}-d{6}(?!w)	*****_**_ *****

個人情報のパターン	個人情報の例	正規表現パターン	置換される値
事業者登録番号	111-11-01111	(?<!w)d{3}-d{2}-d{5}(?!w)	***_**- *****
法人登録番号	111111-1111111	(?<!w)d{6}-?d{7}(?!w)	*****_ *****

ログパーシング

ログパーサを使用すると、不規則な形のログをクエリが可能な構造化された形に変更できます。WhaTapログモニタリングは、次の2つのタイプのパーサを提供します。

- **GROKパーサ**：任意の形で収集されるログを正規表現とGROKを活用してパーシングします。
- **JSONパーサ**：JSON形式で収集されるログをパーシングします。

ⓘ 共通の注意事項

- 同じカテゴリに複数のパーサが登録されている場合、最初にマッチングされるパーサのみが適用されます。
- WhaTapはWhaTapサービスの安定性に影響を与えるパーサを無効にする場合があります。。

GROKパーサ

ログが不規則な形で収集される場合、GROKパーサを使用してログをパーシングできます。GROKパターンは、named regular expressionsを提供し、正規表現をより簡単かつ便利に使用できます。

GROKを開始する

GROKは2種類の文法を提供します。

1. `%{SYNTAX:SEMANTIC}`：GROKライブラリーが提供するパターンです。**named regular expressions**を活用してタグを抽出できます。活用例は[次のリンク](#)を参照してください。
 - **SYNTAX**：GROKが提供するnamed regular expressionsを指定します。
 - **SEMANTIC**：マッチングされる値に付ける名前を指定します。

ⓘ named regular expressions

GROKが提供するパターンです。複雑な正規表現に名前を付けて使用できるよう、GROKが提供す

❗ る機能です。

name	regular expression
WORD	<code>\b\w+\b</code>
SPACE	<code>\s*</code>
NOTSPACE	<code>\S+</code>
UUID	<code>[A-Fa-f0-9]{8}-(?:[A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}</code>

WhaTapで提供するすべてのnamed regular expressionsを確認したい場合は、次の[リンク](#)を参考にしてください。

2. `(?<SEMANTIC>REGX)` : 正規表現の**named capturing group**文法です。正規表現を活用して、ユーザーの意図に合わせてタグを抽出できます。活用例は[次のリンク](#)を参照してください。

- **SEMANTIC** : マッチングされる値に付ける名前を指定します。
- **REGX** : マッチングに使用する正規表現を入力します。

❗ named capturing group

正規表現の文法です。

- capturing group: 複数のトークンを1つにまとめて1つのマッチング単位で使用する機能を意味します。
- named capturing group: capturing groupに名前を付けたものです。
- 文字列マッチングの例を見てみましょう。 dev@whatap.io
 - 例1 : `(\w+)@(\w+\.\w+)`
 - 例2 : Eメール全体のマッチング、およびusernameとdomainを追加でマッチングする場合 `(?<username>\w+)@(?<domain>\w+\.\w+)`

%{SYNTAX:SEMANTIC} 活用の例

次は %{SYNTAX:SEMANTIC} を活用する例です。

Sample log

```
[2023-08-08 02:02:30,101 GMT][INFO ][i.w.y.l.c.LogSinkDexScheduleThread.realProcess(159)] 8 VirtualLog 20230808
02:01:00.000 {area=4, city=5} 56ms
```

サンプルログを見て、各単語が意味する内容を推測できます。各部分をsemanticな単語に置き換えると、次のように表現できます。

semantic replace

```
[date][logLevel][caller] projectCode logCategory dexBuildStartTime {area=areaEnum, city=cityEnum} dexBuildElapsed
```

semanticな単語すべて、正規表現に書き換えることができます。GROKパーサを使用すると、事前に定義されたnamed regular expressionsを活用できます。ここで使われた `TIMESTAMP_ISO8601`、`LOGLEVEL`、`DATA` は、GROKが提供する named regular expressionsです。これらの値は、それぞれ次の正規表現に置き換えられてマッチングされます。

- name: `TIMESTAMP_ISO8601`
 - regular expression: `%{YEAR}-%{MONTHNUM}-%{MONTHDAY}[T|t]%{HOUR}:%{MINUTE}(:%{SECOND})?%{ISO8601_TIMEZONE}?`
- name: `LOGLEVEL`
 - regular expression: `LOGLEVEL`
`([Aa]lert|ALERT|[Tt]race|TRACE|[Dd]ebug|DEBUG|[Nn]otice|NOTICE|[Ii]nfo|INFO|[Ww]arn?(?:ing)?|WARN?(?:ING)?|[Ee]rr?(?:or)?|ERR?(?:OR)?|[Cc]rit?(?:ical)?|CRIT?(?:ICAL)?|[Ff]atal|FATAL|[Ss]evere|SEVERE|EMERG(?:ENCY)?|[Ee]merg(?:ency)?)`
- name: `DATA`
 - regular expression: `.*`

GROK parsing pattern

```
\[%{TIMESTAMP_ISO8601:date}\sGMT\s\]\[%{LOGLEVEL:level}\s\]\[%{DATA:caller}\]
```

上記の内容でパースングすると、次のようにタグを抽出できます。このようにGROKの `%{SYNTAX:SEMANTIC}` パター

ンは、複雑で長い正規表現を簡単に適用できる機能を提供します。

Tag extraction

```
- date : 2023-08-08 02:02:30,101
- caller : i.w.y.l.c.LogSinkDexScheduleThread.realProcess(159)
- level : LEVEL
```

(?<SEMANTIC>REGX) 活用の例

named regular expressionsでマッチングされない部分は (?<SEMANTIC>REGX) パターンを使ってパースできます。上記の[サンプルログ](#)から `%{SYNTAX:SEMANTIC}` パターンだけでパースされない領域は次の通りです。

Unparsed area

```
8 VirtualLog 20230808 02:01:00.000 {area=4, city=5} 56ms
```

該当ログの各部分をsemanticな単語に置き換えると、次のように表現できます。

semantic replace

```
projectCode logCategory dexBuildStartTime {area=areaEnum, city=cityEnum} dexBuildElapsed
```

このように不規則な形の文字列は、次のような (?<SEMANTIC>REGX) パターンを使ってパースできます。

サンプルログパースキーワード別にマッチングされる正規表現

パースキーワード	(?<SEMANTIC>REGX)
8	(?<projectCode>\d)
VirtualLog	(?<logCategory>\w*)
20230808 02:01:00.000	(?<dexBuildStartTime>\d{8}\s\d{2}:\d{2}:\d{2}\.\d{3})
area=4	area=(?<areaEnum>\d)
city=5	city=(?<cityEnum>\d)

パーシングキーワード	(?<SEMANTIC>REGX)
56 ms	(?<dexBuildElapsed>\d{2})ms

正規表現の基本文法

文法	意味	別名
?	0 or 1	-
+	1 or more	-
*	0 or more	-
a{5}	exactly 5	-
\w	word character	[a-zA-Z_0-9]
\s	white space	-
.	any character except newline	
[abc]	any of	-
[^abc]	not a,b, or c	-
[a-z]	character between a and z	-
[1-3[7-9]]	union (combining two or more character classes)	-
[1-6&&[3-9]]	intersection (共通集合)	-
[0-9&&[^2468]]	subtraction (差集合)	-
a{2,}	2 or more	-
a{1,3}	between 1 and 3	-



文法	意味	別名
<code>a+?</code>	match as few as possible	-
<code>{2,3}?</code>	match as few as possible	-
<code>(abc)</code>	capturing group (複数の文字列をsingle unitで処理する)	-
<code>\d</code>	digit	<code>[0-9]</code>
<code>\D</code>	non-digit	<code>[^0-9]</code>
<code>\W</code>	non-word character	-
<code>\S</code>	non-white space	-

このようにパージングされたキーワードをスペース(`\s`)と、特殊文字escape(`\{`、`\,`、`\}`)で接続すると、次のようにパターンを適用できます。

GROK parsing pattern

```
(?<projectCode>\d)\s(?<logCategory>\w*)\s(?<dexBuildStartTime>\d{8})\s\d{2}:\d{2}:\d{2}\.\d{3})\s{\area=(?<areaEnum>\d),\scity=(?<cityEnum>\d)\}\s(?<dexBuildElapsed>\d{2})ms
```

上記の内容でパージングすると、次のようにタグを抽出できます。

Tag extraction

```
- projectCode : 8
- logCategory : VirtualLog
- dexBuildStartTime : 20230808 02:01:00.000
- areaEnum : 4
- cityEnum : 5
- dexBuildElapsed : 56
```

GROKを適用する

ログ設定 > ログプライマリパーサ設定

1. GROKパターンパーサを適用するには、**ログ設定**メニューの**ログプライマリパーサ設定**タブに移動します。
2. **+ 追加する**を選択した後**パーサ入力欄**で**GROK**パーサを選択します。
3. **パターン登録**ボタンを選択すると、右側にパターン登録とシミュレーション画面が表示されます。
4. **パターン**と**ログ**入力後、**シミュレーション**ボタンをクリックして、適用するパターンでパーシングに成功することを確認します。

パターン 例:

```
[%{TIMESTAMP_ISO8601:date}\sGMT][%{LOGLEVEL:level}\s][%{DATA:caller}]\s(?:<projectCode>\d)\s(?:<logCategory>\d{2}:\d{2}:\d{2}\.\d{3})\s\s{area=(?:<areaEnum>\d),\scity=(?:<cityEnum>\d)\}\s(?:<dexBuildElapsed>\d{2})ms
```

ログ例:

```
[2023-08-08 02:02:30,101 GMT][INFO ][i.w.y.l.c.LogSinkDexScheduleThread.realProcess(159)] 8  
VirtualLog 20230808 02:01:00.000 {area=4, city=5} 56ms
```

5. シミュレーションの成功時**シミュレーション結果**と**性能測定の結果**を照会できます。
6. シミュレーション後、**パターン適用**ボタンをクリックすると、選択したパーサーに入力したパターンが適用されます。
7. パターン適用後、**カテゴリ**と**ログ検出条件**、**パターン**を入力します。

×

パーサ 追加

パーサ*

WhaTap Monitoringには、サービスの信頼性に影響を与える可能性があるパーサーを無効にできる権限があります。

GROK

パーサーパターン*

入力したパターンでログが正常に解析されるかどうかをシミュレートし、パターンを登録します。

成功しました。

```

\[%{TIMESTAMP_ISO8601:date}\sGMT\}\[%{LOGLEVEL:level}\s\}\[%{DATA:caller}\}\s(?:<projectCode>\d)\s(?:
<logCategory>w*)\s(?:<dexBuildStartTime>\d{8}\s\d{2}:\d{2}:\d{2})\s\{area=(?:<areaEnum>\d),\scity=(?
<cityEnum>\d)\}\s(?:<dexBuildElapsed>\d{2})ms

```

パーサーパターンの修正

カテゴリ*

同じカテゴリに複数のパーサーが登録されている場合、最初に一致するパーサーのみが適用されます。

AppLog

ログ検出条件

フィルタを入力しなかった場合、または検索キー／値を1つだけ入力した場合、そのパターンはすべてのログに適用される。

oname

demo-8100

追加

• カテゴリ

ログカテゴリを選択します。 **カテゴリ**は必須入力項目です。

• ログ検出条件

- 条件を満たすログのみパーサが適用されます。
- 検索キーと検索値を選択するか、直接入力します。
- ログ検出条件は、すべてのパーサが実行される前に適用されます。つまり、パーサの結果として追加されるタグを使用できません。

8. 追加ボタンを選択してパーサを登録します。

- !

 ログパーサー一覧から該当パーサの優先順位を変更するか、**活性化**および**修正**、**削除**できます。
- パーサーシミュレーション後、パターンを登録できます。

434

❗ GROKパーサの注意事項

- GROKパーサは `%{SYNTAX:SEMANTIC}` 、 `%{SYNTAX:SEMANTIC}` 二つのパターンをサポートします。
- `%{SYNTAX:SEMANTIC}` パターンを使用する時、 `SEMANTIC` を必ず入力します。
- `%{SYNTAX:SEMANTIC}` パターンを使用する時、 `SEMANTIC` は一つのパーサの中でユニークである必要があります。
- `(?<SEMANTIC>REGX)` パターンを使用する時、 `SEMANTIC` は文字(a-z、A-Z)、数字(0-9)、および指定された特殊文字(`.` 、 `_` 、 `-`)のみが使えます。
- `SEMANTIC` は文字(a-z、A-Z)で始めなければなりません。
- `SEMANTIC` は文字(a-z、A-Z)または数字(0-9)で終わる必要があります。

GROK追加パターンの選択

GROKパーサのシミュレーション時に、基本パターン(`WORD` 、 `NUMBER` 、 `IP` など)以外に特定のログ形式に合った追加パターンを選択できます。Java、Nginx、MySQLなど32個のパターンタイプが11個のカテゴリにグループ化されて提供されます。

追加パターンカテゴリのドロップダウン

1. パーサのシミュレーション画面で**追加パターン (選択)**領域のドロップダウンをクリックしてください。
2. カテゴリ別にグループ化されたパターンタイプ一覧から、必要なパターンタイプを選択してください。検索入力欄にパターンタイプ名または説明を入力して検索することもできます。
3. 選択したパターンタイプは、シミュレーション時に基本パターンとともにロードされます。

❗ 基本パターン(Core)は常に自動的に含まれます。追加パターンの選択は任意です。

カテゴリ別パターンタイプ一覧

カテゴリ	パターンタイプ
Web Servers	haproxy, httpd, iis, nginx, squid

カテゴリ	パターンタイプ
Application Servers & Frameworks	java, log4j, rails, ruby, springboot
Databases	mongodb, mysql, oracle, postgresql, redis
Cloud	aws
System & Infrastructure	docker, kubernetes, linux-syslog
Messaging & Search	elasticsearch, kafka
Network & Security	bind, bro, firewalls, junos, zeek
Mail	exim, postfix
Monitoring & Backup	bacula, mcollective, nagios
Build Tools	maven

活用例

Tomcat Catalina 8のログをパースングするには、**java**パターンタイプを追加で選択した後、`%{CATALINA8_LOG}` パターンを使用してください。

Sample log

```
31-Jul-2020 16:40:38 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Server version name: Apache Tomcat/8.5.57
```

GROK parsing pattern

```
%{CATALINA8_LOG}
```

Tag extraction

```
- log_timestamp : 31-Jul-2020 16:40:38
- log.level : INFO
```



```
- java.log.origin.thread.name : main
- java.log.origin.class.name : org.apache.catalina.startup.VersionLoggerListener
- log.origin.function : log
- log_message : Server version name: Apache Tomcat/8.5.57
```

パターン詳細表示

選択した追加パターンタイプにどのパターンが含まれているかを詳細画面で確認できます。パターン定義(正規表現)とサンプルログを照会し、シミュレーションにすぐに適用できます。

パターン詳細表示

追加パターンタイプを選択した後、選択されたパターンタイプのタグをクリックすると、右側に詳細画面が開きます。詳細画面はシミュレーションフォームを隠さずに並んで表示されるため、パターン定義を確認しながらシミュレーションを進めることができます。

各パターン項目で次の機能を使用できます。

ボタン	機能	説明
サンプル	サンプルログ入力	該当パターンにマッチングする例ログをシミュレーションパースングログ入力欄に自動的に入力します。
選択	パターン挿入	<code>%{パターン名}</code> 形式でパターン入力欄に挿入します。該当パターンタイプも自動的に選択されます。
コピー	クリップボードコピー	<code>%{パターン名}</code> 形式をクリップボードにコピーします。

- ① パターン項目をクリックすると、パターン定義(正規表現)が展開されます。複数のパターンを同時に展開して定義を比較できます。
- パターン定義に含まれる `%{PATTERN:field}` 参照は、パターン名、フィールド名、リテラル文字がそれぞれ異なる色で区別されて表示されます。
- サンプルボタンで入力されたログでそのままシミュレーションを実行してパースング結果を確認できます。

シミュレーションサンプルログの保存

シミュレーションに使用したログは、パーサ設定とともに自動的に保存されます。パーサを修正する際に、以前に使用したサンプルログがシミュレーション画面に自動的に復元され、毎回ログを再入力する必要がありません。

JSONパーサ

ログがJSONフォーマットで収集される場合、JSONパーサを使用して簡単かつ便利にパーシングできます。

JSONを適用する

ログ設定 > ログプライマリパーサ設定

1. JSONパターンパーサを適用するには**ログ設定**メニューの**ログプライマリパーサ設定**タブに移動します。
2. **+ 追加する**を選択した後**パーサ入力欄**で**JSON**パーサを選択します。
3. **パターン登録**ボタンを選択すると、右側にパターン登録とシミュレーション画面が表示されます。
4. **パターン**と**ログ**入力後、**シミュレーション**ボタンをクリックして、適用するパターンでパーシングに成功することを確認します。

Example

```
2023-08-08 02:43:28,615 -- {"host":"10.21.3.24","method":"POST","status":"200","url":"http://dev.whatap.io/yard/api/flush"} --
```

例のログで**Prefix**と**Postfix**を `--` に指定し、**Ignore**に `url` を指定時 `host`、`method`、`status` 3つ**タグ**のみ生成します。

- **Prefix** ログでJSONフォーマットが開始する場所を指定します。ログ全体がJSONフォーマットの場合は、空の値に設定します。
 - **Postfix** ログでJSONフォーマットが終わる位置を指定します。ログ全体がJSONフォーマットの場合は、空の値に設定します。
 - **Ignore** JSONフォーマットの中でタグを生成しないキーを指定します。
5. シミュレーションの成功時**シミュレーション結果**と**性能測定の結果**を照会できます。

×

パーサ シミュレーション

ガイド表示

パターン

agenttime, category, content, logContent, message, pcode, time, timestamp は予約語です。
予約語はログ検出条件、パターンで登録しても動作しません。

Prefix

-

Postfix

-

Ignore

url

ログ *

ログシミュレーションは必須ですする必要があり、
シミュレーションの結果に応じてパターン登録が必要になる場合があります。

2023-08-08 02:43:28.615 -
{"host":"10.21.3.24","method":"POST","status":"200","url":"http://devote.whatap.io/yard/api/flush"} -
-

シミュレーション

シミュレーション結果 成功しました。

キー	値	結果
method	POST	Ok
host	10.21.3.24	Ok
status	200	Ok

性能測定の結果 成功しました。

シミュレーション数	最小時間(ns)	最大時間(ns)	平均時
1	4,491	4,491	4,491
10	3,322	3,930	3,525
100	2,662	177,949	8,084
1,000	2,654	476,818	4,305
10,000	2,648	311,233	3,823
100,000	2,650	60,522,941	3,713
1,000,000	2,647	44,113,067	3,014

パターン適用

6. シミュレーション後、パターン適用ボタンをクリックすると、選択したパーサーに入力したパターンが適用されます。

7. パターン適用後、カテゴリとログ検出条件、パターンを入力します。

439

×

パーサ 追加

パーサ*

WhaTap Monitoringには、サービスの信頼性に影響を与える可能性があるパーサーを無効にできる権限があります。

JSON

パーサーパターン*

ログが正常に解析されるかどうかをシミュレートします。シミュレーションは必須でなければならず、シミュレーション結果によってはパターン登録がさらに必要になる場合があります。

✔ 成功しました。

Prefix --

Postfix --

Ignore url

シミュレーション検証とパターン修正

カテゴリ*

同じカテゴリに複数のパーサーが登録されている場合、最初に一致するパーサーのみが適用されます。

AppLog

ログ検出条件

フィルタを入力しなかった場合、または検索キー／値を1つだけ入力した場合、そのパターンはすべてのログに適用される。

oname

demo-8100

追加

• カテゴリ

ログカテゴリを選択します。 **カテゴリ**は必須入力項目です。

• ログ検出条件

- 条件を満たすログのみパーサが適用されます。
- 検索キー**と**検索値**を選択するか、直接入力します。
- ログ検出条件**は、すべてのパーサが実行される前に適用されます。つまり、パーサの結果として追加される**タグ**を使用できません。

8. 追加ボタンを選択してパーサを登録します。

- ⓘ ログパーサー一覧から該当パーサの**優先順位**を変更するか、**活性化**および**修正**、**削除**できます。
- パーサを登録する前に**シミュレーション**を通じて登録するパターンが正常であるか確認できます。GROK

440



パーサ登録シミュレーションプロセスと同じです。 [次の文書](#)を参考にしてください。

JSONフォーマット活用例

Sample log

```
{"host":"10.21.3.24","method":"POST","status":"200","url":"http://dev.whatap.io/yard/api/flush"}
```

上記のようなサンプルログが収集された場合パーサ 追加画面でJSONパーサを選択します。複雑なパーシングロジックを作成する必要なく、ログ分析に必要なタグを次のように抽出できます。

Tag extraction

```
- host : 10.21.3.24
- method : POST
- status : 200
- url : http://dev.whatap.io/yard/api/flush
```

JSONフォーマットを一部構成する時の活用例

Some JSON format sample log

```
2023-08-08 02:43:28,615 -- {"host":"10.21.3.24","method":"POST","status":"200","url":"http://dev.whatap.io/yard/api/flush"}
--
```

もし、例のようにログの一部のみがJSONフォーマットで構成されている場合はPrefixとPostfixを指定してください。WhaTapログモニタリングはPrefixとPostfix間の領域をJSONフォーマットで認識してからパーシングします。

Tag extraction

```
- host : 10.21.3.24
- method : POST
- status : 200
- url : http://dev.whatap.io/yard/api/flush
```

警告アラート

監視対象のシステムまたはアプリケーションで問題が発生した際に、あらかじめ設定した条件に基づいてユーザーへ通知を送信する機能です。通知は、メール、SMS、メッセージャー、アプリのプッシュ通知など、さまざまなチャネルを通じて配信されます。

たとえば、CPU 使用率、メモリ使用量、ディスク I/O、ネットワークトラフィック、トランザクション数、エラー発生、レスポンスタイムの遅延などの主要なパフォーマンス指標にアラート基準を設定しておく、その基準を超過した場合や異常な兆候が検出された際に通知が送信されます。

WhaTap の AI ベースのアラート機能を使用すると、複雑な条件を手動で設定する必要はありません。AI がアプリケーションの正常な動作パターンを学習し、異常パターンを自動的に検知して通知します。また、タグカウントアラート機能を活用することで、特定のタグを基準に集計された値に条件を設定でき、複雑な状況でも高精度な検知が可能です。

主な機能

- ・ **イベント設定:** 閾値ベースのアラート条件および通知チャネルの設定
- ・ **通知設定:** ユーザーごとの通知チャネルおよび時間帯設定
- ・ **イベント履歴:** アラート発生履歴の確認

権限

イベントを追加または設定するには、**アラート設定権限** が必要です。

- ・ **閲覧:** メトリクスイベント設定を確認するには、**閲覧・分析権限** が必要です。
- ・ **追加／編集／削除:** 設定を変更するには、**アラート設定権限** が必要です。
- ・ **通知受信:** 通知を受信するには、**通知受信権限** が必要です。

📌 参考資料

詳細については、[メンバー権限](#) を参照してください。

イベント設定

イベント設定では、監視対象のシステムやアプリケーションで発生する可能性のある異常兆候を基準に、イベントを設定・管理できます。設定した条件が満たされると、メール、SMS、メッセージャー、アプリのプッシュ通知など、希望するチャンネルで通知を受信できます。監視の目的に応じて、イベントを柔軟に構成できます。

メトリクス

メトリクスイベントは、リアルタイムデータを基にシステムおよびアプリケーションのパフォーマンス指標を監視できます。

複合メトリクス

複合メトリクス機能では、複雑なルールを活用してイベントを作成し、警告アラートを送信できます。

異常値

異常値機能では、想定外のパターンを検出して警告アラートを送信するよう設定できます。

リアルタイムログ

リアルタイムで収集されたログから設定したイベント条件が検出された場合、警告アラートを送信します。

複合ログ

ログの内容を基準に警告通知を発生させる複合ログイベントを設定します。

ホーム画面 > プロジェクト選択 >  アラート通知 > イベント設定

イベント設定では、監視対象のシステムやアプリケーションで発生する可能性のある異常兆候を基準に、イベントを設定・管理する機能を提供します。設定した条件を満たすと、メール、SMS、メッセージャー、アプリプッシュなど、希望するチャンネルで通知を受信できます。監視目的に応じてイベント構成を設計できます。

メトリクス

メトリクスイベントは、リアルタイムデータを基にシステムおよびアプリケーションのパフォーマンス指標を監視できます。

複合メトリクス

複合メトリクス機能では、複雑なルールを活用してイベントを作成し、警告アラートを送信できます。

異常値

異常値機能では、想定外のパターンを検出して警告アラートを送信するよう設定できます。

リアルタイムログ

リアルタイムで収集されたログから設定したイベント条件が検出された場合、警告アラートを送信します。

複合ログ

ログの内容を基準に警告通知を発生させる複合ログイベントを設定します。

参考



- イベント設定時は、不要なイベント発生を防ぐために適切なしきい値を設定してください。
- 通知設定では、問題解消時に通知される回復通知もあわせて検討すると、運用対応に役立ちます。

共通機能

すべてのイベント設定は **JSON** または **Excel 形式** でダウンロードできます。同じ種類のプロダクト設定を共有することで、繰り返し発生するイベント設定作業を削減できます。

- **JSON 一括編集**: すべてのイベント条件を JSON 形式で一括編集できます。
- **JSON 一括ダウンロード**: 現在登録されているイベント条件を JSON ファイルとして保存できます。
- **Excel 一括ダウンロード**: イベント一覧を Excel ファイルとしてダウンロードし、外部レポートやバックアップに活用できます。

JSON 一括編集

1. 画面上部の [JSON 一括編集] ボタンをクリックします。
2. **JSON 一括ダウンロード** 機能でダウンロードした JSON ファイルを読み込むか、直接編集します。
3. [上書き] ボタンをクリックして保存します。
4. 確認メッセージが表示されたら、[上書き] ボタンをクリックします。



上書き後は、以前のイベント設定に戻すことはできません。

- `productType` または `platform` プロパティの値を任意に変更すると、他のプロジェクトでイベントが正常に動作しなくなる可能性があります。
- JSON 編集画面には検証機能が組み込まれています。形式や値が正しくない場合、編集画面下部にエラーメッセージが表示され、[上書き] ボタンは無効になります。



```
},
"basic": [
  {
    "criticalEnabled": "false",
    "warningEnabled": true,
    "silentPeriod": 300000,
  }
]
```

❗ 「criticalEnabled」には、「string」タイプではなく、「boolean」タイプの値が必要です。

読み込み

キャンセル

適用

JSON データ構造



```
{
  "productType": "{productType}",
  "platform": "{platform}",
  "events": {
    "HITMAP": {
      ...
    },
    "LOG_REALTIME": [
      {...}
    ],
    "METRICS": [
      {...}
    ],
    "COMPOSITE_LOG": [
      {...}
    ],
    "BASIC": [
      {...}
    ],
    "COMPOSITE_METRICS": [
      {...}
    ],
    "ANOMALY": [
      {...}
    ]
  }
}
```



}

JSON フィールド	説明
productType	プロダクトタイプ（例: APM、Infra など）
platform	プラットフォーム情報（例: java、node など）
events	イベントデータ
└ HITMAP	ヒットマップパターン
└ LOG_REALTIME	リアルタイムログイベント
└ METRICS	メトリクス
└ COMPOSITE_LOG	複合ログイベント
└ BASIC	基本イベント
└ COMPOSITE_METRICS	複合メトリクス
└ ANOMALY	異常値

JSON 一括ダウンロード

画面上部の [JSON 一括ダウンロード] ボタンをクリックします。

- JSON ファイル名は `integrated-event-YYYYMMDD.json` 形式です。
- JSON データの構造や `events` 属性配下のサブ属性は、プロダクト（ `platform` ）またはプロダクトタイプ（ `productType` ）によって異なる場合があります。



JSON データ内の `productType` または `platform` プロパティの値を任意に変更すると、他のプロジェクトでイベントが正常に動作しなくなる可能性があります。

Excel 一括ダウンロード

画面上部の [Excel 一括ダウンロード] ボタンをクリックします。

メトリクス

ホーム > プロジェクト選択 >  アラート通知 > イベント設定 > メトリクス タブ

プロジェクトに設定されているすべてのメトリクスベースのイベント一覧を確認できます。設定済みのメトリクスイベントを確認、修正、削除したり、新しいイベントを追加したりできます。

基本オプション

- **イベントレベル:** イベントの深刻度を示し、正常（Info）、警告（Warning）、危険（Critical）の3段階に分類されます。
- **エージェント重複アラート抑止:** 同じ名前のエージェントが同時に2つ以上動作している場合、選択した時間（30秒、1分、3分、5分、10分）の間はイベントが発生しません。
- **検索:** イベント名を基準に検索します。
- **JSON ダウンロード／アップロード:** JSON エディタでイベントルールを直接編集するか、ファイルとしてダウンロードして編集後に再アップロードできます。
- **+ イベント追加:** 新しいメトリクスイベントを追加します。
- **カラムサイズ調整:** 各カラムの境界をドラッグして幅を調整できます。

表 | メトリクスイベント一覧構成

項目	説明
No.	イベントの番号
	イベントの有効化状態 – 有効: 条件を満たすとイベントが発生 – 無効: イベントは動作せず、条件も評価されません
	イベント設定の編集および削除
イベント名	ユーザーが指定したイベント名 – 最大255文字

項目	説明
ルール	イベント発生条件
対象	イベントが適用される監視対象
発生回数	イベント発生の基準となる期間と回数 – 選択した時間内に設定回数分条件を満たすとイベントが発生
一時停止	アラート発生後、同一イベントの発生を一定時間停止 – 選択可能時間: 使用しない、5分、10分、15分、20分、30分、1時間、2時間、3時間、6時間、12時間、1日 – 回復通知が有効な場合、RECOVERED（正常）通知後、選択した時間は同一イベントが発生しても通知されません
回復通知	Critical および Warning レベルのイベントが解消された際に、RECOVERED（正常）通知を受信するかどう か – 有効: 進行中から正常状態に遷移すると RECOVERED 通知を送信し、イベント履歴で進行中／解消状態を区別して確認可能 – 無効: しきい値を超えるたびに単発イベントが発生
イベント受信	該当イベント通知を受信するユーザーまたはグループ（受信タグ）

イベント設定

メトリクスイベントは、監視対象で発生する事象を条件に基づいて定義し、効率的に通知を受信できるよう設定するための機能です。

イベント追加

メトリクスイベントを追加するには、**イベント設定** > **メトリクス** タブに移動し、右上の [+ イベント追加] ボタンをクリックします。

メトリクスイベント追加手順
Step 1. テンプレート選択: イベントの基本構成を選択します。
Step 2. イベント条件定義: どの条件でイベントを発生させるかを設定します。
Step 3. イベント対象選択: イベントが適用される監視対象を選択します。
Step 4. 基本情報および通知設定: イベント名、メッセージ、受信対象を設定します。

テンプレート選択

1. イベントの構成方法に応じて、画面左側の3つのテンプレートのいずれかを選択します。
 - **新規作成:** カテゴリやメトリクスを一から自由に構成できます。複雑な条件や詳細な監視ルールが必要な上級ユーザーに適しています。
 - **クイック設定:** あらかじめ定義されたカテゴリ、メトリクス、基本しきい値が提供され、しきい値のみを調整して素早く適用できるため、初心者に適しています。
 - **高度な設定:** あらかじめ定義されたカテゴリやメトリクスを基に、条件を追加したり自由に変更したりできます。

❗ メトリクスイベントでは、さまざまな設定方法をサポートしています。利用する監視プラットフォームによって、提供されるテンプレートは異なる場合があります。

イベント条件定義

イベント条件定義のステップでは、イベント発生の判断に使用するメトリクス設定、発生回数、一時停止、回復通知、イベント動作時間を設定します。

指標設定

1. 検索欄にカテゴリを入力するか、カテゴリー一覧からカテゴリ（指標）を選択します。
指標設定を進めるには、カテゴリの選択が必要です。

- ・ カテゴリ一覧では、カテゴリ名、データ収集間隔、キーを確認できます。直近3時間にプロジェクトで収集されたメトリクスデータが表示されます。

2. 一覧にない指標を使用する場合は、**直接入力**を選択してカテゴリを手動で入力します。

❗ カテゴリや指標が不明な場合は、メトリクス照会を活用してください。

初めて指標を設定する場合、メトリクス照会を利用すると効率的に指標を見つけられます。

- ・ メトリクス照会で指標を検索すると、関連するカテゴリや詳細情報を確認できます。
- ・ 検索結果から必要なメトリクスの[選択]ボタンをクリックすると、選択した値がイベント設定画面に自動的に反映されます。

メトリクス検索

All 13件 / 全 149件

カテゴリ	名前	単位	タイプ	詳細	マージ	
ApplicationHostResource	CpuSys	pct	field	System領域のCPU使用率	OM: avg TM: avg	選択
ApplicationHostResource	CpuIrq	pct	field	Interrupt Request (IRQ)を使用したCPU使用率	OM: avg TM: avg	選択
ApplicationHostResource	CpuSteal	pct	field	実際にCPUを待機する時間を%で表示した値	OM: avg TM: avg	選択
ApplicationHostResource	CpuCores	count	field	CPUコア数	OM: avg TM: avg	選択
ApplicationHostResource	CpuUser	pct	field	User領域のCPU使用率	OM: avg TM: avg	選択
ApplicationHostResource	CpuProc	pct	field	Proc領域のCPU使用率	OM: avg TM: avg	選択
ApplicationHostResource	Cpu	pct	field	CPU使用率	OM: avg TM: avg	選択
ApplicationHostResource	CpuWait	pct	field	ディスク入出力を待つ間(wait)に使用したCPU使用率	OM: avg TM: avg	選択
ProcessPrivate	Private	na		Number of Bytes Private		

< 1 > 50件 / ページ

3. イベントレベル（Critical、Warning、Info）を選択します。1つの指標に対して複数のレベルを設定できます。

❗ イベントレベルの動作

複数の条件を満たしても、最も優先度の高いレベル（Critical → Warning → Info）のみがイベントとして発生します。詳細は [レベル動作ガイド](#) を参照してください。

- ・ イベントレベルは、危険（Critical）、警告（Warning）、正常（Info）に分類されます。
- ・ 同一指標に対して段階的なしきい値でレベルを設定することを推奨します。

4. 選択したイベントレベルのフィールド、演算子、値を入力します。

a. + 追加 をクリックすると条件を追加できます。

- && : すべての条件を同時に満たした場合 にイベントが発生
- || : いずれか 1 つ以上の条件を満たした場合 にイベントが発生

b. - をクリックすると、その条件が削除されます。

❗ 条件式の書き方や対応する演算子については、[条件設定ガイド](#) を参照してください。

発生回数

条件が何回満たされた場合にイベントを発生させるかを設定します。継続的な問題のみを検知したい場合に有効です。

1. 発生頻度を選択します。

- 連続: 設定した回数分、イベントが連続して発生した場合にイベントを発生
- 直近: 選択した時間内に、設定した回数分イベントが発生した場合にイベントを発生
 - 選択可能時間: 5秒, 10秒, 15秒, 30秒, 1分, 2分, 3分, 4分, 5分, 10分, 30分, 1時間

2. 発生回数を入力します。

- Interval の値は、選択したカテゴリのデータ収集間隔を示します。(Interval: 10s)

一時停止

過剰なイベント発生を防ぐため、イベント動作を一時停止する時間を選択してください。イベント発生後、設定した時間が経過するまで同一イベントは発生しません。

- 選択可能時間: 使用しない, 5分, 10分, 15分, 20分, 30分, 1時間, 2時間, 3時間, 6時間, 12時間, 1日

❗ 解消通知が有効な場合

正常状態の通知 (RECOVERED) を受信した時点から、設定した時間が経過するまで、同一イベントが再発しても通知は送信されません。

解消通知

解消通知機能を有効にすると、イベント発生後の状態変化（進行中 → 解消）を追跡できます。**イベント履歴** メニューでは、進行中のイベントとして表示されます。

- 解消通知は **Critical** および **Warning** レベルにのみ適用されます。Info レベルは常に単発イベントとして動作します。
- **Critical** または **Warning** レベルのイベントが解消されると、正常（RECOVERED）状態の通知が送信されます。
- トグルボタンをクリックして **有効化** または **無効化** してください。
 -  **有効**: イベントは状態ベースで動作します。しきい値を超えると進行中となり、値がしきい値未満に回復すると解消状態に遷移します。
 -  **無効**: しきい値を超えるたびに単発イベントが発生します。

イベント動作時間

イベントが特定の時間帯（勤務時間／非勤務時間／点検時間など）にのみ動作するよう、タグを設定してください。タグを設定しない場合、イベントが有効な状態では 24 時間常時動作します。

1. **+ 追加** をクリックします。
2. イベント動作時間で **+ 新しいタグを作成** をクリックします。
3. **イベント動作時間タグ作成** で **タグ名**、**曜日**、**時間**、**色** を選択し、**[適用]** ボタンをクリックします。
4. 作成したタグはタグ一覧で確認でき、チェックボックスを選択すると適用されます。
5. タグを修正または削除するには、 をクリックし、**イベント動作時間タグ修正** で修正または削除してください。

 タグを削除すると、そのタグが適用されているすべてのユーザーからタグが削除されます。ただし、イベントルールで使用中のタグは削除できません。

シミュレーション

シミュレーション機能では、設定したイベント条件を過去データに基づいて事前にテストできます。想定されるイベント件数が多すぎる、または少なすぎる場合は、条件を調整して最適なしきい値を見つけることができます。

- **ルール検証:** 新しく作成したイベントルールを過去データに適用した場合の結果を確認できます。
- **しきい値チューニング:** イベントが頻発する、またはまったく発生しない場合に、適切なレベルへしきい値を調整します。

シミュレーション開始

1. イベント条件の設定が完了したら、[シミュレーション] ボタンをクリックします。
2. 想定イベント件数と可視化された結果を確認します。

シミュレーション結果の理解

1. 時系列データ

指標の時間帯別変化と各レベルのしきい線がグラフで表示され、イベント発生タイミングを直感的に確認できます。

2. レベル別発生状況

画面右側に Critical、Warning、Info レベルごとのイベント発生回数が表示され、全体の分布や頻度を把握できます。

3. 対象別詳細情報

グラフ上の特定ポイントにマウスを合わせると、その時点の正確な指標値と監視対象が表示されます。複数の対象がある場合は色で区別され、対象ごとにパターンが異なる場合はフィルタリング機能で個別分析が可能です。

4. ピーク区間の特定

指標が急激に上昇する時間帯を確認し、問題が頻発する時間帯やパターンを把握できます。

🟢 シミュレーション活用のヒント

• 複数時間帯でのシミュレーション

平日／週末、ピーク／非ピーク時間帯をそれぞれシミュレーションし、時間帯ごとの最適なしきい値を設定してください。



- 実際の障害期間を含める

過去に障害が発生した期間を含めてシミュレーションを実行し、現在のルールが障害を正しく検知できるか確認しましょう。

- 対象フィルタリングの追加

監視対象が多い場合、結果が複雑になります。フィルタリング機能を使って特定の対象のみを選択し、明確な結果を確認してください。

- しきい値を段階的に調整

一度に大きく変更するのではなく、10 パーセント単位で段階的に調整しながら最適値を探すことをおすすめします。

イベント対象選択

イベント対象を指定しない場合、プロジェクト内のすべての対象に対してイベントが発生し、多数のイベントが生成される可能性があります。そのため、イベントを適用する対象を明確に設定することを推奨します。

❗ イベント対象を変更すると、イベント件数が変わる場合があります。シミュレーション ボタンをクリックして再実行し、想定イベント件数を確認してください。

対象選択

イベントを適用する対象は、**選択入力** または **直接入力** の方法で指定できます。

- **選択入力** 直接入力

1. タグ、演算子、値を選択または入力してください。
 - a. タグが見つけない場合は、**メトリクス照会** を使ってタグを検索できます。
2. **+ 追加** をクリックして、条件（**&&**、**||**）でイベント対象を追加してください。

監視対象のタグ、演算子、値を直接入力してください。

- 直接入力を選択すると、入力欄の下に使用可能なタグ一覧（**タグ選択入力**）が表示されます。

```
ex. endsWith(okindName, 'example_name') && container == 'prod.billing'  
ex. ${4xxErrorType} == '401'
```

❗ 入力方法を変更すると、作成中の内容は初期化されます。

基本情報および通知設定

イベント名、メッセージ、受信対象を設定します。設定した内容はイベント履歴や実際の通知メッセージにそのまま表示されるため、識別しやすく分かりやすい名称にすることが重要です。

イベント有効化

設定した条件を満たした際にイベントを発生させるかどうかを設定します。

- ☒ **有効**: 設定した条件を満たすとイベントが発生します。
- ☐ **無効**: 条件を判定しません。

イベント名

イベントのタイトルとして使用するイベント名を入力してください。設定したイベント名は **イベント一覧** および **イベント履歴** メニューに表示され、この名前で検索することができます。

メッセージ

イベントメッセージを入力してください。入力したメッセージは、イベント履歴および通知に使用されます。🕒 をクリックすると、過去に作成したメッセージ履歴を確認できます。メッセージには変数を使用して、イベント発生時点の実際の値を含めることができます。

- **変数の使用ルール**
 - 変数形式: `${指標名}`
 - 設定した **カテゴリ**と同一**カテゴリ内**の指標のみを変数として使用できます。
 - 複数の変数を組み合わせて、より詳細なメッセージを作成できます。
 - 使用可能な変数一覧は **メトリクス照会** メニューで確認できます。

- 変数の使用例

	イベント設定	イベント発生時
タイトル	Disk 使用量増加	Disk 使用量増加
メッセージ	Disk = \${disk}%	Disk = 89.9%

イベント受信

イベント通知を受信するメンバーを指定します。

- 全体受信** タグ選択受信

プロジェクトで通知設定が有効になっているすべての対象（メンバー、チャンネル）に通知が送信されます。

指定したタグを持つプロジェクトメンバーに通知が送信されます。

1. **タグ選択受信** を選択します。
2. **受信タグ** 項目で [+ タグ追加] または [+] ボタンをクリックします。
3. **イベント受信タグ** 画面の下にある + **新しいタグを作成** をクリックします。
4. **タグ作成** 画面でタグ名を入力し、色を選択した後、[タグ作成] ボタンをクリックしてタグを作成します。
5. 作成したタグは、**タグ一覧**の  をクリックして修正または削除できます。
6. **イベント受信タグ** 画面の **タグ一覧** から希望するタグを選択すると適用されます。

ⓘ 受信タグ

受信タグは、通知を受け取るユーザーやチャンネルを **グループ単位で管理するための機能** です。イベントに受信タグを設定すると、そのタグが付与されたユーザーにのみ通知が送信されます。受信タグを設定しない場合、プロジェクト内のすべてのユーザーに通知が送信されます。

イベント修正／削除

1. **警告アラート** > **イベント設定** の **メトリクス** タブに移動します。

2. イベント一覧で、修正または削除したいイベントの  をクリックします。
3. イベントルール修正 画面でオプションを修正し、[保存] ボタンをクリックします。
 - a. 選択したイベントを削除する場合は、イベントルール修正 画面右上の [削除] ボタンをクリックします。

イベント共有

メトリクスイベント設定を JSON ファイルとして保存し、他のユーザーと設定を共有したり、他のユーザーの設定を取り込んだりすることができます。

- JSON ファイル名: `event-rules-YYYYMMDD.json`

エクスポート

1. 画面右上の [JSON ] ボタンをクリックします。
2. JSON 編集画面が表示されたら、[ エクスポート] ボタンをクリックします。
 - イベントを検索した状態でエクスポート機能を使用すると、検索結果のみが JSON ファイルとしてダウンロードされます。
3. JSON ファイルがダウンロードされたら、共有したい他のユーザーに渡してください。

インポート

1. 画面右上の [] ボタンをクリックします。
2. エクスポート 機能でダウンロードした JSON ファイルを選択します。
3. JSON 編集画面が表示されたら、[一覧に追加] または [上書き] を選択します。

 同じ種類の製品間での利用を推奨します。異なる製品のプロジェクトからイベント設定を取り込むことは可能ですが、正常に動作しない場合があります。

JSON 形式で編集する

1. 画面右上の [JSON ] ボタンをクリックします。

2. 編集画面が表示されたら、JSON 形式に従って内容を修正してください。
3. 修正が完了したら、画面下部の [保存] ボタンをクリックします。

❗ 修正した内容が JSON 形式に合っていない場合、画面下部にエラーメッセージが表示され、保存できません。表示されるエラーメッセージは、形式エラーの内容によって異なる場合があります。

JSON error

JSON データ構造

```
{
  "version": 2,
  "eventId": "zcef7s7f27rum1",
  "enabled": true,
  "stateful": false,
  "title": "Active Transaction",
  "message": "Active Transaction = ${active_tx_count}",
  "category": "app_counter",
  "alertLabel": [
    "oid"
  ],
  "repeatCount": 1,
  "repeatDuration": 0,
  "silent": 300000,
  "receiver": [],
  "timeTag": [],
  "selectString": "",
  "conditions": [
    {
      "level": 20,
      "enabled": true,
      "rule": "active_tx_count > 100"
    }
  ],
  "createTime": 1763632674345,
  "lastModifiedTime": 1763632674347,
  "lastModifiedUser": "support@whatap.io",
  "basic": true,
  "metaId": "java004",
}
```



```

"selectCondition": {
  "oid": [
    "-1010758404",
    "-250906941"
  ]
}
}

```

表 | JSON データ構造

JSON フィールド	型	説明	備考
version	Integer	イベントルールのバージョン	変更制限
eventId	String	イベントルールの一意識別子	
enabled	boolean	イベントの有効／無効	
stateful	boolean	状態ベースイベントかどうか	
title	String	ユーザーが設定したイベントルール名	
message	String	ユーザーが設定したイベントルールメッセージ	
category	String	データカテゴリ	
alertLabel	List<String>	イベント状態管理のため、カテゴリごとに指定される基本識別子（プライマリキー）値	変更制限
repeatCount	Integer	繰り返し回数	
repeatDuration	Long	繰り返し期間	
silent	Integer	一時停止時間	
receiver	List<String>	受信タグのキー値一覧	

JSON フィールド	型	説明	備考
timeTag	List<String>	動作時間タグのキー値一覧	
selectString	String	対象選択	
conditions	List<Object>	レベル別の発生条件一覧	
└ level	byte	レベル値	Critical: 30、Warning: 20、Info: 10
└ enabled	boolean	レベルの有効／無効	
└ rule	String	レベル発生条件	
createTime	Long	イベントルールの初回作成時刻	
lastModifiedTime	Long	イベントルールの最終修正時刻	
lastModifiedUser	String	イベントルールを最後に修正したアカウント	
basic	boolean	クイック設定で作成されたイベントかどうか	変更制限
metald	String	クイック設定テンプレートの一意識別子	変更制限: この値はクイック設定で作成されたイベントにのみ存在します (basic=true)。
selectCondition	Object	クイック設定の対象選択値	この値はクイック設定で作成されたイベントにのみ存在します (basic=true)。

❗ 「変更制限」と表示されたフィールドはイベント動作に影響を与える可能性があるため、値を変更しないことを推奨します。

付録

発生条件・対象選択ガイド

メトリクスアラートにおけるイベントの発生条件とイベント対象の選択は、同一の構文を使用します。ただし、イベント発生条件ではフィールド（Field）のキーを変数として使用し、イベント対象の選択ではタグ（Tag）のキーを変数として使用します。

レベル動作ガイド

イベントレベルは、危険（Critical）、警告（Warning）、正常（Info）の3段階に分類されます。

1. 優先順位に基づくイベント発生

複数のレベル条件を同時に満たした場合、最も優先度の高いイベントのみが発生します。

例

```
設定:  
- Warning: CPU > 70%  
- Critical: CPU > 90%  
  
現在の状態: CPU 95%  
→ 結果: Critical イベントのみ発生（Warning は抑制される）
```

2. レベル上昇

進行中状態を持つイベントにおいて、低いレベルのイベントが進行中の状態で高いレベルの条件を満たすと、両方のレベルが進行中状態になります。

例

```
設定:  
- Warning: CPU > 70%  
- Critical: CPU > 90%  
  
シナリオ:
```

- 1) CPU 80% → Warning 発生 (進行中)
- 2) CPU 95% → Warning (進行中)、Critical (進行中)

3. レベル下降

進行中状態を持つイベントで、高いレベルから低いレベルへ遷移した場合、高いレベルは解消され、低いレベルが維持されます。

例

設定:

- Warning: CPU > 70%
- Critical: CPU > 90%

シナリオ:

- 1) CPU 97% → Critical 発生
- 2) CPU 85% → Critical 解消、Warning 維持
- 3) CPU 65% → Warning 解消

4. 動作フロー

- Warning: CPU > 70%
- Critical: CPU > 90%

メトリクス値: 60% → 75% → 92% → 85% → 60%

状態: 正常 Warning Critical Warning 正常

 (発生) (発生) (維持) (解消)

 Warning

 (維持)

① 進行中状態を持つイベントは、条件が満たされている間は継続して有効となり、条件が解除されるまで維持されます。

条件設定ガイド

❗ メトリクス名が数字で始まる場合、または特殊文字を含む場合は、`${メトリクス名}` の形式で記述する必要があります。

- 括弧 `()`, `[]`
- 演算子 `+`, `-`, `*`, `/`, `%`
- 区切り文字 `:`, `@`, `#`, `,`, `,`, (空白)
- その他の特殊文字 `!`, `^`, `&`, `|`, `~`, ````, `=`

`${}` 使用例:

```
${cpu(xos)} > 50
${mem[0]} >= 100
${cpu-usage} > 80
${namespace:cpu} > 70
${metric name} > 60
${4xx_error} > 10
```

`${}` なしで使用可能: 英字、アンダースコア (`_`)、ドット (`.`) のみを含む場合

```
cpu > 80
cpu_usage > 50
CPUUtilization.Average > 0.8
```

1. 比較演算子

数値比較

```
より大きい:  cpu > 80
以上:        cpu >= 80
より小さい:  memory < 1000
以下:        memory <= 1000
等しい:      status == 200
等しくない:  error != 0
```

文字列比較

```
status == 'OK'  
region == "us-east-1"
```

2. 算術演算子

- 基本的な算術演算

```
加算:   cpu + 10 >= 90  
減算:   memory - 100 >= 500  
乗算:   cpu * 2 >= 100  
除算:   disk / 1024 >= 100  
剰余:   value % 10 == 0
```

- 括弧による優先順位の制御

```
(cpu + memory) * 2 >= 200  
(disk - used) / total >= 0.2  
((cpu + memory) / 2) >= 50
```

- 負の値の表現

```
cpu > -100
```

3. 論理演算子

- AND 演算 (&&)

```
cpu > 80 && memory > 1000  
${cpu(xos)} > 50 && ${mem(xos)} > 60
```

- OR 演算 (||)

```
cpu > 90 || memory > 90  
disk < 10 || network > 1000
```

- 複合論理演算

```
(cpu > 50 && memory > 50) || disk < 20
cpu > 80 && (memory > 1000 || disk > 500)
```

4. パターンマッチング演算子

- LIKE 演算子

```
oname like 'prod-*'
url like '*error*'
message like 'WARN%'
```

- NOT LIKE 演算子

```
oname not like 'test-*'
url not like '*debug*'
```

5. 組み込み関数

- Null チェック関数

```
isNull(value)      # null かどうかを確認
isNotNull(value)   # null でないかを確認
nvl(value, 0)      # null の場合に既定値を返す
isEmpty(str)       # 空文字列かどうかを確認
isNotEmpty(str)    # 空文字列でないかを確認
```

- 集計関数

```
sum(cpu, memory, disk)    # 合計
avg(cpu, memory)          # 平均
max(cpu, memory, disk)    # 最大値
min(cpu, memory, disk)    # 最小値
count(value1, value2, value3) # 件数
```

- 数学関数

```
round(cpu, 2)    # 四捨五入（小数点以下2桁）
```

- 文字列関数

```
length(str)      # 文字列の長さ
startsWith(str, 'prefix') # 接頭辞の確認
endsWith(str, 'suffix') # 接尾辞の確認
indexOf(str, 'search') # 文字列の位置
substring(str, 0, 10) # 部分文字列
trim(str)        # 空白の削除
replace(str, 'old', 'new') # 文字列の置換
hasStr(str, 'search') # 文字列の包含判定
```

- 条件関数

- **if:** 条件式が `true` の場合は `true値` を返し、`false` の場合は `false値` を返す条件関数

形式

```
if(条件, true値, false値)
```

例

```
if(value >= 90, 'High', 'Medium') == 'High'
```

- **decode:** 値と条件を比較し、一致した条件の結果を返します。一致しない場合は既定値を返します

形式

```
decode(値, 条件1, 結果1, 条件2, 結果2, ..., 条件N, 結果N, 既定値)
```

例

```
decode(value, 1, 'Low', 2, 'Medium', 3, 'High', 'Unknown') == 'Unknown'
```

- **in:** 値が指定した候補値のいずれかと一致した場合は `true` 、それ以外は `false` を返します

形式

```
in(値, 候補値1, 候補値2, ..., 候補値N)
```

例

```
in(status, 200, 201, 204) == false
```

❗ 記述時の注意事項

- 推奨事項
 - 簡単な変数名は `${}` を使わずに記述（例: `cpu`, `memory`）
 - 特殊文字を含む場合は必ず `${}` を使用
 - 複雑な演算は括弧で明確に区切る
 - 関数名は正確に入力（大文字と小文字を区別）
- 避けるべき事項
 - 不完全な式の記述（例: `cpu >` , `memory &&` ）
 - 括弧の不一致（例: `(cpu > 80 && cpu > 70)` ）
 - 演算子の連続使用(例: `cpu >> 80` , `cpu >< 80`)

複合メトリクス

ホーム > プロジェクト選択 >  アラート通知 > イベント設定 > 複合指標 タブ

複雑なルールを利用してイベントを作成し、警告通知を送信できます。

基本オプション

- **検索:** イベント名またはイベント受信タグを選択して、目的のイベントを検索します
- **JSON ダウンロード／アップロード:** JSON エディタでイベントルールを直接編集するか、ファイルとしてダウンロードして編集後に再アップロードできます
- **通知メッセージ設定:** 受信するメッセージの詳細項目を選択し、項目ごとに内容を編集します
- **+ イベント追加:** 新しい複合メトリクスイベントを追加します

 権限

複合指標 イベントを設定するには **アラート設定** 権限が必要です。この権限は **ホーム > プロジェクト選択 > 管理 > 統合メンバー管理** で、対象メンバーの  をクリックして付与できます。

表 | 複合メトリクスイベント一覧の構成

項目	説明
イベントタイトル	ユーザーが指定したイベント名 - 最大 255 バイトまで入力可能
条件	イベント発生条件
イベント状態が解消された場合の追加通知	Critical および Warning レベルのイベントが解消された際に、RECOVERED（正常）状態の通知を受信するかどうか - 有効: 進行中 → 正常状態に切り替わると RECOVERED 通知を送信し、イベント履歴で進行中／解消状態を区別して確認できます - 無効: しきい値を超えるたびに単発のイベントが発生します

項目	説明
イベント受信タグ	該当イベントの通知を受信するユーザーまたはグループ
	イベントの修正および削除
	イベントの有効／無効状態 - 有効: 条件を満たすとイベントが発生 - 無効: イベントは動作せず、条件も評価されません

イベント設定

複合指標 イベントは、メトリクスデータのクエリ言語である **MXQL** を基にイベント条件を作成します。**チャートから作成** 機能では、MXQL の自動補完を支援するコンボボックス機能を提供します。

イベント追加

1.  **アラート通知 > イベント設定** から **複合指標** タブをクリックします。
2. 画面右側の **[+ イベント追加]** ボタンをクリックします。
3. **複合指標** 画面で **チャートから作成** をクリックします。
 - a. 提供されているテンプレートを利用できます。
4. **イベント設定** 画面で、先にイベントデータを取得し、その後イベント条件を設定します。

イベントデータ照会

イベントデータを照会してチャートを構成します。**イベント設定** 画面右側の **イベントデータ照会** で、**ウィジェット** または **テキスト** オプションを選択してイベントを設定します。

• ウィジェット テキスト

時系列チャートを構成するオプションを使用して、イベント設定に利用する **MXQL** を自動補完できます。

1. データを照会する日付を選択します。
2. データを照会する **カテゴリ**（対象）を選択します。

3. **フィルター** 項目の **+フィルター追加** をクリックしてイベント条件を作成します。
 - a. フィルター画面で、演算式、タグ、フィルター値を選択します。
 - b. **+追加** をクリックして条件を追加できます。
 - c. 条件を削除する場合は、条件の横にある **-** をクリックします。
4. **グループ化** 項目で、グループ化されたメトリクスデータを選択します。複数選択が可能です。
5. **タイムユニット** 項目で、**秒**、**分**、**時間** 単位のいずれかを選択し、データを分割する時間基準を設定します。
6. **フィールド** 項目で、イベント発生条件に使用するフィールドを選択します。複数選択が可能です。

 テキストモードで入力した MXQL は、ウィジェットモードと互換性がありません。

MXQL を直接編集できるエディタが表示されます。

1. MXQL を直接修正し、**照会** ボタンをクリックします。
2. 日付を選択すると、該当内容をチャートまたはグラフで確認できます。

通知

警告通知設定の基本情報を入力します。

- **イベント有効化**: トグルボタンを有効にすると、イベントが動作します。
- **レベル**: イベントレベル (Critical、Warning、Info) を選択します。
 - **イベント状態が解消された場合の追加通知**: トグルを有効にすると、イベントが解消され、正常 (RECOVERED) 状態の通知が送信されます。
- **タイトル**: イベントルールのタイトルを入力します。
- **メッセージ**: イベント発生時に受信する通知メッセージの内容を入力します。メッセージ内で変数を使用すると、イベント発生時点の実際の値を含めることができます。

Message example

 変数使用ルール



- 変数形式: \${メトリクス名}
- 設定した **カテゴリ**と同一**カテゴリ** 内のメトリクスのみ変数として使用できます。
- 複数の変数を組み合わせて、詳細なメッセージを構成できます。
- 使用可能な変数一覧は **メトリクス照会** で確認できます。

イベント発行条件

- **データ照会範囲**: リアルタイムデータの照会範囲を選択します。
 - 選択可能な範囲: 30秒, 1分, 2分, 5分, 10分, 30分, 1時間, 2時間, 3時間
- **条件**: イベント発行条件を入力します。イベントデータ照会に含まれるフィールドのみ使用できます。

付加情報

- **インターバル**: 選択した時間間隔（1分, 5分, 10分, 30分, 1時間, 12時間）ごとにアラート条件を確認します。
- **サイレンス**: イベント発生後、選択した時間（unset, 1分, 5分, 10分, 30分, 1時間, 12時間）の間、同一イベントは発生しません。
- **イベント動作時間**: 勤務時間、非勤務時間、点検時間など、特定の時間帯にイベントが動作するようタグを設定します。タグを設定しない場合、イベントは有効な状態で24時間常時動作します。
 1. **+追加** をクリックします。
 2. **イベント動作時間** で **+新しいタグ作成** をクリックします。
 3. **イベント動作時間タグ作成** で **タグ名**、**曜日**、**時間**、**色** を選択し、**[適用]** ボタンをクリックします。
 4. 作成したタグはタグ一覧で確認でき、チェックボックスを選択すると適用されます。
 5. 修正または削除するタグの をクリックし、**イベント動作時間タグ修正** で編集または削除できます。



タグを削除すると、そのタグが適用されているすべてのユーザーから削除されます。ただし、イベントルールで使用中のタグは削除できません。

- **イベント受信タグ**: 受信タグを設定すると、該当タグを持つプロジェクトメンバーに通知が送信されます。受信タグを設定しない場合、プロジェクト内のすべてのユーザーに通知が送信されます。

1. **+ タグ追加** または **+** をクリックします。
2. **イベント受信タグ** 画面下部の **+ 新しいタグ作成** をクリックします。
3. **タグ作成** 画面でタグ名を入力し、色を選択した後、**[タグ作成]** ボタンをクリックします。
4. 作成したタグは、**タグ一覧** の  をクリックして修正または削除できます。
5. **イベント受信タグ** 画面の **タグ一覧** から適用したいタグを選択します。

✔ イベント設定時にイベント受信タグを選択すると、該当タグを持つプロジェクトメンバーおよび 3rd-party プラグインに通知を送信できます。**アラート通知 > 通知設定** で、それぞれにタグを割り当てることができます。

イベントルールテスト

設定した条件を過去データに基づいてテストし、どの程度のイベントが発生するかを予測できます。

1. イベントルールをテストする日付を選択し、**[確認]** ボタンをクリックします。
2. **[実行]** ボタンをクリックして、イベント発生件数を確認します。

イベント修正／削除

1. **アラート通知 > 通知設定** の **複合指標** タブに移動します。
2. 複合メトリクス一覧で、修正または削除したいイベントの  をクリックします。
3. **通知設定** 画面でオプションを修正し、**[保存]** ボタンをクリックします。
 - a. 選択したイベントを削除する場合は、**通知設定** 画面右上の **[削除]** ボタンをクリックします。

イベント共有

メトリクスイベント設定を JSON ファイルとして保存し、他のユーザーと設定を共有したり、他のユーザーの設定を取り込んだりすることができます。

- JSON ファイル名: **event-rules- YYYYMMDD .json**

エクスポート

1. 画面右上の [JSON ] ボタンをクリックします。
2. JSON 編集画面が表示されたら、[ エクスポート] ボタンをクリックします。
 - イベントを検索した状態でエクスポート機能を使用すると、検索結果のみが JSON ファイルとしてダウンロードされます。
3. JSON ファイルがダウンロードされたら、共有したい他のユーザーに渡してください。

インポート

1. 画面右上の [] ボタンをクリックします。
2. エクスポート 機能でダウンロードした JSON ファイルを選択します。
3. JSON 編集画面が表示されたら、[一覧に追加] または [上書き] を選択します。

 同じ種類の製品間での利用を推奨します。異なる製品のプロジェクトからイベント設定を取り込むことは可能ですが、正常に動作しない場合があります。

JSON 形式で編集する

1. 画面右上の [JSON ] ボタンをクリックします。
2. 編集画面が表示されたら、JSON 形式に従って内容を修正してください。
3. 修正が完了したら、画面下部の [保存] ボタンをクリックします。

 修正した内容が JSON 形式に合っていない場合、画面下部にエラーメッセージが表示され、保存できません。表示されるエラーメッセージは、形式エラーの内容によって異なる場合があります。

JSON error

通知メッセージ設定

受信するメッセージの詳細項目を選択し、項目ごとに内容を編集できます。

付録

MXQL 文法ガイド

複合指標 は MXQL クエリを基に動作します。詳細な文法については、[MXQL 文法ガイド](#) を参照してください。

異常値

ホーム > プロジェクト選択 >  **アラート通知** > **イベント設定** > **異常検知** タブ

異常検知 は、メトリクスデータから異常な上昇または下降パターンを検出し、通常とは異なる動きが検知された場合にイベントを発生させる機能です。

基本オプション

- **検索:** イベント名またはイベント受信タグを選択して、目的のイベントを検索します。
- **+ イベントルール追加:** 新しい異常値イベントを追加します。

表 | 異常値イベント一覧の構成

項目	説明
イベントメッセージ	イベント発生時に送信されるメッセージ
カテゴリ	メトリクスデータを分類する単位で、メトリクスイベント設定の基準となるカテゴリ
フィールド	異常を検知する指標（メトリクスフィールド）
フィルター	異常値の対象となるデータ範囲を制限する条件
上昇／下降パターン	上昇または下降パターンの感度
イベント受信タグ	該当イベントの通知を受信するユーザーまたはグループ
	イベント設定の修正
	イベントの削除
	イベントの有効／無効状態 - 有効: 条件を満たすとイベントが発生 - 無効: イベントは動作せず、条件も評価されません

イベント設定

メトリクスデータを基に、値の動きが通常とは異なり上昇または下降した場合に、警告通知を送信するよう設定できます。

イベントルール追加

1. 画面右側の [+ イベントルール追加] ボタンをクリックします。
2. イベント基本情報, メトリクス, 異常値, 付加情報 を順に設定します。

イベント基本情報

1. イベントレベル (Critical, Warning, Info) を選択します。
 - イベントの重要度および通知レベルが決定されます。
2. イベント発生時に受信する通知メッセージを入力します。
 - メッセージ内で変数を使用すると、イベント発生時点の実際の値を含めることができます。

❗ 使用可能な変数

- \$pcode : プロジェクトコード
- \$category : カテゴリ
- \$field : フィールド
- \$value : 発生値
- \$oname : イベント対象

メトリクス

1. カテゴリ を選択します。
 - 異常を検知するメトリクスデータの基準となるカテゴリです。

2. イベント発生条件に使用する **フィールド** を選択します。複数選択が可能です。
3. イベント対象を絞り込むためのフィルターを選択します。複数選択が可能です。

異常値

1. **上昇パターン** と **下降パターン** の感度を選択します。
2. 各パターンのトグルボタンをクリックして、有効または無効を切り替えます。
 - 上昇パターンのみを検知する場合: 上昇パターン **ON**, 下降パターン **OFF**
 - 下降パターンのみを検知する場合: 下降パターン **ON**, 上昇パターン **OFF**

付加情報

イベント動作時間

勤務時間、非勤務時間、点検時間など、特定の時間帯にイベントが動作するようタグを設定します。タグを設定しない場合、イベントは有効な状態で 24 時間常時動作します。

1. **+ 追加** をクリックします。
2. **イベント動作時間** で **+ 新しいタグ作成** をクリックします。
3. **イベント動作時間タグ作成** で **タグ名**、**曜日**、**時間**、**色** を選択し、**[適用]** ボタンをクリックします。
4. 作成したタグはタグ一覧で確認でき、チェックボックスを選択すると適用されます。
5. 修正または削除するタグの  をクリックし、**イベント動作時間タグ修正** で編集または削除できます。

 タグを削除すると、そのタグが適用されているすべてのユーザーから削除されます。ただし、イベントルールで使用中のタグは削除できません。

イベント受信タグ

イベント受信タグを設定すると、該当タグを持つプロジェクトメンバーに通知が送信されます。受信タグを設定しない場合、プロジェクト内のすべてのユーザーに通知が送信されます。

1. **+ タグ追加** または **+** をクリックします。
2. **イベント受信タグ** 画面下部の **+ 新しいタグ作成** をクリックします。

3. タグ作成 画面でタグ名を入力し、色を選択した後、[タグ作成] ボタンをクリックします。
4. 作成したタグは、タグ一覧の  をクリックして修正または削除できます。
5. イベント受信タグ 画面の タグ一覧 から適用したいタグを選択します。

✔ イベント設定時にイベント受信タグを選択すると、該当タグを持つプロジェクトメンバーおよび 3rd-party プラグインに通知を送信できます。アラート通知 > 通知設定 で、それぞれにタグを割り当てることができます。

イベント修正

1. アラート通知 > イベント設定 の 異常検知 タブに移動します。
2. 異常値 一覧で、修正するイベントの右側にある  をクリックします。
3. 異常値 画面でオプションを修正し、[保存] ボタンをクリックします。

イベント削除

1. アラート通知 > イベント設定 の 異常検知 タブに移動します。
2. 異常値 一覧で、削除するイベントの右側にある  をクリックします。
3. 削除確認画面で [削除] ボタンをクリックします。

リアルタイムログイベント

ホーム > プロジェクト選択 >  アラート通知 > イベント設定 > リアルタイムログ タブ

収集したログデータを条件に合わせてフィルタリングして警告通知を設定できます。ログイベントを使用するには、ログモニタリングを有効化する必要があります。**ログモニタリングの有効化**については、各製品のログドキュメントを参照してください。

- ・ **リアルタイムログイベント**: リアルタイムで収集したログで、設定したイベント条件が満たされるとイベントが発生します。

基本オプション

- ・ **検索**: イベント名またはイベント受信タグを選択して目的のイベントを検索
- ・ **JSONダウンロード/アップロード**: イベントルールはJSONエディタで直接編集するか、ファイルにダウンロードして修正後に再アップロード可能
- ・ **通知メッセージ設定**: 受信するメッセージの詳細項目を選択し、項目ごとに内容を編集
- ・ **+ イベント追加**: 新しいログイベントを追加

表 | リアルタイムログイベント一覧構成

項目	説明
イベント名	通知タイトルとして使用されるイベント名 - 最大255バイトまで入力可能
カテゴリ	ログ分類名（ログフォルダ名）
検索キー	ログデータで検索したい項目 - 例: HTTPレスポンスステータスコードの値にアクセスする場合、検索キー <code>status</code>
検索値	検索キーに対応する実際のデータ。ログに入力した文字列が含まれる場合に警告通知を送信 - 例: 検索キー <code>status</code> 、検索値 <code>200</code> を設定した場合、HTTPステータスコード200を含むログデータ収集時に警告通知が発生

項目	説明
イベント発生一時停止	イベント通知発生後、同じイベントの発生を一定時間停止 - 選択可能時間: 使用しない、5分、10分、15分、20分、30分、1時間、2時間、3時間、6時間、12時間、1日 - 解消通知が有効な場合、正常（RECOVERED）状態の通知受信後、選択した時間中は同じイベントが発生しても通知されない
イベント対象フィルタリング	イベントが適用される対象の条件 - 対象条件を設定しない場合、収集されるすべてのデータに対して通知を判断 - 選択入力: タグ、演算子、値を選択して条件を構成 - 直接入力: 条件を直接入力して対象を指定
イベント受信タグ	イベント通知を受け取るユーザーまたはグループ（受信タグ）
	イベントの編集と削除
	イベントの有効化状態 - 有効: 条件を満たすとイベント発生 - 無効: イベントが動作せず、条件をチェックしない

イベント設定

ログイベントは、リアルタイムで収集されるログ、または特定条件を満たすログが一定数蓄積された際に警告通知を設定できる機能です。

リアルタイムログイベントの追加

1. アラート通知 > イベント設定のリアルタイムログタブをクリックしてください。
2. 画面右側の [+ イベント追加] ボタンをクリックしてください。
3. イベントルール追加ウィンドウでログイベント条件を設定してください。
 - イベント名: イベントタイトルとして使用する名前を入力してください。

- **イベント有効化:** トグルボタンを有効にすると、条件を満たしたときにイベントが発生します。
- **レベル:** イベントレベル (**Critical**、**Warning**、Info) を選択してください。
- **メッセージ:** イベント発生時に受け取る通知メッセージを入力してください。メッセージに変数を使用して、イベント発生時点の実際の値を含めることができます。

✔ 変数使用ルール

- 変数形式: \${指標名}
- 設定したカテゴリと同じカテゴリ内の指標のみ変数として使用できます。
- 複数の変数を組み合わせて詳細なメッセージを構成できます。
- 使用可能な変数一覧は**メトリクス照会**で確認できます。

- **カテゴリ:** ログ分類名（ログフォルダ名）をリストから選択するか、直接入力してください。カテゴリを選択してから検索キーと検索値を入力できます。
- **検索キー:** ログデータで検索したい項目を入力してください。
 - 例: HTTPレスポンスステータスコードの値にアクセスする場合、**検索キー** status
- **検索値:** 検索キーに対応する実際のデータを入力してください。ログに入力した文字列が含まれる場合に警告通知を送信します。
 - 例: **検索キー** status、**検索値** 200 を設定した場合、HTTPステータスコード200を含むログデータ収集時に警告通知が発生
- **イベント対象フィルタリング:** イベントが適用される対象の条件を入力してください。入力値がない場合、収集されるすべてのデータに対して通知を判断します。
 - **選択入力:** タグ、演算子、値を選択して対象条件を設定してください。

❗ 条件式の記述方法とサポートされる演算子については、[条件設定ガイド](#)を参照してください。

- **直接入力:** 条件を直接入力してください。
- **イベント発生一時停止:** 通知を一時停止する時間を選択してください。イベント発生後、設定した時間中は同じイベントが発生しません。

- 選択可能時間: 使用しない、1分、2分、3分、5分、10分、20分、30分、1時間、3時間、6時間、12時間
- イベント動作時間: 特定の時間帯（業務/非業務/メンテナンス時間など）にイベントが動作するようにタグを設定してください。タグを設定しない場合、イベントが有効な状態で24時間常時動作します。
 1. + 追加をクリックしてください。
 2. イベント動作時間で**+新しいタグを作成**をクリックしてください。
 3. イベント動作時間タグの作成でタグ名、曜日、時間、色を選択し、[適用] ボタンをクリックしてください。
 4. 作成したタグはタグリストで確認でき、チェックボックスを選択すると適用されます。
 5. 変更または削除するタグの  アイコンをクリックして、イベント動作時間タグの編集でタグを編集または削除できます。

 タグを削除すると、そのタグが適用されているすべてのユーザーのタグが削除されます。ただし、イベントルールで使用中のタグは削除できません。

- イベント受信タグ: イベントに受信タグを設定すると、そのタグを持つプロジェクトメンバーに通知が送信されます。受信タグを設定しない場合、プロジェクト内のすべてのユーザーに通知が送信されます。
 1. + タグ追加または+をクリックしてください。
 2. イベント受信タグウィンドウ下部の**+ 新しいタグを作成**をクリックしてください。
 3. タグ作成ウィンドウでタグ名を入力し、色を選択してから、[タグ作成] ボタンをクリックしてタグを作成してください。
 4. 作成されたタグはタグリストの  アイコンをクリックして編集または削除してください。
 5. イベント受信タグウィンドウのタグリストから目的のタグを選択すると適用されます。

 イベント設定時にイベント受信タグを選択して、そのタグを持つプロジェクトメンバーとサードパーティプラグインに通知を送信できます。アラート通知 > 通知設定でプロジェクトメンバーとサードパーティプラグインにそれぞれタグを指定できます。

4. イベントルール追加ウィンドウで設定が完了したら、[保存] ボタンをクリックしてください。

イベントの編集/削除

1. アラート通知 > イベント設定のリアルタイムログタブに移動してください。
2. ログイベントリストで編集または削除するイベントの  アイコンをクリックしてください。
3. イベント設定画面でオプションを変更し、[保存] ボタンをクリックしてください。
 - a. 選択したイベントを削除するには、イベント設定画面の右上の [削除] ボタンをクリックしてください。

イベントの共有

メトリクスイベント設定を JSON ファイルとして保存し、他のユーザーと設定を共有したり、他のユーザーの設定を取り込んだりすることができます。

- JSON ファイル名: `event-rules-YYYYMMDD.json`

エクスポート

1. 画面右上の [JSON ] ボタンをクリックします。
2. JSON 編集画面が表示されたら、[ エクスポート] ボタンをクリックします。
 - イベントを検索した状態でエクスポート機能を使用すると、検索結果のみが JSON ファイルとしてダウンロードされます。
3. JSON ファイルがダウンロードされたら、共有したい他のユーザーに渡してください。

インポート

1. 画面右上の [] ボタンをクリックします。
2. エクスポート 機能でダウンロードした JSON ファイルを選択します。
3. JSON 編集画面が表示されたら、[一覧に追加] または [上書き] を選択します。

 同じ種類の製品間での利用を推奨します。異なる製品のプロジェクトからイベント設定を取り込むことは可能ですが、正常に動作しない場合があります。

JSON 形式で編集する

1. 画面右上の [JSON 

❗ 修正した内容が JSON 形式に合っていない場合、画面下部にエラーメッセージが表示され、保存できません。表示されるエラーメッセージは、形式エラーの内容によって異なる場合があります。

JSON error

付録

発生条件・対象選択ガイド

メトリクスアラートにおけるイベントの発生条件とイベント対象の選択は、同一の構文を使用します。ただし、イベント発生条件ではフィールド（Field）のキーを変数として使用し、イベント対象の選択ではタグ（Tag）のキーを変数として使用します。

レベル動作ガイド

イベントレベルは、危険（Critical）、警告（Warning）、正常（Info）の3段階に分類されます。

1. 優先順位に基づくイベント発生

複数のレベル条件を同時に満たした場合、最も優先度の高いイベントのみが発生します。

例

設定:

- Warning: CPU > 70%

- Critical: CPU > 90%

現在の状態: CPU 95%
→ 結果: Critical イベントのみ発生 (Warning は抑制される)

2. レベル上昇

進行中状態を持つイベントにおいて、低いレベルのイベントが進行中の状態で高いレベルの条件を満たすと、両方のレベルが進行中状態になります。

例

設定:
- Warning: CPU > 70%
- Critical: CPU > 90%

シナリオ:
1) CPU 80% → Warning 発生 (進行中)
2) CPU 95% → Warning (進行中)、Critical (進行中)

3. レベル下降

進行中状態を持つイベントで、高いレベルから低いレベルへ遷移した場合、高いレベルは解消され、低いレベルが維持されます。

例

設定:
- Warning: CPU > 70%
- Critical: CPU > 90%

シナリオ:
1) CPU 97% → Critical 発生
2) CPU 85% → Critical 解消、Warning 維持
3) CPU 65% → Warning 解消

4. 動作フロー

- Warning: CPU > 70%
- Critical: CPU > 90%

```

メトリクス値: 60% → 75% → 92% → 85% → 60%
状態:   正常   Warning   Critical   Warning   正常
        (発生)   (発生)   (維持)   (解消)
           Warning
           (維持)

```

- ❗ 進行中状態を持つイベントは、条件が満たされている間は継続して有効となり、条件が解除されるまで維持されます。

条件設定ガイド

- ❗ メトリクス名が数字で始まる場合、または特殊文字を含む場合は、`${メトリクス名}` の形式で記述する必要があります。

- 括弧 `()`, `[]`
- 演算子 `+`, `-`, `*`, `/`, `%`
- 区切り文字 `:`, `@`, `#`, `,`, ``` (空白)
- その他の特殊文字 `!`, `^`, `&`, `|`, `~`, ````, `=`

`${}` 使用例:

```

${cpu(xos)} > 50
${mem[0]} >= 100
${cpu-usage} > 80
${namespace:cpu} > 70
${metric name} > 60
${4xx_error} > 10

```

`${}` なしで使用可能: 英字、アンダースコア (`_`)、ドット (`.`) のみを含む場合

```

cpu > 80
cpu_usage > 50
CPUUtilization.Average > 0.8

```

1. 比較演算子

数値比較

```
より大きい:  cpu > 80
以上:        cpu >= 80
より小さい:  memory < 1000
以下:        memory <= 1000
等しい:      status == 200
等しくない:  error != 0
```

文字列比較

```
status == 'OK'
region == "us-east-1"
```

2. 算術演算子

- 基本的な算術演算

```
加算:  cpu + 10 >= 90
減算:  memory - 100 >= 500
乗算:  cpu * 2 >= 100
除算:  disk / 1024 >= 100
剰余:  value % 10 == 0
```

- 括弧による優先順位の制御

```
(cpu + memory) * 2 >= 200
(disk - used) / total >= 0.2
((cpu + memory) / 2) >= 50
```

- 負の値の表現

```
cpu > -100
```

3. 論理演算子

- AND 演算 (&&)

```
cpu > 80 && memory > 1000
${cpu(xos)} > 50 && ${mem(xos)} > 60
```

- OR 演算 (||)

```
cpu > 90 || memory > 90
disk < 10 || network > 1000
```

- 複合論理演算

```
(cpu > 50 && memory > 50) || disk < 20
cpu > 80 && (memory > 1000 || disk > 500)
```

4. パターンマッチング演算子

- LIKE 演算子

```
oname like 'prod-*'
url like '*error*'
message like 'WARN%'
```

- NOT LIKE 演算子

```
oname not like 'test-*'
url not like '*debug*'
```

5. 組み込み関数

- Null チェック関数

```
isNull(value)      # null かどうかを確認
isNotNull(value)   # null でないかを確認
nvl(value, 0)      # null の場合に既定値を返す
isEmpty(str)       # 空文字列かどうかを確認
isNotEmpty(str)    # 空文字列でないかを確認
```

- 集計関数

```
sum(cpu, memory, disk)    # 合計
avg(cpu, memory)          # 平均
max(cpu, memory, disk)    # 最大値
min(cpu, memory, disk)    # 最小値
count(value1, value2, value3) # 件数
```

- 数学関数

```
round(cpu, 2)    # 四捨五入（小数点以下2桁）
```

- 文字列関数

```
length(str)      # 文字列の長さ
startsWith(str, 'prefix') # 接頭辞の確認
endsWith(str, 'suffix') # 接尾辞の確認
indexOf(str, 'search') # 文字列の位置
substring(str, 0, 10) # 部分文字列
trim(str)        # 空白の削除
replace(str, 'old', 'new') # 文字列の置換
hasStr(str, 'search') # 文字列の包含判定
```

- 条件関数

- **if:** 条件式が `true` の場合は `true値` を返し、`false` の場合は `false値` を返す条件関数

形式

```
if(条件, true値, false値)
```

例

```
if(value >= 90, 'High', 'Medium') == 'High'
```

- **decode:** 値と条件を比較し、一致した条件の結果を返します。一致しない場合は既定値を返します

形式

```
decode(値, 条件1, 結果1, 条件2, 結果2, ..., 条件N, 結果N, 既定値)
```

例

```
decode(value, 1, 'Low', 2, 'Medium', 3, 'High', 'Unknown') == 'Unknown'
```

- **in**: 値が指定した候補値のいずれかと一致した場合は `true`、それ以外は `false` を返します

形式

```
in(値, 候補値1, 候補値2, ..., 候補値N)
```

例

```
in(status, 200, 201, 204) == false
```

❗ 記述時の注意事項

- 推奨事項
 - 簡単な変数名は `${}` を使わずに記述（例: `cpu`, `memory`）
 - 特殊文字を含む場合は必ず `${}` を使用
 - 複雑な演算は括弧で明確に区切る
 - 関数名は正確に入力（大文字と小文字を区別）
- 避けるべき事項
 - 不完全な式の記述（例: `cpu >` , `memory &&` ）
 - 括弧の不一致（例: `(cpu > 80 && cpu > 70)` ）
 - 演算子の連続使用(例: `cpu >> 80` , `cpu >< 80`)

複合ログイベント設定

ホーム > プロジェクト選択 >  アラート通知 > イベント設定 > 複合ログ タブ

収集したログデータを条件に合わせてフィルタリングして警告通知を設定できます。ログイベントを使用するには、ログモニタリングを有効化する必要があります。**ログモニタリングの有効化**については、各製品のログドキュメントを参照してください。

- **複合ログイベント**: 設定したデータ照会範囲内のログを基準に、ログ検索条件を満たすログ数が閾値以上の場合にイベントが発生します。

! 権限

複合メトリクスイベントを設定するには**通知設定権限**が必要です。通知設定権限は、**ホーム画面 > プロジェクト選択 > 管理 > 統合メンバー管理**で、権限を付与するメンバーの  アイコンをクリックして設定できます。

基本オプション

- **検索**: イベント名またはイベント受信タグを選択して目的のイベントを検索
- **JSONダウンロード/アップロード**: イベントルールはJSONエディタで直接編集するか、ファイルにダウンロードして修正後に再アップロード可能
- **通知メッセージ設定**: 受信するメッセージの詳細項目を選択し、項目ごとに内容を編集
- **+ イベント追加**: 新しいログイベントを追加

表 | 複合ログイベント一覧構成

項目	説明
イベント名	通知タイトルとして使用されるイベント名
テンプレート	複合ログテンプレート

項目	説明
カテゴリ	ログ分類名（ログフォルダ名）
ルール	イベントが発生する条件
イベント状態が解消されたら追加通知	CriticalとWarningレベルのイベントが解消されたとき、正常（RECOVERED）状態の通知を受信するかどうか - 有効: 進行中 → 正常状態に移行したときRECOVERED通知を送信し、イベント履歴で進行中/解消状態を区別して確認可能 - 無効: 閾値を超えるたびに単発のイベントが発生
イベント発生一時停止	イベント通知発生後、同じイベントの発生を一定時間停止 - 選択可能時間: 使用しない、5分、10分、15分、20分、30分、1時間、2時間、3時間、6時間、12時間、1日 - 解消通知が有効な場合、正常（RECOVERED）状態の通知受信後、選択した時間中は同じイベントが発生しても通知されない
インターバル	ログデータを照会する時間範囲
イベント受信タグ	イベント通知を受け取るユーザーまたはグループ（受信タグ）
	イベントの編集と削除
	イベントの有効化状態 - 有効: 条件を満たすとイベント発生 - 無効: イベントが動作せず、条件をチェックしない

イベント設定

設定したデータ照会範囲内のログを基準に、ログ検索条件を満たすログ数が閾値以上の場合に警告通知を設定できる機能です。

複合ログイベントの追加

1. アラート通知 > イベント設定の複合ログ タブをクリックしてください。
2. 画面右側の [+ イベント追加] ボタンをクリックしてください。
3. イベントルール追加ウィンドウでログイベント条件を設定してください。詳細については、[基本設定](#)、[詳細設定](#)、[追加設定](#)を参照してください。

複合ログイベントの追加

4. イベントルール追加ウィンドウで設定が完了したら、[保存] ボタンをクリックしてください。

基本設定

- イベント名: イベントタイトルとして使用する名前を入力してください。
- イベント有効化: トグルボタンを有効にすると、条件を満たしたときにイベントが発生します。
- レベル: イベントレベル ([Critical](#)、[Warning](#)、[Info](#)) を選択してください。

詳細設定

[テンプレート \(プリセット\)](#)、[メッセージ](#)、[カテゴリ](#)、[フィルター](#)、[イベント発行条件](#)を設定します。

テンプレート (プリセット)

定義済みのプリセットを選択すると、[フィルター](#)、[クエリタイプ](#)、[イベント発行条件](#)が自動的に設定されます。プリセットを選択後、必要に応じて変更できます。

複合ログテンプレートの種類



プリセット	クエリタイプ	デフォルトルール
使用しない	none	rows > 10
2xxステータスコード件数	count	count > 10



プリセット	クエリタイプ	デフォルトルール
3xxステータスコード件数	count	count > 10
4xxステータスコード件数	count	count > 10
5xxステータスコード件数	count	count > 10
正常ステータスコード (2xx, 3xx) 件数	count_v2	include_minus_exclude_count > 10
エラーステータスコード (4xx, 5xx) 件数	count_v2	include_minus_exclude_count > 10
エラー受信件数	count	count > 10
例外受信件数	count	count > 10
毎秒ログ発生件数	rps	rps > 100
フィールド集計 (Aggregation)	aggr	avg > 1000
比率 (Rate)	rate	rate > 10
ユニーク値数 (Cardinality)	cardinality	cardinality > 100
パーセンタイル (Percentile)	percentile	p99 > 3000

メッセージ

イベント発生時に受け取る通知メッセージを入力してください。メッセージに変数を使用して動的なメッセージを構成できます。

- 例示: メッセージ `${host}` でエラー `${count}` 件発生 → 通知メッセージ web-01 でエラー15件発生

表 | メッセージ変数

変数	説明	使用条件
<code>\${count}</code>	条件に一致するログ件数	常に使用可能
<code>\${groupField}</code>	グループ化フィールド名	グループ化通知のみ
<code>\${groupValue}</code>	グループ化フィールド値	グループ化通知のみ
<code>\${フィールド名}</code>	グループ化フィールド値（フィールド名で直接参照）	グループ化通知のみ

❗ `${count}` 変数はグループ化の有無に関わらず、すべての通知で使用できます。 `${groupField}`、`${groupValue}`、`${フィールド名}` 変数はグループ化フィールドが設定されている場合のみ置換されます。

カテゴリ

ログ分類名（ログフォルダ名）をリストから選択するか、直接入力してください。カテゴリを選択してから検索キーと検索値を入力できます。

フィルター

Luceneクエリ構文でログ検索条件を入力します。2種類の入力モードをサポートします。

- フィルターモード [▼](#): フィールドと値を選択してクエリを構成します。
- コードモード [<>](#): Luceneクエリを直接入力します。

表 | Luceneクエリ例

クエリ	説明
<code>level:ERROR</code>	levelフィールドがERRORのログ
<code>status_nxx:status_4xx OR status_nxx:status_5xx</code>	4xxまたは5xxステータスコード
<code>appender:accessLog</code>	accessLog appenderのログ
<code>response_time.n:*</code>	response_time.nフィールドが存在するログ

クエリ	説明
host:web-01 AND level:ERROR	web-01ホストのエラーログ

イベント発行条件

クエリタイプ別に設定します。選択したテンプレート（プリセット）またはクエリタイプに応じて、追加設定項目が表示されます。

表 | イベント発行条件キーの意味

キー	意味
rows / count	フィルター条件に一致するログ件数
rps	毎秒ログ発生件数（件数 / インターバル秒）
avg	対象フィールド値の平均
sum	対象フィールド値の合計
min	対象フィールド値の最小値
max	対象フィールド値の最大値
rate	分子件数 / 分母件数 × 100（%）
cardinality	対象フィールドのユニーク値数
p99 / p95 / p90 / p75 / p50	対象フィールドのパーセンタイル値
include_minus_exclude_count	含むフィルター件数 - 除外フィルター件数

- **件数（count / none）**：設定したフィルター条件に一致するログの件数をカウントします。

表 | 件数設定項目

設定項目	説明
フィルター	ログ検索条件（Luceneクエリ）
イベント発行条件	件数 または 条件に一致するログ {演算子} {閾値}（例: count > 10）

- **毎秒ログ発生件数（rps）**：設定したフィルター条件に一致するログの毎秒発生件数（Requests Per Second）を計算します。インターバル時間中の総件数をインターバル秒数で割って算出します。

表 | 毎秒ログ発生件数設定項目

設定項目	説明
フィルター	ログ検索条件（Luceneクエリ）
イベント発行条件	RPS {演算子} {閾値}（例: rps > 0.5）

- **フィールド集計（aggr）**：指定した数値フィールドの集計値（平均、合計、最小値、最大値）を計算します。イベント発行条件で集計方式をドロップダウンから選択できます。

表 | フィールド集計設定項目

設定項目	説明
フィルター	ログ検索条件（Luceneクエリ）
対象フィールド	集計対象の数値フィールド（例: response_time.n）
イベント発行条件	平均 / 合計 / 最小値 / 最大値 {演算子} {閾値}（例: avg > 1000）

- **比率（rate）**：分子フィルターと分母フィルターの件数比率（%）を計算します。全ログに対する特定条件の比率を監視する際に有用です。

表 | 比率設定項目

設定項目	説明
フィルター（分子）	比率計算の分子となるログ条件（例: level:ERROR）
分母フィルター	比率計算の分母となるログ条件（例: level:*）
イベント発行条件	比率（%） {演算子} {閾値}（例: rate > 10）

- 例示: フィルター level:ERROR、分母フィルター level:*、ルール rate > 10 → 全ログ中のエラー比率が10%を超えると通知が発生
- **ユニーク値数（cardinality）** : 指定したフィールドのユニーク値数を計算します。異常に多くの種類の値が発生する場合を検出する際に有用です。

表 | ユニーク値数設定項目

設定項目	説明
フィルター	ログ検索条件（Luceneクエリ）
対象フィールド	ユニーク値を計算するフィールド（例: host）
イベント発行条件	ユニーク値数 {演算子} {閾値}（例: cardinality > 100）

- **パーセンタイル（percentile）** : 指定した数値フィールドのパーセンタイル値を計算します。レスポンスタイムのテールレイテンシを監視する際に有用です。パーセンタイルの選択を変更すると、イベント発行条件のキーが自動的に更新されます。

表 | パーセンタイル設定項目

設定項目	説明
フィルター	ログ検索条件（Luceneクエリ）
対象フィールド	パーセンタイルを計算する数値フィールド（例: response_time.n）
パーセンタイル	P99、P95、P90、P75、P50から選択

設定項目	説明
イベント発行条件	P99 / P95 / ... {演算子} {閾値} (例: p99 > 3000)

表 | パーセンタイルの意味

パーセンタイル	意味
P50	中央値。全データの50%がこの値以下
P75	全データの75%がこの値以下
P90	全データの90%がこの値以下
P95	全データの95%がこの値以下
P99	全データの99%がこの値以下

- 含む/除外件数 (count_v2) : 含むフィルターの件数から除外フィルターの件数を引いた純件数を計算します。

表 | 含む/除外件数設定項目

設定項目	説明
フィルター (含む)	含むログ条件 (例: status_nxx:status_4xx OR status_nxx:status_5xx)
フィルター除外	除外するログ条件 (例: status_nxx:status_4xx)
分母フィルター	比率計算用の分母 (任意)
イベント発行条件	含む-除外件数 {演算子} {閾値} (例: include_minus_exclude_count > 10)

- 例示: フィルター status_nxx:status_4xx OR status_nxx:status_5xx、フィルター除外 status_nxx:status_4xx
→ 4xx+5xxから4xxを除いた純5xx件数が10を超えると通知が発生

グループ化フィールド（Group By）

グループ化フィールドを設定すると、そのフィールドの値ごとに独立して通知条件が判定されます。ドロップダウンからグループ化するフィールドを選択します。ドロップダウンをクリックすると、現在のカテゴリで使用可能なフィールド一覧がオートコンプリートで表示されます。

- 例示: フィルター `level:ERROR`、グループ化フィールド `host`、ルール `rows > 3`

表 | グループ化設定の比較

項目	グループ化なし	グループ化あり
判定単位	全ログを1つに集計	グループ値ごとに独立して集計
通知発生	1件	条件を満たすグループ数分
通知タイトル	イベントタイトル	[グループ値] イベントタイトル
Stateful	全体基準での状態遷移	グループごとに独立した状態遷移

- `[web-01] エラー通知` — web-01でエラー8件 → 通知発生
- `[web-02] エラー通知` — web-02でエラー5件 → 通知発生
- `[api-01] エラー通知` — api-01でエラー2件 → 条件未達、通知なし

❗ イベント状態が解消されたら追加通知を有効にした場合、グループごとに独立してABNORMAL → RECOVERED状態が遷移します。例えば web-01 のエラーが解消されると `[web-01] RECOVERED 通知` が発生し、他のホストの状態には影響しません。

追加設定

- **インターバル:** 選択した時間（`unset`、`1分`、`5分`、`10分`、`30分`、`1時間`、`12時間`）間隔で通知条件を確認します。
- **無音:** イベント発生後、選択した時間（`unset`、`1分`、`5分`、`10分`、`30分`、`1時間`、`12時間`）中は同じイベントが発生しません。

- **イベント動作時間:** 特定の時間帯（業務/非業務/メンテナンス時間など）にイベントが動作するようにタグを設定してください。タグを設定しない場合、イベントが有効な状態で24時間常時動作します。
 1. **+追加**をクリックしてください。
 2. **イベント動作時間で**+新しいタグを作成****をクリックしてください。
 3. **イベント動作時間タグの作成でタグ名、曜日、時間、色**を選択し、[適用] ボタンをクリックしてください。
 4. 作成したタグはタグリストで確認でき、チェックボックスを選択すると適用されます。
 5. 変更または削除するタグの  アイコンをクリックして、**イベント動作時間タグの編集**でタグを編集または削除できます。

 タグを削除すると、そのタグが適用されているすべてのユーザーのタグが削除されます。ただし、イベントルールで使用中のタグは削除できません。

- **イベント受信タグ:** イベントに受信タグを設定すると、そのタグを持つプロジェクトメンバーに通知が送信されます。受信タグを設定しない場合、プロジェクト内のすべてのユーザーに通知が送信されます。
 1. **+タグ追加**または**+**をクリックしてください。
 2. **イベント受信タグウィンドウ下部の**+新しいタグを作成****をクリックしてください。
 3. **タグ作成ウィンドウ**でタグ名を入力し、色を選択してから、[タグ作成] ボタンをクリックしてタグを作成してください。
 4. 作成されたタグはタグリストの  アイコンをクリックして編集または削除してください。
 5. **イベント受信タグウィンドウのタグリスト**から目的のタグを選択すると適用されます。

 イベント設定時にイベント受信タグを選択して、そのタグを持つプロジェクトメンバーとサードパーティプラグインに通知を送信できます。**アラート通知 > 通知設定**でプロジェクトメンバーとサードパーティプラグインにそれぞれタグを指定できます。

シミュレーション

通知を保存する前に、過去のデータでシミュレーションして通知発生予測件数を確認できます。

1. 通知設定を完了後、下部の時間範囲を選択してください。
2. [シミュレーション] ボタンをクリックしてください。
3. チャートに時間帯別データと通知発生回数が表示されます。

❗ シミュレーション時間範囲を調整して、さまざまな時間帯の通知発生パターンを確認できます。直近1~2分のデータはインデックスビルド遅延により照会されない場合があります。

4. イベントルール追加ウィンドウで設定が完了したら、[保存] ボタンをクリックしてください。

ログエクスプローラーから通知を作成

ログエクスプローラーから直接通知を作成できます。

1. ログ>ログ検索メニューに移動してください。
2. カテゴリと検索クエリを設定してください。
3. 画面右上の[イベント追加] ボタンをクリックしてください。
4. Drawerで通知設定フォームが開き、現在のカテゴリと検索クエリが自動的に入力されます。
5. 追加設定を完了し、[保存] ボタンをクリックしてください。
6. 保存完了後、[移動] ボタンをクリックすると通知一覧に移動します。

イベントの編集/削除

1. アラート通知>イベント設定の複合ログタブに移動してください。
2. ログイベントリストで編集または削除するイベントの✎アイコンをクリックしてください。
3. イベント設定画面でオプションを変更し、[保存] ボタンをクリックしてください。
 - a. 選択したイベントを削除するには、イベント設定画面の右上の[削除] ボタンをクリックしてください。

イベントの共有

メトリクスイベント設定を JSON ファイルとして保存し、他のユーザーと設定を共有したり、他のユーザーの設定を取り込んだりすることができます。

- JSON ファイル名: `event-rules-YYYYMMDD.json`

エクスポート

- 画面右上の [JSON ] ボタンをクリックします。
- JSON 編集画面が表示されたら、[ エクスポート] ボタンをクリックします。
 - イベントを検索した状態でエクスポート機能を使用すると、検索結果のみが JSON ファイルとしてダウンロードされます。
- JSON ファイルがダウンロードされたら、共有したい他のユーザーに渡してください。

インポート

- 画面右上の [] ボタンをクリックします。
- エクスポート 機能でダウンロードした JSON ファイルを選択します。
- JSON 編集画面が表示されたら、[一覧に追加] または [上書き] を選択します。

 同じ種類の製品間での利用を推奨します。異なる製品のプロジェクトからイベント設定を取り込むことは可能ですが、正常に動作しない場合があります。

JSON 形式で編集する

- 画面右上の [JSON ] ボタンをクリックします。
- 編集画面が表示されたら、JSON 形式に従って内容を修正してください。
- 修正が完了したら、画面下部の [保存] ボタンをクリックします。

❗ 修正した内容が JSON 形式に合っていない場合、画面下部にエラーメッセージが表示され、保存できません。表示されるエラーメッセージは、形式エラーの内容によって異なる場合があります。

JSON error

データベース

ホーム > プロジェクト選択 >  **アラート通知** > イベント設定 classic

データベースの指標ごとにイベント条件を設定し、アラート通知を送信します。指標ごとにしきい値を入力することで、**危険** および **警告** レベルに分けて通知できます。

基本オプション

データベースインスタンスの主要指標を基準にイベント条件を設定できます。

- ・ **インスタンス**: イベント条件を設定するデータベースエージェント
- ・ **データベース**: モニタリング対象のデータベース
- ・ **エージェント非アクティブ警告**: 設定した時間後に応答がない場合、エージェントから警告通知を送信

表 | データベースイベント設定画面の構成

項目	説明
メトリクス	イベント条件を設定するデータベース指標
危険	Critical レベルのしきい値
警告	Warning レベルのしきい値
検知回数	指標が 5 秒ごとに収集される際、しきい値を連続して超過した回数 - 例: 3 回に設定した場合、5 秒 × 3 回連続 条件で通知
状態解消時の追加通知	イベント発生後、正常状態に復旧したことを知らせる追加通知 - DBX エージェント 1.6.30 以上対応
	イベント削除
	イベントの有効 / 無効 - 有効: 条件を満たすとイベント発生

項目	説明
	- 無効: イベントは動作せず条件も評価されません

イベント追加

選択した **メトリクス** 項目に基づき、入力した **危険** および **警告** 条件が **検知回数** 分だけ発生すると、アラート通知を送信するよう設定できます。

1. アラート通知 > イベント設定 **classic** に移動します。
2. インスタンス でアラートを設定するエージェントを選択します。
3. [+ イベント追加] ボタンをクリックします。
4. **メトリクス** から目的の指標を選択します。
5. **危険** と **警告** の比較演算子 (**<** , **≤** , **>** , **≥**) を選択し、数値を入力します。
 - 設定値を超過するとアラートが発生します。
6. **検知回数** に希望する数値を入力します。
 - 5 秒間隔で設定回数連続発生すると通知されます。
 - 例: 検知回数が 3 の場合
5 秒 (1 回目) - 5 秒 (2 回目) - 5 秒 (3 回目) → 合計 15 秒 → 通知
7. **状態解消時の追加通知** を有効にすると、イベントが解消 (RECOVERED) された際に追加通知が送信されます。

❗ 状態解消時の追加通知 機能は DBX エージェント バージョン 1.6.30 以上 で対応しています。
8. 右端のトグルボタンを有効 () にするとアラート通知が送信されます。
 - a. イベントを削除する場合は、該当イベントの  をクリックしてください。

アラート複製

現在選択しているインスタンスに作成したアラートイベントを、他のインスタンスに複製できます。

1. アラート通知 > イベント設定 **classic** に移動します。
2. [アラート複製] ボタンをクリックします。
3. アラート複製 画面で複製先インスタンスを選択します。すべてのインスタンスに複製する場合は **全体選択** を選択してください。
4. [複製] ボタンをクリックしてアラートを複製します。

❗ 複製先インスタンスに既存のアラートイベントがある状態で **アラート複製** 機能を使用すると、既存のアラートイベントは削除されます。

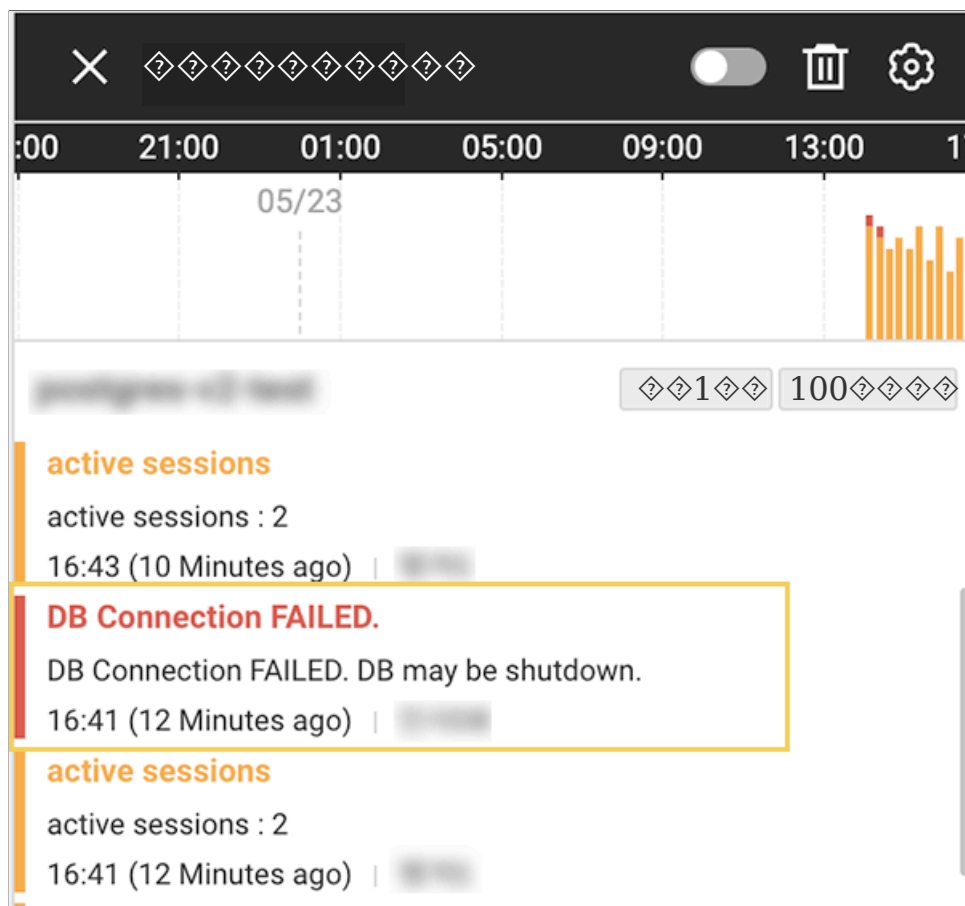
エージェント非アクティブ警告

設定した時間後に応答がない場合、エージェントから警告通知を送信します。

1. アラート通知 > イベント設定 **classic** に移動します。
2. エージェント非アクティブ警告 で希望する時間（ **30秒** , **1分** , **2分** , **3分** , **5分** , **10分** , **30分** ）を選択します。
3. 右側のトグルボタンをクリックして有効化します。
 -  **有効**: 設定した時間中にエージェントからデータを受信できない場合、**INACTIVE MONITORING AGENT** イベントが発生します。

基本アラート: Connection FAILED

イベントルールを別途設定しなくても、エージェントが DB 接続に 15 回以上失敗した場合、アラートが送信されます。DB シャットダウンやネットワーク障害、アカウントまたは権限の問題が原因となる場合があります。



通知設定

ホーム > プロジェクト選択 >  アラート通知 > 通知設定

ユーザー別アラート受信設定

 **ユーザー別アラート受信設定** における他メンバーの情報は、**メンバー管理** 権限を持つメンバーのみが確認できます。それ以外のメンバーには、必須情報がマスキングされて表示されます。

- イベント通知の一括受信設定およびアクセス設定のためのモバイル端末管理は、**アカウント情報** で行うことができます。
- メンバー権限の詳細については、[こちらのドキュメント](#) を参照してください。

通知手段の設定

メール、**SMS**、**WhatsApp**、**モバイル** による通知を提供します。

ユーザー別アラート受信設定 セクションで、受信したい通知手段のチェックボックスを選択すると警告アラートを受信できます。通知手段のチェックボックスを解除すると、その手段ではアラートは送信されません。

 プロジェクトの最高管理者を除き、すべてのユーザーは自分自身の受信設定のみ変更できます。

メール通知

メール通知は、会員登録時に入力したメールアドレス宛に送信されます。

SMS通知

SMSで通知を受信できます。

 SMS通知機能は、有料プロジェクトでのみ利用できます。

1. 画面右上の **プロフィールアイコン** をクリックし、**アカウント管理** に移動します。
2. **アカウント情報** の **ユーザー電話番号** セクションで、[一般携帯電話] ボタンをクリックします。
3. 認証番号を受信する **電話番号** を入力します。
 - SMS通知を受信できる電話番号は、**韓国の携帯電話番号のみ**対応しています。
4. [保存] ボタンをクリックします。

✔ 登録した電話番号の変更および削除

- 電話番号を変更する場合は、[変更] ボタンをクリックしてください。SMS認証が必要です。
- 電話番号を削除する場合は、[削除] ボタンをクリックしてください。削除後は、SMSによる通知を受信できなくなります。

WhatsApp通知

WhatsAppを通じて通知を受信できます。

ⓘ WhatsApp通知機能は、有料プロジェクトでのみ利用できます。

1. 画面右上の **プロフィールアイコン** をクリックし、**アカウント管理** に移動します。
2. **アカウント情報** の **ユーザー電話番号** セクションで、[WhatsApp] ボタンをクリックします。
3. 認証番号を受信する **電話番号** を入力します。
4. [認証番号送信] ボタンをクリックします。
5. WhatsAppアプリケーションに送信された6桁の認証番号を入力します。
 - 10分以内に認証番号を受信できなかった場合は、[認証番号再送信] ボタンをクリックしてください。
6. [認証] ボタンをクリックします。

モバイル通知

モバイル通知を受信するには、モバイル端末にWhaTapアプリケーションをインストールし、ログインする必要があります。

- WhaTapアプリケーションをダウンロード – [Android](#)
- WhaTapアプリケーションをダウンロード – [iOS](#)

繰り返し通知（エスカレーション）設定

アラート発生後も状況が解消されない場合、初回アラート発生時刻を基準に通知を繰り返す間隔を設定できます。この設定は **Critical** レベルのアラートにのみ適用されます。

例えば、初回アラート発生から即時（0分）、1時間後、1日後に通知を繰り返す場合は、**繰り返し通知（エスカレーション）** カラムに **0,1H,1D** と入力します。

1. **アラート通知 > 通知設定** に移動し、**ユーザー別アラート受信設定** セクションを開きます。
2. **繰り返し通知（エスカレーション）** 項目で設定します。
 - **M**: 分、**H**: 時間、**D**: 日。単位を省略した場合は分単位として扱われます。
 - 数値、または 数値+単位（**M, H, D**）の形式で入力してください。入力形式が正しくない場合はメッセージが表示されます。
3. [保存] ボタンをクリックしてください。保存しない場合、設定は反映されません。

イベント受信タグ

イベント設定時に **イベント受信タグ** を選択すると、そのタグを持つプロジェクトメンバーや 3rd-party プラグインに通知を送信できます。イベント受信タグを設定しない場合、**すべてのプロジェクトメンバー** に警告アラートが送信されます。

イベント受信タグの追加・編集・削除

 イベントに **適用中のイベント受信タグ** は削除できません。

1. **アラート通知 > 通知設定** に移動します。
2. **ユーザー別アラート受信設定** セクションのユーザー一覧で、**+ タグ追加** または **+** をクリックします。
3. **イベント受信タグ** 画面で **+ 新しいタグ作成** をクリックします。
4. **タグ作成** 画面でタグ名を入力し、色を選択して [タグ作成] ボタンをクリックします。

5. 作成されたタグは、タグ一覧の  をクリックして編集または削除できます。
6. イベント受信タグ画面の **タグ一覧** から使用したいタグを選択すると適用されます。

イベント受信タグを解除する

1. アラート通知 > 通知設定 に移動し、**ユーザー別アラート受信設定** セクションを開きます。
2. イベント受信タグを解除したいユーザーの **イベント受信タグ** 項目にある + をクリックします。
3. イベント受信タグ画面で、解除したいタグの X をクリックすると、そのタグが解除されます。

受信タグ未設定のアラートを受信する

イベント設定時にイベント受信タグを選択すると、そのタグを持つプロジェクトメンバーや 3rd-party プラグインに通知を送信できます。

- イベント受信タグが設定されていないアラートを受信するには、**受信タグ未設定のアラートを受信** オプションを有効にしてください。
- イベント受信タグが設定されているアラートのみ受信したい場合は、このオプションを無効にしてください。

❗ すべての警告アラートを受信しない場合は、このオプションを無効にし、**イベント受信タグ** が一つも選択されていない状態にしてください。

イベント受信モードの選択

ユーザー別アラート受信設定 セクションで、メンバーまたはサードパーティチャネルごとにイベント通知の受信範囲を次の3つのモードから選択できます。受信モードは、各メンバーまたはチャネル行の**イベント受信タグ**列で選択します。

Mode	Description
すべての通知を受信	イベント受信タグに関係なく、すべての警告通知を受信
選択したタグのみ通知を受信	選択したイベント受信タグを持つ警告通知のみ受信。タグを最低1つ以上選択
すべての通知を受信しない	いかなるイベント通知も受信しない。選択したタグは削除されずそのまま保存

- 選択したタグのみ通知を受信モードでは、タグ選択領域で通知を受信するタグを1つ以上選択してください。
- イベント受信タグが指定されていない警告通知まで受信するには、タグ選択領域で**未分類仮想タグ**を選択してください。**未分類タグ**は通常のタグと同様に選択または解除でき、タグ自体を編集・削除することはできません。

❗ すべての通知を受信しないモードでは、該当メンバーまたはチャネルがイベント通知をまったく受信しません。このとき選択したタグは削除されず保存されるため、再び**選択したタグのみ通知を受信モード**に切り替えると、以前に選択したタグ設定をそのまま使用できます。

通知受信言語の選択

ユーザーが受信するアラート通知の言語を選択できます。

1. アラート通知 > 通知設定 に移動します。
2. ユーザー別アラート受信設定 セクションのユーザー一覧で、言語を設定したいユーザーの一番左にある ► ボタンをクリックします。
3. 通知受信言語 を選択します。

曜日・時間別通知設定

ユーザーごとに、曜日および時間帯別の通知受信可否を設定できます。

1. アラート通知 > 通知設定 に移動します。
2. ユーザー別アラート受信設定 セクションのユーザー一覧で、曜日・時間別通知を設定したいユーザーの一番左にある ► ボタンをクリックします。
3. 警告アラートを受信したい曜日を選択するか、時間帯を入力します。
 - 通知手段ごとに設定できます。

サードパーティプラグイン

外部アプリケーションを通じて警告アラートを受信できます。

- 対応アプリケーション: Slack、Telegram、Teams、Teams Workflows、Jandi、AlertNow、Webhook、Webhook

JSON、Opsgenie、PagerDuty、LINE、ilert

❗ WhaTap のサポート範囲に含まれない社内メッセンジャーは、標準の **Webhook** または **Webhook JSON** を使用して連携できます。

表 | サードパーティプラグイン一覧

項目	説明
プラグイン名	プラグインの名称
認証キー	認証キー
認証値	認証値
受信レベル	警告、全体、危険のいずれか
繰り返し通知（エスカレーション）	問題が解消されない場合、初回通知時刻から再通知する間隔 - Critical レベルのアラートにのみ適用
Smart メッセージ（AI）	Telegram プラグイン追加時に AI 通知機能を提供 - 対応言語：韓国語、英語、日本語
イベント受信タグ	プロジェクト内の特定メンバーまたはチームに対する通知受信設定
削除	サードパーティプラグインの削除

サードパーティプラグインの追加

1. アラート通知 > 通知設定 に移動します。
2. サードパーティプラグイン セクションで **追加** ボタンをクリックします。
3. サードパーティプラグイン追加 画面で、使用するサービスのタブを選択します。
4. 選択したサービスの案内（[サードパーティプラグイン通知の追加方法](#)）に従って設定します。
5. 受信テスト ボタンをクリックして設定を確認します。

- すべて完了したら **追加** または **登録** ボタンをクリックします。

サードパーティプラグイン通知の追加方法

Slack

- サードパーティプラグイン追加 画面の **Slack** タブで、[**Slackに登録**] ボタンをクリックすると、Slack のチームおよびチャンネルに **WhaTapNotiBot** を登録できます。

Telegram

- Telegramで @WhaTapApmNotiBot を検索してください。

- ボットを選択し、[開始] ボタンをクリックします。
- チャット画面に `/license` コマンドと、通知を受信したいプロジェクトのアクセスキーを入力してください。



① 使用可能なコマンド一覧

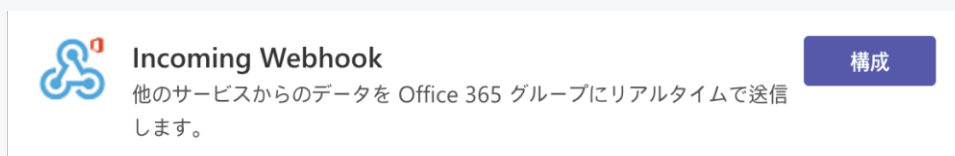
- `/help` : コマンド一覧を表示
- `/license` : 通知を受信するプロジェクトのアクセスキーを入力
- `/remove` : 通知を受信しないプロジェクトコードを入力
- `/list` : 現在通知を受信しているプロジェクトコードを表示

Teams

- 通知を受信したいチャンネルの **コネクタ** に移動します。



2. Webhook を追加し、[構成] ボタンをクリックします。



3. Webhook 名を入力し、[作成] ボタンをクリックします。



Incoming Webhook

フィードバックの送信

Incoming Webhook コネクタを利用すると、トラッキング対象の活動についての通知を外部サービスから受け取ることができます。このコネクタを使うには、別サービスで設定を作成する必要があります。そのサービスが、Office 365 コネクタ形式 対応の Web フックをサポートしている必要があります。

* が付いているフィールドは必須です

着信 Web フック (Incoming Webhook) をセットアップするには、名前を入力して [作成] を選んでください。*

この着信 Web フック (Incoming Webhook) からのデータに関連付ける画像をカスタマイズします。

イメージをアップロードします。



既定のイメージ

作成

キャンセル

4. 生成された Webhook URL をコピーします。

以下の URL をコピーしてクリップボードに保存してから、[保存] を選んでください。この URL は、グループへのデータ送信元となるサービスの画面で必要になります。

`https://whatap.webhook.office.com/webf`



URL は最新です。

完了

削除

5. WhaTap モニタリングサービスの サードパーティプラグインの追加 画面に戻り、Teams タブ下部の Channel Name と Webhook URL に値を入力します。

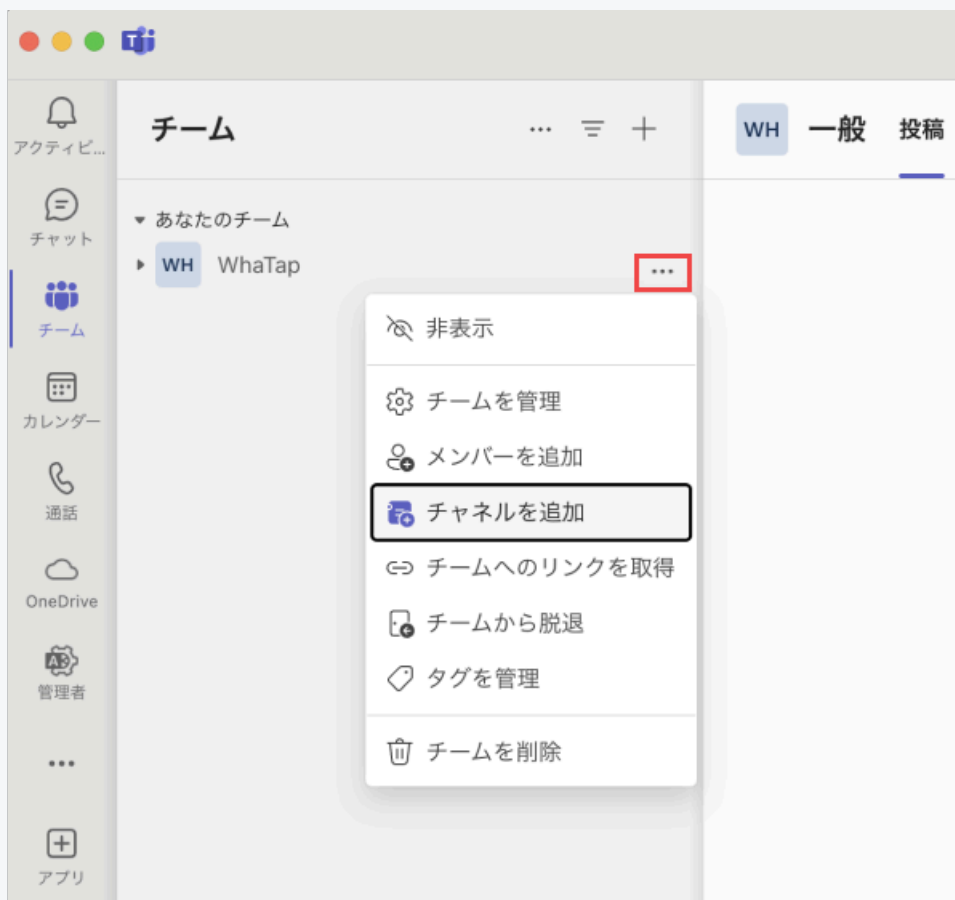
6. [登録] ボタンをクリックします。



Teams Workflows

- ① ワークフローを使用するには、**Microsoft Teams** にアクセス可能なアカウントと **ビジネスプランのサブスクリプション**が必要です。

1. チーム メニューの **その他の操作 (...)** > **チャンネルを追加** をクリックします。



2. チャンネルの作成画面で **チャンネル名** と **説明** を入力します。チャンネルの種類は **標準** を選択し、[作成] ボタンをクリックします。



チャンネルの作成

チャンネル名 *

WhaTapのイベントチャンネル

説明

他のユーザーが、適切なチャンネルを見つけられるように説明を入力します

チャンネルの種類を選択する * ⓘ

選択

☒ 標準
チームの全員がアクセスできます。

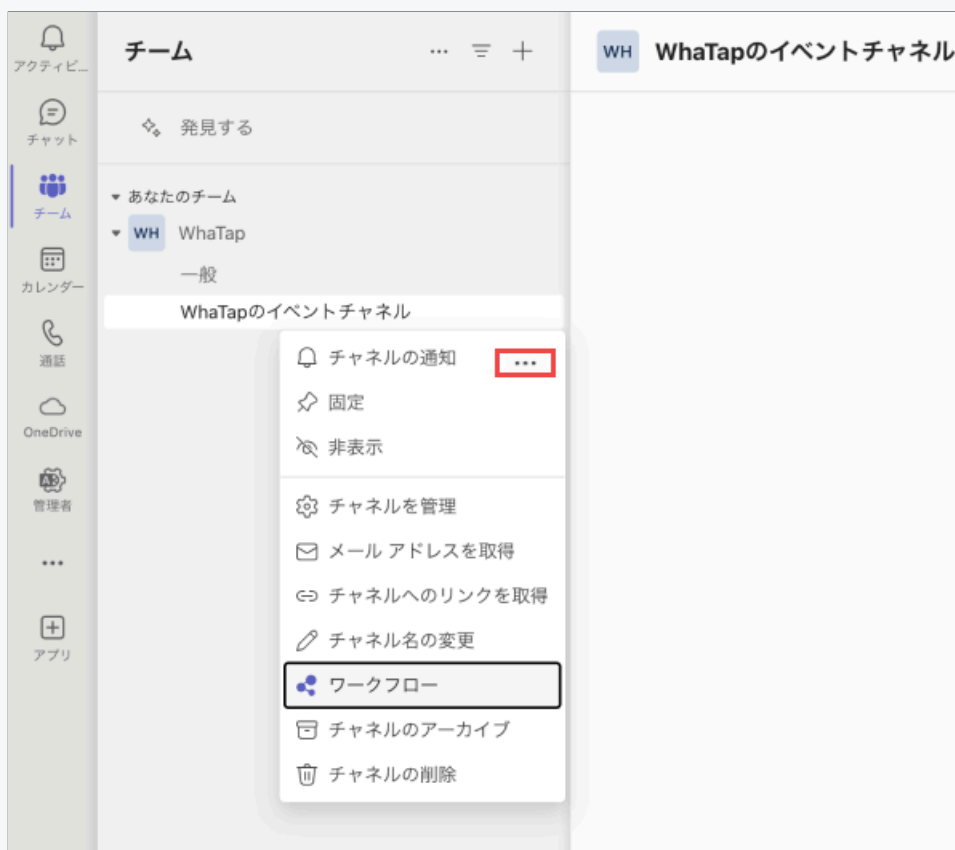
☐ 共有
組織内または組織外の人またはチームがアクセスできます。

☐ プライベート
チームの特定のユーザーがアクセスできます。

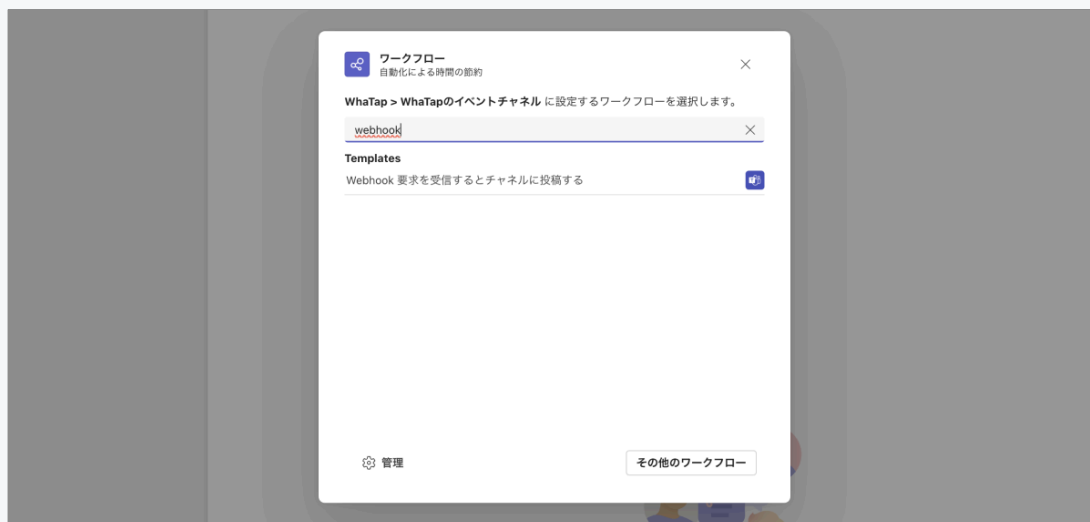
キャンセル 作成

❗ チャンネルの種類を **共有** にするとワークフローは使用できません。**プライベート** を選択した場合、ワークフローは追加できますが通知は受信できません。

- 作成したチャンネルの **その他の操作 (...)** > **ワークフロー** をクリックします。



4. ワークフロー ダイアログの一覧から Webhook 要求を受信したらチャンネルに投稿 を選択し、[ワークフローを追加] ボタンをクリックします。



5. 名前と接続の状態を確認し、[次へ] ボタンをクリックします。
6. 詳細を確認後、[ワークフローを追加] ボタンをクリックします。
7. ワークフローが追加されました。生成された URL をコピーします。



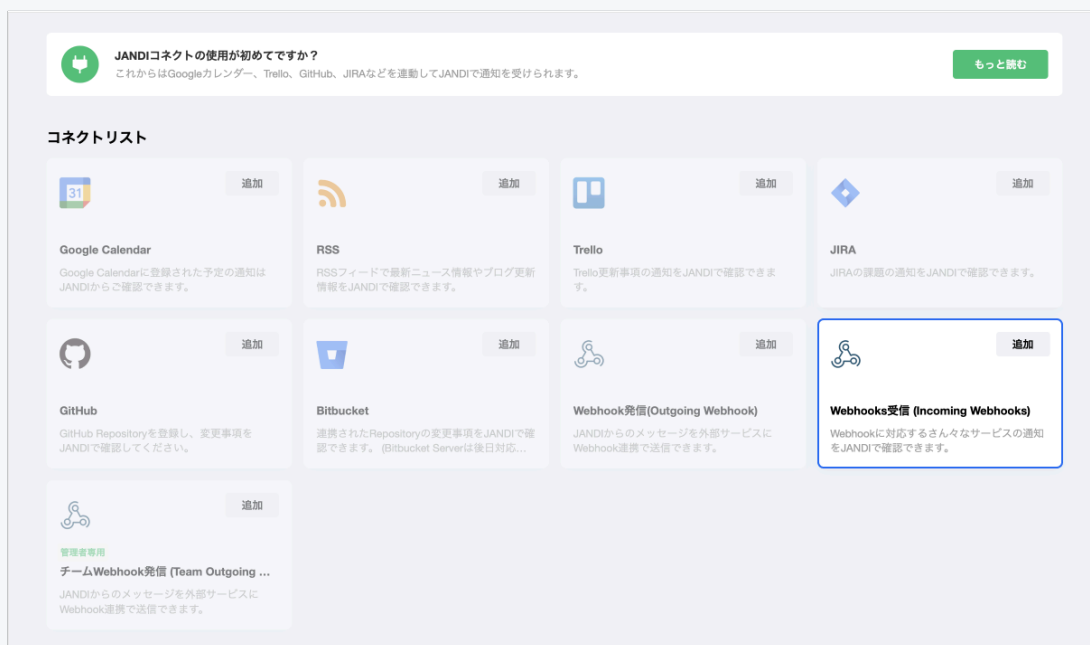
8. WhaTap モニタリングサービスの **サードパーティプラグインの追加** 画面に戻り、**Teams Workflows** タブ下部の **Channel Name** と **Webhook URL** に値を入力します。
9. [登録] ボタンをクリックします。

Jandi





1. 通知を受信するトピックまたはチャットで、**Jandi Connect 設定** に移動します。
2. **コネクト一覧** から **Webhook 受信 (Incoming Webhook)** を選択し、[連携する] ボタンをクリックします。



3. [**Webhook 受信を追加**] ボタンをクリックします。
4. **Webhook URL 登録** 項目に表示される Webhook URL をコピーします。



5. WhaTap モニタリングサービスの **サードパーティプラグインの追加** 画面に戻り、**Jandi** タブ下部の **Channel Name** と **Webhook URL** に値を入力します。
6. [**登録**] ボタンをクリックします。



1. AlertNow サービスにアクセスし、設定 > Integration メニューに移動します。



2. Integration 画面で [Integration 作成] ボタンをクリックします。
3. Integration タイプの中から **WhaTap** を選択します。
4. Integration 作成 画面で、名前 および 命名規則 を設定します。

5. [確認] ボタンをクリックすると、Integration が作成されます。
6. 作成された Integration の一覧を確認し、該当する Integration を選択します。



インテグレーション

インテグレーションの作成 検索語を入力 最後のアップデート: 2024-02-21 11:25:10

統合名	統合タイプ	統合キー	サービス名
ワットテスト	ワタップ		ワットテスト

7. 詳細画面で、URL 右側のコピーアイコンをクリックして URL をコピーします。

← インテグレーション

ワットテスト

基本情報

統合タイプ	ワタップ
統合キー	fda527a5140
URL	https://alertnowitgr.opsnow.com/integration/whatap/v1/
コンストラクタ	リサ

クリップボードにコピーする

8. Whatap モニタリングサービスの **サードパーティプラグインの追加** 画面に戻り、AlertNow タブ下部の **Channel Name** と **Webhook URL** に値を入力します。
9. [登録] ボタンをクリックします。

Webhook



1. 通知を受信する Webhook を登録します。登録した URL に対して、通知の **title** と **text** が POST メソッドで送信されます。

```
{
  "title": "Alert Title",
  "text": "Sample warning message"
}
```

2. **Webhook Name** と **Webhook URL** の項目を入力します。
3. [登録] ボタンをクリックします。

Webhook JSON



1. 通知を受信する Webhook を登録します。登録した URL に対して、通知データが **JSON 形式で POST** メソッドにより送信されます。JSON に含めることができる置換変数については、**使用可能な変数**を参照してください。
 - `${something}` は WhaTap が提供する置換値を表します。対応する全一覧と意味については、**使用可能な変数**を参照してください。
 - イベント状態（`status`）は、**イベント発生**（`on`）と**イベント解消**（`off`）に区別されます。
 - `level` は `None`、`Info`、`Warning`、`Critical` に区別されます。

```
{
  "pcode": "${pcode}",
  "projectName": "${projectName}",
  "time": "${time}",
  "oid": "${oid}",
  "oname": "${oname}",
  "title": "${title}",
  "message": "${message}",
  "level": "${level}",
  "status": "${status}",
  "metricName": "${metricName}",
  "metricValue": "${metricValue}",
  "metricThreshold": "${metricThreshold}",
  "uuid": "${uuid}",
  "okind": "${okind}",
  "onode": "${onode}"
}
```

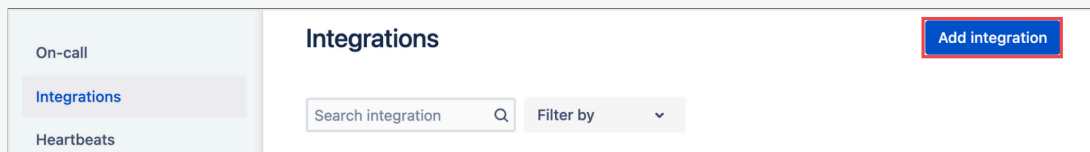
2. **Webhook Name** と **Webhook URL** の項目を入力してください。
3. ヘッダー値を追加するには、**ユーザーヘッダー値の追加** の + 追加 をクリックし、key と value を入力してください。
4. [登録] ボタンをクリックしてください。

Opsgenie

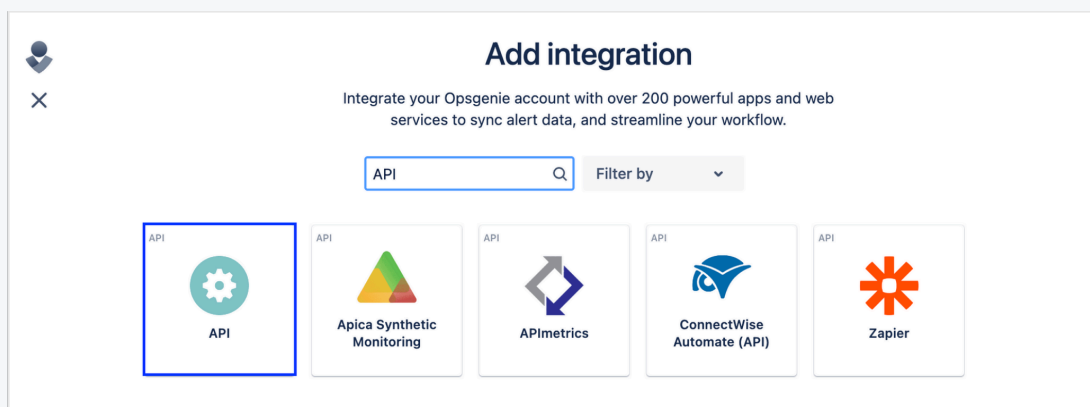




1. **Teams** メニューで、アラートを受信するチームを選択してください。
2. **Integration** メニューに移動し、[Add Integration] ボタンをクリックしてください。



3. インテグレーション一覧で **API** を検索し、選択してください。



4. API 名を設定し、チームを割り当て、各権限を設定した後、[Save Integration] ボタンをクリックしてください。



Forwarding rules
Your on-call schedule

ON-CALL
Global policies
All schedules
All escalations

CONNECTED APPS
Atlassian products

INTEGRATIONS
Integrations
Heartbeats
Slack
GitLab

Settings

Name:

Assigned to Team: ?

API Key: ?

Read Access: ?
☒

Create and Update Access: ?
☒

Delete Access: ?
☒

Restrict Configuration Access: ?
☐

Enabled: ?
☒

Suppress Notifications: ?
☐

[Save Integration](#) [Cancel](#)

5. Whatap モニタリングサービスの **サードパーティプラグインの追加** 画面に戻り、Opsgenie タブ下部の **Name** と **API Key** を入力してください。
6. [登録] ボタンをクリックしてください。



PagerDuty



1. [PagerDuty](#) アカウントを登録し、ログインしてください。
2. PagerDuty で **Services > Service Directory** メニューに移動し、[+ New Service] ボタンをクリックしてください。
3. サービス情報 (Name) を入力し、[Next] ボタンをクリックしてください。
4. **Escalation Policy** を設定し、[Next] ボタンをクリックしてください。



PagerDuty

5. **Alert Grouping** と **Transient Alerts** を設定し、[Next] ボタンをクリックしてください。

PagerDuty

6. **Integrations** で **Event API V2** を選択し、[Create Service] ボタンをクリックしてください。

PagerDuty

7. 作成されたサービス画面で **Integration Name** と **Integration Key** の値をコピーしてください。

PagerDuty

8. WhaTap モニタリングサービスの **サードパーティプラグインの追加** 画面に戻り、**Integration Name** と **Integration Key** を入力してください。
9. [登録] ボタンをクリックしてください。

❗ メッセージを受信できない、またはプラグインが登録されない場合は、**Integration Name** または **Integration Key** の値が正しいか確認してください。

LINE



1. LINE アプリをインストールしてください。
2. 以下の QR コードをスキャンして、**WhaTapLabs** 公式チャンネルを友だち追加してください。

LINE 3rd パーティプラグイン追加

3. 次のコマンドを使用して、プロジェクトを LINE と連携し、通知を設定してください。

コマンド

```
/help # 使用可能なコマンド一覧を表示
/list # 連携されているプロジェクト一覧を表示
/license [アクセスキー] # 新しいプロジェクト連携を追加
```



```
/remove [プロジェクトコード] # プロジェクト連携を解除
/language [ko|ja|en] # 通知言語を設定（韓国語／日本語／英語）
```

- プロジェクトのアクセスキーは **管理 > プロジェクト管理** で確認できます。
- LINE 内で `/language` コマンドを使用した場合、その設定は **該当する LINE チャットルーム内のみ** に適用されます。他の 3rd パーティプラグインやプロジェクトの通知言語は変更されません。

使用例

```
/license XXXXXXXXXX # プロジェクトを連携
/language ko # 言語を韓国語に設定
/list # 連携済みプロジェクトを確認
```



- **プロジェクトアクセスキー**: WhaTap プロジェクト内の **管理 > プロジェクト管理** で確認できます。
- **言語設定**: `/language` コマンドは LINE チャットルーム内でのみ有効で、他の通知チャンネルの言語は変更されません。

ilert



1. ilert コンソールの **Alert sources** 画面右上にある **[+ Create alert source]** をクリックします。
2. **Integration type** ステップ下部の **Filter integrations** 検索欄に **WhaTap** を入力して検索します。
3. 検索結果から **WhaTap** タイルを選択し、**[Next]** ボタンをクリックします。
4. アラートソース名を入力し、チームを指定した後、**[Next]** ボタンをクリックします。
 - エスカレーションポリシーは、新規作成または既存のものを選択できます。
5. アラートのグループ化方式を選択し、**[Continue setup]** ボタンをクリックします。
 - 最初に選択した **Do not group alerts** は後から変更可能です。
6. 次の画面で **アラートテンプレートや優先度などの追加設定** を行うことができます。ここでは [



Finish setup] ボタンをクリックして設定を完了します。

7. 設定が完了すると、完了画面で **API キー** または **Webhook URL** が生成されます。
8. **Webhook JSON** タブで生成された **Webhook URL** をコピーし、下部の **Webhook URL** 入力欄に貼り付けます。
9. [登録] ボタンをクリックします。

大量アラート発生の防止

短時間に大量のアラートが発生した場合、指定した時間だけアラート通知を一時的に停止します。たとえば、5 分間に 20 件のイベントが発生すると、その後 5 分間は警告アラートが送信されません。設定した **停止時間** が経過すると、大量アラート防止機能は自動的に解除されます。

1. 次の画面に移動してください。 **アラート通知 > 通知設定 > 大量アラート通知発生防止** セクション
2. トグルボタンをクリックして **有効化** または **無効化** します。
 -  **有効**: 大量アラートを検知した場合、一時的に通知を停止
 -  **無効**: アラート通知を再開
3. **検知時間**, **検知回数**, **停止時間** を設定してください。
 - **検知時間** 内に **検知回数** 以上のイベントが発生すると、**停止時間** の間、警告アラートが停止されます。



大量アラートによってブロックされた場合、ブロック案内ボックスが表示されます。大量アラート防止機能をすぐに解除するには、ボックス内の **大量アラート発生中止** ボタンをクリックしてください。

イベント通知フォーマット

共通アラートテンプレート

共通テンプレートは、ほとんどのケースで適用できます。さまざまな環境において、同一のフォーマットでアラートを受信できます。

- プロダクトタイプ: アプリケーション、データベース、Kubernetes
- イベントタイプ: アプリケーションアラート、データベースアラート、メトリクスアラート
- イベントチャンネル: SMS、Mobile、3rd-party プラグイン、プラグイン

イベントタイトル

Event title format

```
[Level][Platform][ProjectName][ApplicationName][EventTitle]
```

Event title example

```
[Info][JAVA][アプリケーションプロジェクト][TC-0-1-8081][CRITICAL_HIGH_MEMORY]
```

❗ ApplicationName が設定されていない場合、イベントタイトルから省略されます。

- Platform は以下のいずれかで表示されます。
 - JAVA
 - NODEJS
 - PYTHON
 - PHP
 - DOTNET
 - GO
 - POSTGRESQL



- ORACLE
- MYSQL
- MSSQL
- BSM_JAVA
- CLOUDWATCH
- TIBERO
- KUBERNETES
- KUBE_NS
- URLCHECK
- URLCHECK_ADMIN
- CUBRID
- ALTIBASE
- CLUSTER
- REDIS
- MONGODB
- VR
- RUM

イベントメッセージ

イベントメッセージに含めることができる情報です。 `optional` が `false` の場合、常にメッセージに含まれます。
`optional` が `true` の場合、該当データが確認できるときのみ表示されます。

イベントメッセージ例

```
Project Name : アプリケーションプロジェクト  
Project Code : 3  
Agent Name : TC-0-1-8081  
Message : RECOVERED: Memory is too high. less than 10%  
Event Time : 2022-04-12 18:53:24 +0900
```

Event OFF Time : 2022-04-12 18:53:24 +0900
 Alert Type : APPLICATION_MEMORY
 Metric Name : memory
 Metric Value : 20
 Metric Threshold : 10
 Stateful : true

表 | イベントメッセージ構成要素

En	Ko	対応するアラートタイプ	説明
Project Name	プロジェクト名	全て	-
Project Code	プロジェクトコード	全て	-
Agent Name	エージェント名	全て (optional)	oname
Message	イベントメッセージ	全て	-
Alert Type	イベント種別	全て	下記の AlertType 表を参照
Event Time	イベント発生時間	全て	2022-04-13 10:40:49 +0900 の +0900 は GMT オフセット
Event Off Time	イベント解除時間	全て (optional)	2022-04-13 10:40:49 +0900 の +0900 は GMT オフセット
Metric Name	メトリクス名	全て (optional)	イベント条件判定に使用されるメトリクス名
Metric Value	メトリクス値	全て (optional)	メトリクス値が閾値を超えた場合に含まれる
Metric Threshold	メトリクス閾値	全て (optional)	メトリクス条件の閾値

En	Ko	対応するアラートタイプ	説明
Stateful	復旧イベント通知	全て (optional)	復旧通知が有効な場合 <code>true</code> 、それ以外は <code>false</code>
Event Rule	イベント発生条件	メトリクスアラート	-
Target Filter	イベント対象選択	メトリクスアラート	特定対象から収集されたメトリクスのみ評価
Repeat Count	イベント繰り返し回数	メトリクスアラート	繰り返し時間内に指定回数条件を満たすと発生
Repeat Duration	イベント繰り返し時間	メトリクスアラート	繰り返し回数を評価する時間
Receiver	受信者	メトリクスアラート	-
Query	MXQL クエリ	複合メトリクスアラート	-
Rule	イベント発生条件	複合メトリクスアラート	-
Query Period	クエリ期間	複合メトリクスアラート	-
Query Interval	クエリ間隔	複合メトリクスアラート	-
Silent Time	サイレント時間	複合メトリクスアラート	-
Query	URL	Exception アラート	例外を発生させたリクエストの URL
TXID	トランザクション ID	Exception アラート	-

En	Ko	対応するアラートタイプ	説明
Class	エラークラス名	Exception アラート	-
Log Message	ログメッセージ	サーバーファイル ログアラート	-
Log File	ログファイルパス	サーバーファイル ログアラート	-
IP	IP	サーバーアラート全 体	-
CPU	CPU	サーバーアラート全 体	イベント発生時のスナップショット
CPU_load1	CPU_load1	サーバーアラート全 体	イベント発生時のスナップショット
CPU_loadPerCore	CPU_loadPerCore	サーバーアラート全 体	イベント発生時のスナップショット
Memory	メモリ	サーバーアラート全 体	イベント発生時のスナップショット
Swap	スワップ	サーバーアラート全 体	イベント発生時のスナップショット
Disk Name Used Percent Free Size IO Percent	ディスクパフォー マンス	サーバーアラート全 体	イベント発生時のスナップショット
Name Bps Pps	トラフィックパフ ォーマンス	サーバーアラート全 体	イベント発生時のスナップショット
Message Time Name	処理履歴メッセー ジ	サーバーアラート全 体	-

❗ このセクションでは、該当するイベントで提供可能な情報を最大限に表示します。

- **AlertType** は以下のいずれかの値で表示されます。

AlertType	説明
APPLICATION_CPU	アプリケーション CPU アラート
APPLICATION_MEMORY	アプリケーション メモリアラート
APPLICATION_DISK	アプリケーション ディスクアラート
APPLICATION_ACTIVE_TRANSACTION	アプリケーション アクティブTXアラート
APPLICATION_ERROR_TRANSACTION	アプリケーション エラートランザクションアラート
APPLICATION_SLOW_TRANSACTION	アプリケーション トランザクション応答時間アラート
METRICS	メトリクスアラート
COMPOSITE_METRICS	複合メトリクスアラート
ANOMALY	異常値アラート
LOG_REALTIME	リアルタイムログアラート
COMPOSITE_LOG	複合ログアラート
SERVER_REBOOT	サーバー 再起動アラート
SERVER_NO_DATA	サーバー データ未受信アラート
SERVER_PORT	サーバー ポートアラート
SERVER_NETWORK_IOPS	サーバー ネットワーク IOPS アラート
SERVER_NETWORK_BPS	サーバー ネットワーク BPS アラート



AlertType	説明
SERVER_DISK_IO	サーバー ディスク I/O アラート
SERVER_DISK_QUOTA	サーバー ディスク使用量アラート
SERVER_DISK_INODE	サーバー inode アラート
SERVER_CPU	サーバー CPU アラート
SERVER_MEMORY	サーバー メモリアラート
SERVER_CPU_STEAL	サーバー CPU steal アラート
SERVER_MEMORY_SWAP	サーバー スワップメモリアラート
SERVER_LOG_FILE	サーバー ログファイルアラート
SERVER_WINDOW_EVENT	サーバー Windows イベントアラート
SERVER_OFF	サーバー アラート OFF 通知
SERVER_ACKNOWLEDGE	サーバー 処理確認通知
SERVER_PROCESS_COUNT	サーバー プロセス数アラート
SERVER_PROCESS_CPU	サーバー プロセス CPU アラート
SERVER_PROCESS_MEMORY	サーバー プロセス メモリアラート
SERVER_PROCESS_OFF	サーバー プロセスアラート OFF 通知
AGENT_ACTIVE	エージェント 有効化アラート
AGENT_INACTIVE	エージェント 無効化アラート



AlertType	説明
AGENT_REACTIVATED	エージェント 再有効化アラート
URL	URL アラート
TOO_MANY_EVENT	イベント多発アラート
CLOUD_WATCH	CloudWatch アラート
EXCEPTION	例外アラート

イベント履歴

ホーム > プロジェクト選択 > 🔍 アラート通知 > イベント履歴

イベント履歴では、プロジェクトで発生したさまざまなイベントを確認し、詳細情報を分析できます。Elasticsearch DSL クエリをサポートしており、指定した条件に一致するイベントを迅速かつ正確に検索できます。

イベント履歴の確認

検索結果でイベントタイトルをクリックすると、そのイベントが発生した時間帯の詳細データを分析できる画面に移動します。これにより、イベント発生前後の状況を総合的に把握できます。

1. アラート通知 > イベント履歴 に移動してください。
2. イベント履歴 画面の 時間選択 で、イベント履歴を確認する期間を設定してください。
3. フィルター入力欄にクエリを入力し、[虫眼鏡] ボタンをクリックしてください。
 - クエリを入力しない場合、すべてのイベントが表示されます。

🔧 カラム設定

表示するカラムや非表示にするカラムを設定できます。

1. イベント履歴 画面の左上にある [🔧 カラム設定] ボタンをクリックしてください。
2. カラム設定 画面で、表示または非表示にするカラムを選択してください。
 - 表示項目 でカラムの順序を変更できます。
 - カラム名検索 欄で目的のカラムを検索できます。

カラム項目	説明
イベント名	イベントのタイトル
メッセージ	イベントのメッセージまたはスナップショット情報

カラム項目	説明
イベント状態	イベントの現在の状態
エージェント名	イベントが発生した対象エージェントの名前 (Oname) - 特定のエージェントに紐づかないイベントの場合は空欄で表示されます
エージェントグループ	イベントが発生した対象エージェントの種類 (OkindName) - 特定のエージェントに紐づかないイベントの場合は空欄で表示されます
エージェントノード	イベントが発生した対象エージェントのサーバー (OnodeName) - 特定のエージェントに紐づかないイベントの場合は空欄で表示されます
イベント発生時刻	イベントが発生した時刻
イベント解消時刻	イベントが解消された時刻 - 解消状態を持たないイベントの場合は - と表示されます

3. 設定が完了したら、[確認] ボタンをクリックしてカラム設定を反映してください。

↓ CSV ダウンロード

検索結果を CSV ファイル形式でダウンロードできます。

1. アラート通知 > イベント履歴 で [CSV] ボタンをクリックしてください。
2. 現在のフィルター条件に一致するイベント一覧が CSV ファイルとして保存されます。

進行中のイベントのみ表示

現在進行中の状態のイベントのみを表示できます。

1. アラート通知 > イベント履歴 で [進行中のイベントのみ表示] ボタンをクリックしてください。
2. フィルタークエリに次の条件が自動的に追加され、検索が実行されます。

```
Stateful: true and Status : "ON"
```

イベント検索

- **正確な検索:** 必要なイベントだけを取得できるよう、条件を正確に指定
- **複合条件:** 複数のフィールドを同時に検索し、複雑な条件も一度に処理
- **柔軟なパターン:** ワイルドカードや部分一致など、さまざまな検索方式をサポート
- **高速パフォーマンス:** インデックス化されたデータを活用した高速検索

検索可能なフィールド

フィールド名および値は大文字と小文字を区別しません。

- **フィールド名タグ:** フィールドがサポートされるイベントタイプ
- **フィールドタイプ:** フィールドのデータ型
 -  #: Number
 -  T: String
 -  C: Boolean
 -  D: Date

イベント情報フィールド

フィールド名	タイプ	説明	例
Title	String	イベントタイトル	"Database Connection Error"
OffTitle	String	イベント解消タイトル	"RECOVERED: Database Connection Error"
Message	String	イベントメッセージ	"Connection timeout occurred"
OffMessage	String	イベント解消メッセージ	"RECOVERED: Connection timeout occurred"

フィールド名	タイプ	説明	例
Level	String	現在のイベントレベル	Critical, Warning, Info
OriginLevel	String	元のイベントレベル	Critical, Warning, Info
Status	String	イベント状態	ON, OFF, CANCEL, ACKNOWLEDGE, MAINTENANCE, DISABLED
ActCount	Number	発生回数	2
MetricName	String	指標名	"memory"
MetricValue	String	指標値	"85.5"
OffValue	String	解消時の値	"72.8"
MetricThreshold	String	閾値	"80"
alertType	String	イベント種別	"BASIC", "METRICS", "TRANSACTION" など
alertId	String	イベントルールの一意識別子	"zf3uojer0fv4v7"

イベントタイプ別対応フィールド

ⓘ 特定のイベントタイプでのみサポートされるフィールドです。該当しないイベントタイプの場合、値が存在しないことがあります。

フィールド名	タイプ	イベント種別	説明	例
eventRule	String	基本、メト	イベント発生	"memory ≥ 80"

フィールド名	タイプ	イベント種別	説明	例
		リクス	ルール	
field	String	リアルタイムログ	ログ検索キー	"content"
keyword	String	リアルタイムログ	ログ検索値	"Error"
logCategory	String	リアルタイムログ	ログカテゴリ	"AppLog"
logContent	String	リアルタイムログ	ログ内容	"00:00000:00009:2025/04/06 23:03:52.55 server Error: 1601, Severity: 17, State: 3\n00:00000:00009:2025/04/06 23:03:52.55 server There are not enough 'user connections' available to start a new process."

対象フィールド

フィールド名	タイプ	説明	例
Oid	Number	エージェントの一意識別子	-98765432
Oname	String	エージェント名	"web-server-01"
Okind	Number	エージェント種別の一意識別子	867318026

フィールド名	タイプ	説明	例
OkindName	String	エージェント種別名	"web-server"
Onode	Number	エージェントノードの一意識別子	334634079
OnodeName	String	エージェントノード名	"production-node-1"

イベント固有値フィールド

フィールド名	タイプ	説明	例
Id	Number	イベントの一意識別子	5768121
UUID	String	イベント固有識別子	"550e8400-e29b-41d4-a716"

状態／フラグフィールド

フィールド名	タイプ	説明	例
Stateful	Boolean	状態ベースイベントかどうか	true, false
Disabled	Boolean	無効化されたイベントかどうか	true, false
Escalation	Boolean	エスカレーション有無	true, false
SystemEvent	Boolean	システムイベントかどうか	true, false

検索タイプ別クエリ構文

❗ 検索の基本構文については、[WhaTap ログ検索クエリ構文](#) ドキュメントを参照してください。

1. キーワード検索

「Database Connection」という文字列を含むイベント

```
title: "Database Connection"
```

2. 複数値検索

指定した複数の値のうち、いずれかを含むイベントを検索します。

```
title: Database Connection
```

現在のイベントレベルが Info または Warning のイベント

```
level: info warning
```

3. パターン検索

ワイルドカードを使用してパターン検索ができます。

名前パターンで特定のエージェントで発生したイベントを検索

```
oname: web-*-prod
```

タイトルが Connection で終わるイベントを検索

```
title: *Connection
```

4. 複合条件検索

複数の条件を組み合わせて、より正確に検索できます。

レベルが Critical で、タイトルに Database を含むイベント

```
level: critical and title: Database
```

特定のエージェントにおける Warning または Critical のイベント

```
oname: web-server-01 and level: Warning Critical
```

5. OR 条件検索

複数の条件のうち、**いずれか一つでも満たす**イベントを検索します。

タイトルまたはメッセージに特定のキーワードを含むイベント

```
title: Connection or message: Connection
```

6. 除外条件

特定の条件を除外して検索できます。

Info レベルを除いたすべてのイベント

```
not level: info
```


リアルタイムアラート

リアルタイムアラート機能は、イベント設定で設定したイベントが発生した際に、ポップアップ形式で表示されます。イベント発生と同時に通知を受け取ることで、問題を迅速に把握し、対応することができます。また、すでに発生した通知や見逃した通知の履歴を確認することで、リアルタイム対応が難しかった状況を把握し、解決に役立てることができます。

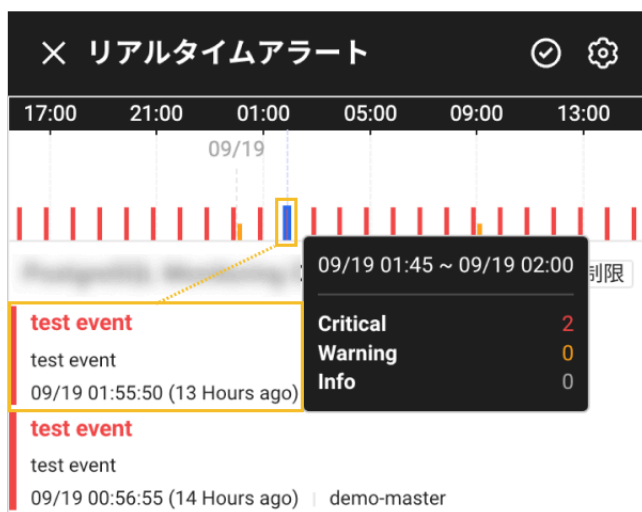
リアルタイムアラートを確認する

リアルタイムアラートは、画面右上にポップアップとして表示されます。通知が発生した時刻、タイトル、メッセージを確認できます。

- 個別の通知を閉じるには、メッセージ右上の ✕ ボタンをクリックしてください。
- 画面に表示されているすべてのポップアップ通知を閉じるには、**すべての通知を閉じる** をクリックしてください。

通知履歴を確認する

1. プロジェクトを選択して表示される画面の最上部にある 🔔 をクリックしてください。
2. リアルタイムアラート ウィンドウが表示されます。
 - 現在のプロジェクトについて、直近24時間以内の通知を最大100件まで確認できます。
 - 個別のイベント項目を選択すると、関連するメニューに移動します。
 - リアルタイムアラート ウィンドウ内の時間別チャートで任意の時間帯を選択すると、該当するイベント項目に移動します。



- すべての通知履歴を確認するには、**全体の通知履歴** ボタンをクリックしてください。**アラート通知 > イベント履歴** メニューに移動します。

❗ リアルタイムアラート の一覧が空の場合、**全体の通知履歴** ボタンが表示されます。受信タグ未設定の通知を受け取る オプションを無効にしてください。

イベントを一括で解消する

リアルタイムアラート の一覧から、進行中のすべてのイベントを一括で解消できます。

イベント解消が完了するまでに、約30秒～1分程度かかる場合があります。

- イベント一括処理** 機能を使用すると、現在進行中のイベントに対する解消通知が大量に発生する可能性があります。
- イベント一括処理** 機能を使用した後でも、イベントルールが有効であり、データが引き続きしきい値を超えている場合、イベントは再度発生することがあります。
- イベント解消通知は、**アラート通知 > イベント設定** メニューでイベントを編集し、**イベントが解消されたときに追加通知** または **解消通知** オプションを無効にすることで停止できます。

使用方法:

1. プロジェクトを選択して表示される画面上部の  をクリックします。
2. リアルタイムアラート ウィンドウが表示されたら、右上の  をクリックします。
3. リアルタイムアラートリストの進行中イベントを一括で解消します。 というメッセージが表示されたら、解消 ボタンをクリックします。

リアルタイムアラートを設定する

1. プロジェクトを選択して表示される画面上部の  をクリックします。
2. リアルタイムアラート ウィンドウが表示されたら、右上の  をクリックします。
3. リアルタイムアラート設定画面で各項目を設定します。
 - ポップアップ通知: リアルタイムアラート ウィンドウが閉じている状態でも、ポップアップ形式で通知を表示します。
 - 解消済みイベントを表示: リアルタイムアラート 一覧に解消済みイベントも含めて表示します。
 - 通知音の試聴: ブラウザで再生される通知音を事前に確認できます。Critical または Warning の横にある  をクリックしてください。音が再生されない場合は、ブラウザの設定をご確認ください。

システムイベント

WhaTap モニタリングは、プラットフォーム全体で自動的に検知・生成される共通イベントを提供します。本ドキュメントでは、システムによって発生するイベントの種類、発生条件、および対応方法について説明します。

EVENT_BLOCKED_BY_SYSTEM

プロジェクト内で短時間に過剰な数のイベントが発生した場合、WhaTap システムがこれを検知し、イベントの発生を一時的にブロックします。イベントがブロックされると、システムは次のメッセージをユーザーに通知します。このイベントが発生した場合は、イベント設定内容を確認し、必要に応じて調整してください。

 **Critical**



イベントメッセージ

Excessive events detected, event blocked for stability. name: server agent

プロジェクト番号	8
プロジェクト名	new server
対象	server agent
イベント名	EVENT_BLOCKED_BY_SYSTEM
イベント日時	(GMT+0900) 2024-11-18 18:03:18

WhaTapにログイン

new serverプロジェクトのserver agentから過度なイベントが検知されたため、イベント発生が遮断されます。server agentから一定期間の間にイベントが発生しなくなると、遮断が自動で解除されます。

ブロック基準

イベントは次の基準に基づいてブロックされます。

1. エージェント単位のブロック

1つのエージェントで **1分以内に10件** のイベントが発生した場合、そのエージェントから発生する以降のイベントはブロックされます。

2. 複数エージェントによる全体ブロック

条件 **1** によりイベントがブロックされたエージェントが **1分以内に3件以上** 発生した場合、プロジェクト全体のイベントがブロックされます。

3. プロジェクト単位のブロック

1つのプロジェクトで **1分以内に100件** のイベントが発生した場合、プロジェクト全体のイベントがブロックされます。

ブロック解除方法

システムによってイベントがブロックされた場合、対象のエージェントまたはプロジェクトで **1分間** イベントが発生しない状態を維持する必要があります。

- **アラート通知 > イベント設定** メニューから設定されているすべてのイベントを無効化してください。**イベント設定** を見直し、過剰なイベントが発生しないように閾値を調整する必要があります。
- イベントを設定する前に、**シミュレーション** 機能を活用してイベント発生回数を事前に予測できます。

連携プロジェクト管理

ホーム画面 > プロジェクト選択 > 管理 > 連携プロジェクト管理

データベースモニタリング(DPM) プロジェクトとアプリケーションのモニタリング(APM) プロジェクトを生成した場合、2つのプロジェクトを連携してDPMで収集したモニタリングデータをAPMプロジェクトで確認できます。

データベースで有効になったセッション(アクティブセッション)と連動したアプリケーションのアクティブTXにより、そのトランザクションの詳細情報をリアルタイムで確認できます。このような統合モニタリングにより、アプリケーションとデータベース間の相互作用を深層分析し、リアルタイムで性能ボトルネック現象を診断して解決できます。

連携プロジェクト管理

連携プロジェクトの登録

連携するプロジェクトを選択してから、「追加」ボタンをクリックします。(サポートタイプ: Server, MySQL, PostgreSQL, Oracle, Oracle Pro, Redis, MSSQL, Altibase, MongoDB, Cubrid, Java, Kubernetes, Namespace)

連携プロジェクトのリスト

プロジェクトコード	プロジェクト名	グループ	タイプ
2994	petclinic-Postgresql-apm		Java

連携プロジェクト管理機能を使用すると、製品または機器中心のモニタリングを超えて、複数のプロジェクト間のモニタリングデータを統合して確認できます。これは、アプリケーション、データベースなど、様々なシステムコンポーネントがどのように相互作用するかを分析するのに役立ちます。特に、性能低下が発生した時に装備の問題なのか個別アプリケーションの問題なのか迅速に把握でき、問題解決時間を大幅に短縮できます。連携プロジェクト管理これにより、ユーザーシステム単位で性能データを総合的に分析し、より可視的なインサイトを確保できます。

詳細については、次の動画を通じて確認できます。

❗ 開始前の参考事項

- 現在はJavaプラットフォームのみサポートします。今後のアップデートでサポートプラットフォームを拡張します。
- 連携DBセッション機能の場合、Javaエージェント2.2.33バージョン以降が必要です。
- アプリケーションプロジェクトでデータベースプロジェクトを連携する方法についての詳細は、[次の文書](#)を参照してください。

連携プロジェクトを追加する

1. 管理 > 連携プロジェクト管理メニューに移動します。
2. 連携プロジェクトの登録セクションでプロジェクト選択入力欄を選択します。

連携プロジェクトの登録

連携するプロジェクトを選択してから、「追加」ボタンをクリックします。(サポートタイプ: Server, MySQL, PostgreSQL, Oracle, Oracle Pro, Redis, MSSQL, Alti)

プロジェクト選択

☐ OTE-APM@Front Java 13 OTE-SELF

☐ DB-REDIS-JINS-TEST Redis 19 Jinslab

☐ DB-MARIADB-JINS-TEST MySQL 20 Jinslab

☐ hsnam-ecs-1 Kubernetes 29

☐ hsnam & vm Server 38

☐ DB-POSTGRESQL-JINS-TEST PostgreSQL 60 Jinslab

☐ SQLServer-V1 MSSQL 82

☐ rum_previous Java 88 swlim

☐ java-apm-docker Java 181 bspark-test

追加

3. 選択できるプロジェクトの一覧が表示されたら、連携するプロジェクトを選択します。文字列を入力して、一致するプロジェクトを検索もできます。
4. プロジェクトを1つ以上選択した後に有効になった追加ボタンを選択します。

連携プロジェクトのリストに選択したプロジェクトが追加されます。

❗ 連携プロジェクトを追加するためには、プロジェクト修正権限が必要です。修正権限がある場合連携プロジェクト管理メニューで追加ボタンが有効になります。

連携プロジェクトを確認する

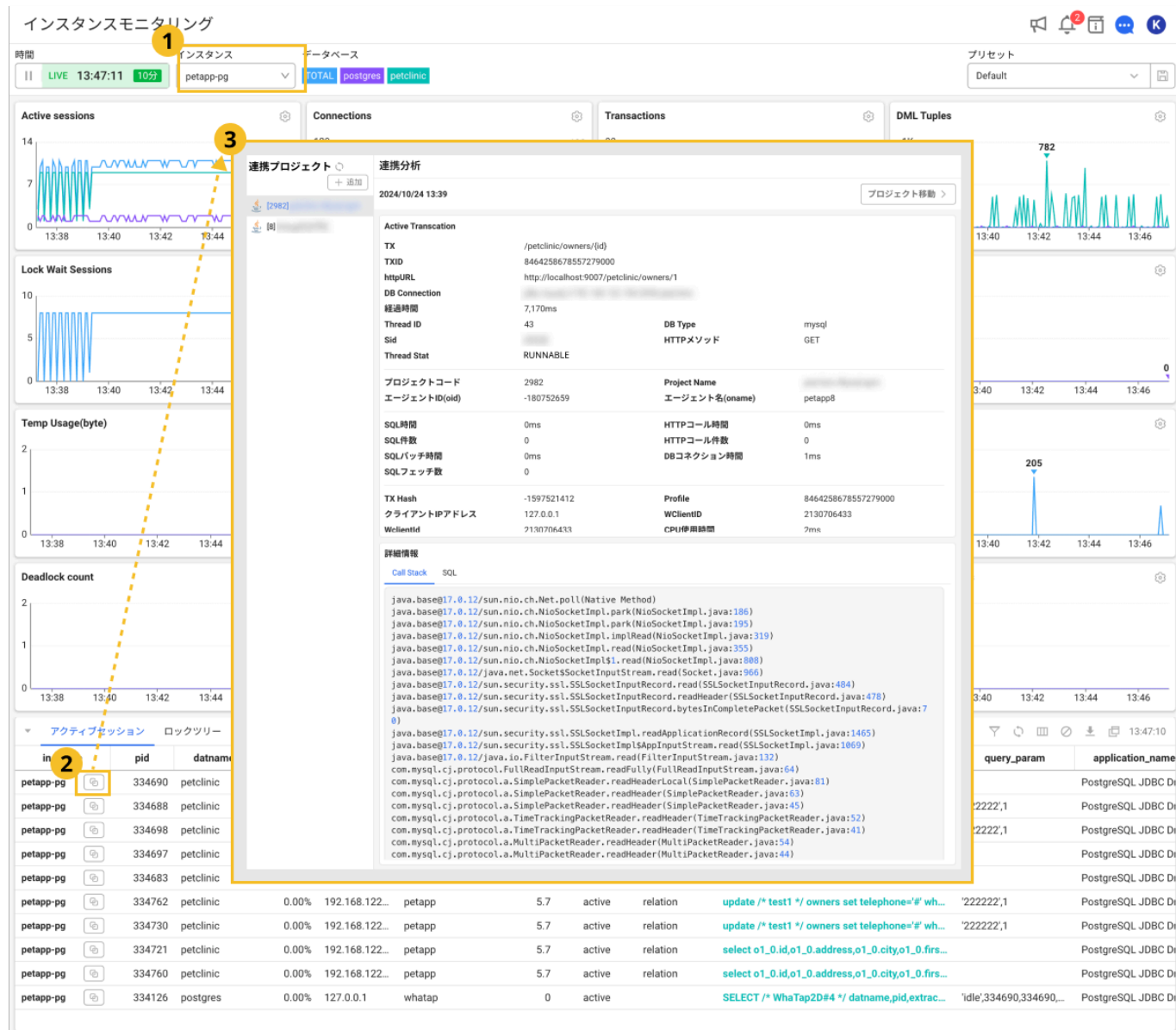
データベースで有効になったセッション(アクティブセッション)と連動したアプリケーションのアクティブTXに関する詳細情報をリアルタイムで提供します。ユーザーは、トランザクションURLおよびデータベース接続情報、DBタイプ、HTTPメソッド、実行時間などの基本情報とともに、トランザクションID、スレッドID、クライアントIP、CPU使用時間などの具体的なメトリクスを確認できます。また、SQLクエリの実行時間、呼び出されたSQLの数、DB接続時間などもモニタリングでき、システムの性能を総合的に分析できます。

特に、スタックトレースによるコールスタック(Call stack) 情報とSQLクエリを一緒に提供し、問題発生時の詳細な原

因を追跡して解決するのに役立ちます。これにより、アプリケーションとデータベース間の相互作用を深く分析し、性能のボトルネック現象を迅速に診断できます。

アクティブTXを追跡する

ダッシュボード > インスタンスモニタリングメニューに移動します。連携分析機能は、リアルタイム照会状態でのみ確認できます。



1. インスタンスオプションで照会したいインスタンス(エージェント)を選択します。
2. アクティブセッションテーブル一覧で連携分析しようとするアクティブセッション項目の🔗ボタンを選択します。
3. 選択したアクティブセッションに関連するトランザクション情報を提供する**連携分析**画面が表示されます。

連携プロジェクト一覧でご希望のプロジェクトを選択し、連携分析情報を確認します。

- ❗ • 画面右上の**プロジェクト移動**ボタンを選択すると、当該プロジェクトの**アプリケーションメニュー**に移動できます。
- 選択したアクティブセッションに連携されたトランザクションがない場合、**データなし**として画面に表示されます。

連携分析画面案内

連携分析画面では、アクティブTXに関する詳細情報およびCall Stack、SQL文を確認できます。

Active Transaction

連携したアクティブTXの詳細を確認できます。

Active Transcation			
TX	/petclinic/owners/{id}		
TXID	8464258678557279000		
httpURL	http://localhost:9007/petclinic/owners/1		
DB Connection	jdbc:mysql://127.0.0.1:3306/petclinic		
経過時間	7,170ms		
Thread ID	49	DB Type	mysql
Sid	49530	HTTPメソッド	GET
Thread Stat	RUNNABLE		
プロジェクトコード	2982	Project Name	petclinic-8464258678557279000
エージェントID(oid)	-180752659	エージェント名(oname)	petapp8
SQL時間	0ms	HTTPコール時間	0ms
SQL件数	0	HTTPコール件数	0
SQLパッチ時間	0ms	DBコネクション時間	1ms
SQLフェッチ数	0		
TX Hash	-1597521412	Profile	8464258678557279000
クライアントIPアドレス	127.0.0.1	WClientID	2130706433
WclientId	2130706433	CPU使用時間	7ms

- **DB Connection:** データベース接続URLです。
- **Thread ID:** そのトランザクションを処理するために使用されたスレッドの固有識別子です。
- **Sid:** 選択したアクティブセッションのIDです。

① SidはアクティブセッションテーブルでPID項目と同じ値です。APMプロジェクトでSQL情報のSID項目とも同じです。

①

▼ アクティブセッション ロックツリー プロセス情報

instance		pid	datname	cpuusage	client_addr	username
petapp-pg		414102	petclinic	0.33%	192.168.122...	petapp
petapp-pg		414200	petclinic	0.00%	192.168.122...	petapp
petapp-pg		414285	petclinic	0.00%	192.168.122...	petapp

連携分析

2024/10/24 13:39 プロジェクト移動 >

Active Transcation

TX

/petclinic/owners/{id}

TXID

8464258678557279000

httpURL

http://localhost:8080/petclinic/owners/1

DB Connection

jdbc:mysql://192.168.122.104:3306/petclinic

経過時間

7,170ms

Thread ID

43

Sid

414102

Thread Stat

RUNNABLE

プロジェクトコード

2982

エージェントID(oid)

-180752659

DB Type

PostgreSQL

HTTPメソッド

GET

Project Name

petclinic Ward app

エージェント名(oname)

petapp8

SQL

接続URL jdbc:mysql://192.168.122.104:3306/petclinic

CRUD Select owners

クエリ (SQLハッシュ: -1083468189) 連携DBセッション パスワード SQLフォーマット 統計 >

SID 414102

```
select cl_b.id,cl_b.address,cl_b.city,cl_b.first_name,cl_b.last_name,pl_b.owner_id,pl_b.id,pl_b.birth_date,pl_b.name,pl_b.type_id,cl_b.telephone from owners cl_b left join pets pl_b on cl_b.id=pl_b.owner_id where cl_b.id=1 order by pl_b.name
```

パラメータマッピング バインド変数 ☒ バインド変数のコピー

バインド変数 : 1

- **Thread Stat:** スレッドの現在の状態です。
- **DB Type:** 連携したデータベースプラットフォームの種類です。

- ❗ • **Javaエージェント2.2.40以前のバージョンではThreadStat値が数値形式で表示されます。文字形式で表示するには、連携したプロジェクトでJavaエージェント2.2.40バージョン以降をインストールします。**
- その他の項目についての詳細は、[次の文書](#)を参照してください。

Call Stack

メソッドや関数呼び出しの順序を追跡する情報を提供します。

詳細情報

[Call Stack](#) [SQL](#)

```
java.base@17.0.12/sun.nio.ch.Net.poll(Native Method)
java.base@17.0.12/sun.nio.ch.NioSocketImpl.park(NioSocketImpl.java:186)
java.base@17.0.12/sun.nio.ch.NioSocketImpl.park(NioSocketImpl.java:195)
java.base@17.0.12/sun.nio.ch.NioSocketImpl.implRead(NioSocketImpl.java:319)
java.base@17.0.12/sun.nio.ch.NioSocketImpl.read(NioSocketImpl.java:355)
java.base@17.0.12/sun.nio.ch.NioSocketImpl$1.read(NioSocketImpl.java:808)
java.base@17.0.12/java.net.Socket$SocketInputStream.read(Socket.java:966)
java.base@17.0.12/sun.security.ssl.SSLSocketInputRecord.read(SSLSocketInputRecord.java:484)
java.base@17.0.12/sun.security.ssl.SSLSocketInputRecord.readHeader(SSLSocketInputRecord.java:478)
java.base@17.0.12/sun.security.ssl.SSLSocketInputRecord.bytesInCompletePacket(SSLSocketInputRecord.java:70)
java.base@17.0.12/sun.security.ssl.SSLSocketImpl.readApplicationRecord(SSLSocketImpl.java:1465)
java.base@17.0.12/sun.security.ssl.SSLSocketImpl$AppInputStream.read(SSLSocketImpl.java:1069)
java.base@17.0.12/java.io.FilterInputStream.read(FilterInputStream.java:132)
com.mysql.cj.protocol.FullReadInputStream.readFully(FullReadInputStream.java:64)
com.mysql.cj.protocol.a.SimplePacketReader.readHeaderLocal(SimplePacketReader.java:81)
com.mysql.cj.protocol.a.SimplePacketReader.readHeader(SimplePacketReader.java:63)
com.mysql.cj.protocol.a.SimplePacketReader.readHeader(SimplePacketReader.java:45)
com.mysql.cj.protocol.a.TimeTrackingPacketReader.readHeader(TimeTrackingPacketReader.java:52)
com.mysql.cj.protocol.a.TimeTrackingPacketReader.readHeader(TimeTrackingPacketReader.java:41)
com.mysql.cj.protocol.a.MultiPacketReader.readHeader(MultiPacketReader.java:54)
com.mysql.cj.protocol.a.MultiPacketReader.readHeader(MultiPacketReader.java:44)
```

SQL

連携されたアクティブTXによって実行されたSQLクエリです。

詳細情報

Call Stack

SQL

SQLフォーマット



```
select o1_0.id,o1_0.address,o1_0.city,o1_0.first_name,o1_0.last_name,p1_0.owner_id,p1_0.id,p1_0.birth_date,p1_0.name,p1_0.type_id,o1_0.telephone from owners o1_0 left join pets p1_0 on o1_0.id=p1_0.owner_id where o1_0.id=? order by p1_0.name
```

連携プロジェクトを削除する

1. 管理 > 連携プロジェクト管理メニューに移動します。
2. 連携プロジェクトのリストセクションで連携を解除しようとするプロジェクトを削除するには、一覧の一番右にある  ボタンを選択します。
3. 確認メッセージが表示されたら削除ボタンを選択します。

連携プロジェクトのリストセクションで削除したプロジェクトは除外されます。

実験室

ユーザーに新機能または試験的な機能を提供します。サイドメニューの  **実験室** または **サイトマップ** からアクセスできます。商品によって提供する機能が異なる場合があります。

✔ サイトマップから実験室メニューにアクセスする

サイドメニューに  **実験室** が表示されていない場合は、 **サイトマップ** からアクセスできます。

1. 画面左下の  **サイトマップ** アイコンをクリックしてください。
2. **サイトマップ** ウィンドウの右下にある **実験室** セクションで、目的のメニューを選択してください。

⚠ 注意

実験室の機能は現在開発中のベータ版であり、予期せぬ挙動やエラーが発生する可能性があります。十分な検証の上でご利用ください。重要なデータがある環境や運用環境では使用を推奨しません。フィードバックや問題点に対するお問い合わせは、サポートチーム(support@whatap.io)までご連絡ください。

MXQL Data

MXQLは、WhaTapの性能データ(メトリクス)を柔軟に検索するためのクエリ言語です。プロジェクトに含まれる複数のエージェントから収集したメトリクスを統合的に照会・活用するために使います。**MXQL**の詳細については、[次のドキュメント](#)をご参照ください。