

Network Performance Monitoring

release date. 2026. 09. 11.

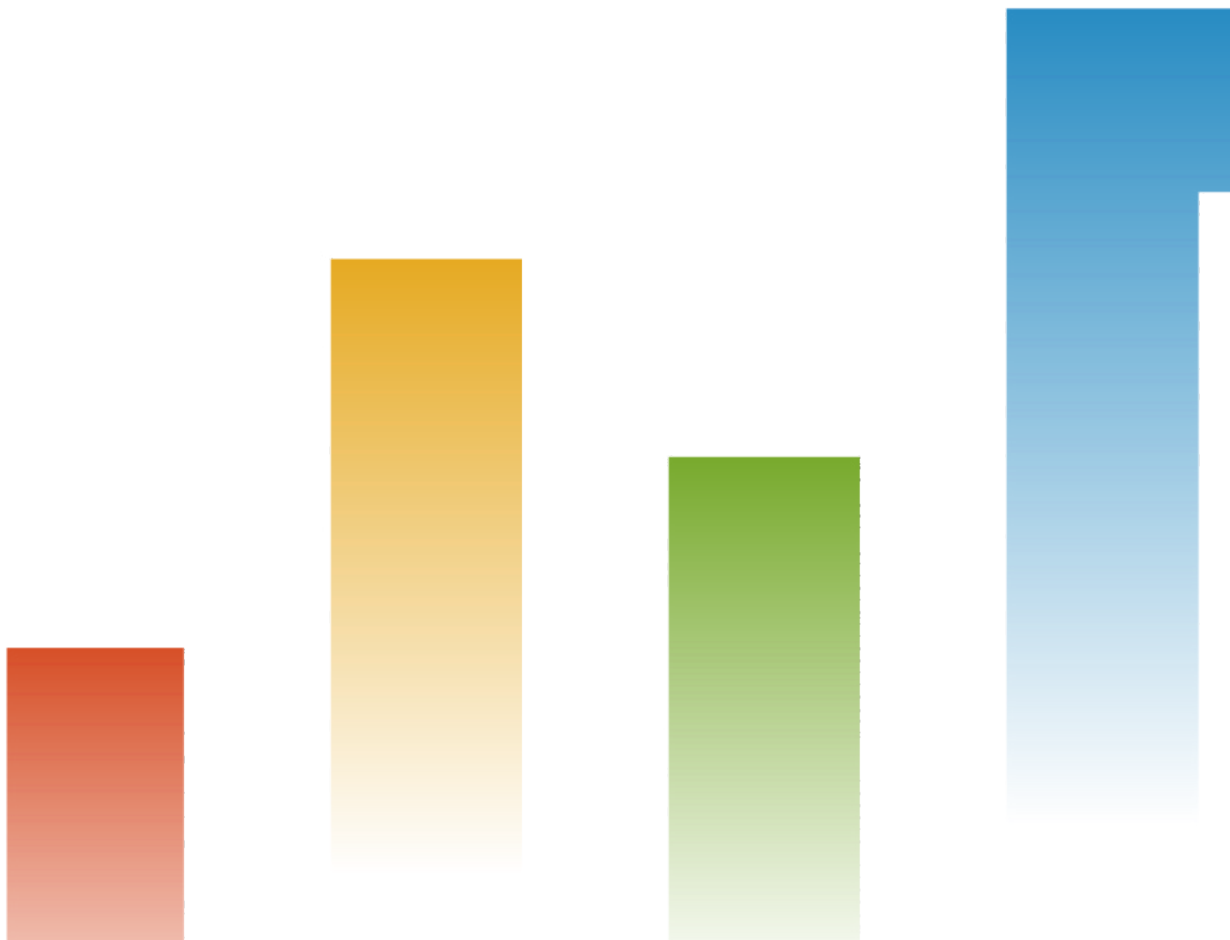


Table of Contents

• ネットワーク性能モニタリング	3
◦ はじめに	8
◦ サポート環境	10
◦ エージェントのインストール	13
▪ AWSリソース情報の収集オプション	16
◦ tagRule設定	19
◦ ダッシュボード	
▪ ネットワークトポロジ	27
▪ ネットワーク推移(TCP)	38
◦ 分析	
▪ TCPセッション / UDPセッション	43
◦ メトリクス	50
▪ ネットワーク性能指標	52
▪ メトリクス照会	56
◦ アラート通知	58
▪ イベント設定	60
▪ メトリクス	63
▪ 通知設定	84
▪ イベント通知フォーマット	106
▪ イベント履歴	114
▪ リアルタイムアラート	122
▪ システムイベント	125
◦ 実験室	127

ネットワーク性能モニタリング

ネットワーク性能モニタリング(Network Performance Monitoring、以降NPM)は、ネットワーク間の混雑および障害を把握し、ユーザーのネットワーク性能改善に活用するためにプロセスのネットワーク通信情報を収集し、可視化するソリューションです。

ネットワーク性能モニタリングの必要性

ネットワーク環境の変化

ネットワークモニタリングは、既存の機器中心のモニタリングからプロセス中心のモニタリングに変化する必要があります。既存のネットワークモニタリングは、サーバーおよびネットワーク機器に従属的な情報をSNMPプロトコルを通じて取得して、可視化する方法で動作します。最近、ネットワーク市場はクラウドと仮想化という大きな枠組みで成長しており、これにより既存のモニタリング方式では障害の状況や要素を判断しづらくなっています。

- クラウド

サーバーおよびネットワーク機器を構成してサービスを運営していた既存のオンプレミス(On-premise)環境から、クラウドサービスプロバイダ(CSP)が提供する仮想環境上でサービスを運営する形に変化しています。物理的な実体がないクラウド環境での重要なモニタリング対象は、実際に作業を遂行するプロセスです。

- 仮想化

コンテナベースの仮想化環境では、複数のコンテナが1つの機器で実行されます。各コンテナは独立したネットワークスタックを持ち、1つの機器でも複数のコンテナ間の通信が存在します。既存の機器ベースのネットワークモニタリングでは、問題が発生したコンテナおよびプロセスまで区別することは困難です。

Network Function Virtualization(NFV)を通じて、別途のH/Wで具現された既存のネットワーク機能をソフトウェアで仮想化させる努力も続いています。

既存のネットワーク性能モニタリング方式では、クラウドと仮想化によって追跡できないグレーゾーンが発生するようになりました。WhaTapのNPMはプロセスを対象に、ネットワーク動作を追跡するため、クラウドと仮想化に対する環境変化に対応できます。

ネットワーク複雑性の増加

Micro Service Architecture(MSA)ベースのサービスが増えており、特有の柔軟性と拡張性によってネットワーク関係が複雑になっています。高可用性(High Availability、HA)構成やScale Outなど、サービス安定化のために行う作業もネットワークの複雑さに影響を与えています。関係が複雑になるにつれ、負荷発生の有無を認識し、負荷区間を把握する過程がますます難しくなっています。

WhaTapのNPMは、各プロセスが実行するネットワーク動作を追跡し、複雑なコンポーネントと状態情報をトポロジを通じて直感的に理解できるように可視化します。

期待効果

プロセス単位で収集される指標を通じて、次の効果が期待できます。

- プロセス単位のネットワーク指標モニタリング
- サービス従属性の把握と改善
- ネットワークコストの最適化
- ネットワークトラブルに対するMTTD、MTTRの減少
 - **MTTD**(Mean Time To Detect): トラブル発生を検出するまでの時間
 - **MTTR**(Mean Time To Repair): トラブル発生を検出した後、復旧までにかかった時間

主な特徴

複雑なネットワークをグループ化して、区間別のパフォーマンス指標をすばやくモニタリングできます。トポロジダッシュボードを通じてネットワークの状況をリアルタイムでモニタリングができるし、タグルール設定機能でネットワーク情報をグループ化および簡素化させることで、ネットワークの障害区間をより素早く把握できます。

• 安定的で詳細なデータ収集

WhaTap NPMでは、eBPF技術を活用してデータを収集します。eBPFは、カーネルで発生する様々なイベントを収集し、サンドボックス環境で追加的な動作を実行できる技術です。カーネルのイベントを収集するので、プロセスレベルの詳細な情報が取得でき、サンドボックスベースで動作するため安定した運用が可能です。

• タグルール設定による可視性の確保

ネットワークモニタリングは、さまざま複雑なネットワークのデータを簡単に理解できるような方法を提供しなければなりません。WhaTap **NPM**は、ユーザーの目的と意図に応じてネットワーク要素を表現できるようにタグルール設定機能をサポートします。ユーザーサービスの特性に合わせて様々なカスタマイズが可能です。

- **Raw Dataによるトラブルシューティング**

サーバーに接続しなくても、障害状況が解消された後も、特定時点でネットワーク状態をメトリクス基準で確認でき、関連するサーバーやプロセスなどの問題把握をサポートします。

① Raw Dataを活用する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。

- **Flexボードによる様々な基準のカスタムチャート**

収集された情報を通じて、様々な基準でカスタムチャートを作成できます。

主な指標の案内

5つのネットワーク指標を介してネットワーク性能をモニタリングします。

- **品質指標**

Latencyと**Jitter**指標を提供し、性能問題およびトラブルの有無を判断する基準として利用します。TCPベースのネットワーク通信を活用するケースから収集が可能です。

- **Latency:** ネットワーク応答時間を意味する指標です。応答時間の高い区間は、最終ユーザーに良くない経験を与える可能性があります。
- **Jitter:** ネットワーク応答時間の変動性を意味する指標です。**Jitter**値が高いという意味は、当該区間のネットワーク混雑、パケット移動経路の変化が頻繁に発生するという意味に解釈できます。特定ネットワーク区間にトラブルが存在する可能性があります。またパケット到着の順番がねじれてパケット再送信およびロスが発生し、サービス品質の低下につながります。

- **ネットワーク使用量の指標**

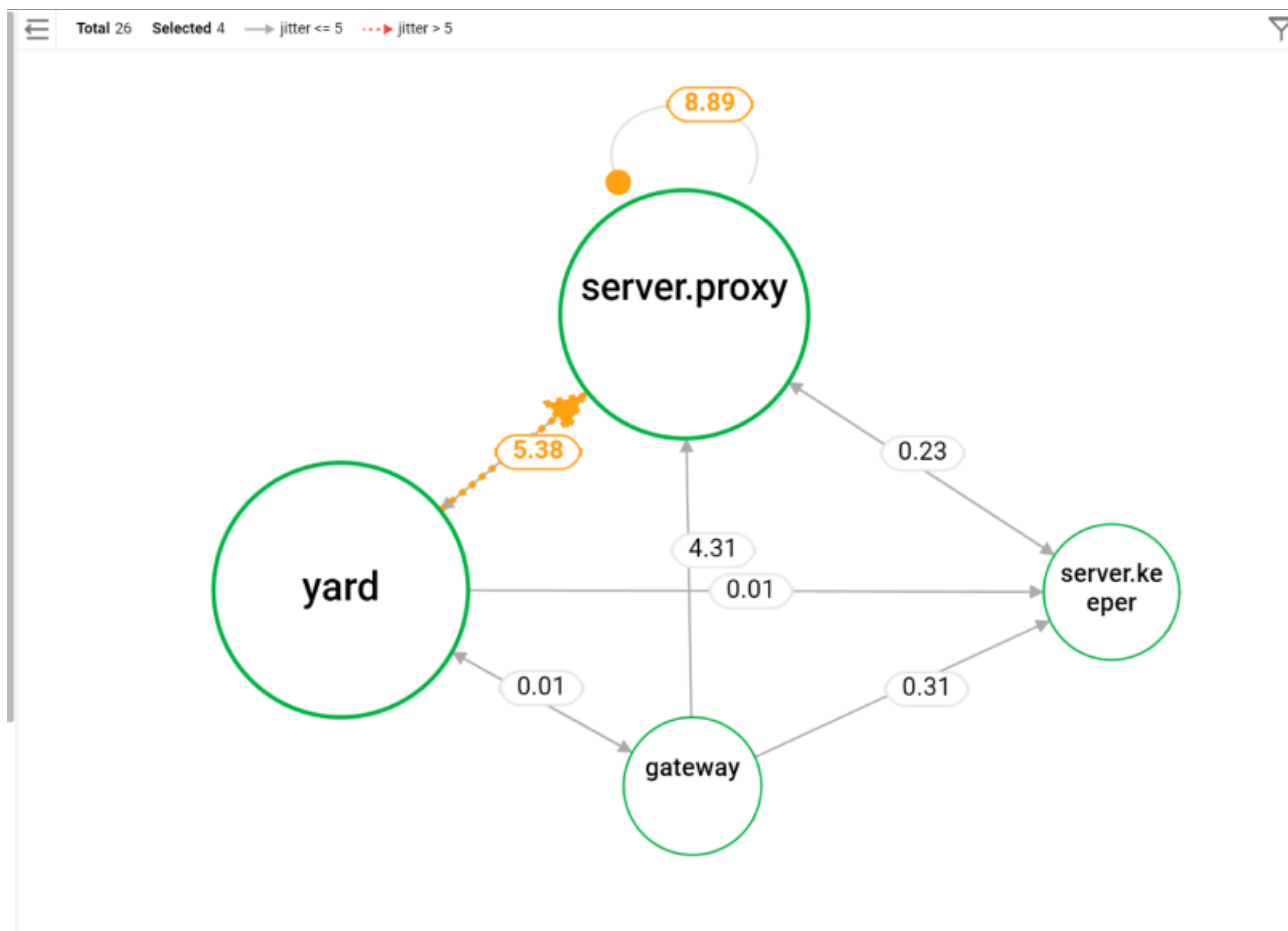
bps、**pps**、**session count**の3つの基準でネットワーク使用量に関する情報を提供し、意図していない外部トラフィックの流入(DDoSなど)やトラフィックパターンなどを通じて、クラウド環境を最適化するのに活用できます。

- **bps:** bits per second(bps)は、毎秒伝送bit数を示す指標です。
- **pps:** packets per second(pps)は、毎秒の提出パケット数を示す指標です。

- **session count**: ユニークなセッション数を示す指標です。

指標活用の案内

- ネットワーク指標はサーバーのCPU使用量と同様に、ユーザーの目標しきい値があることをお勧めします。
 - サービス信頼性のためにSLI(サービスレベル指標)、SLO(サービスレベル目標)を定義し、これをベースに一定水準のサービスを着実に提供できなければなりません。
 - もし目標値が不明な場合は、WhaTapが提供する初期基準でデータを収集し、その後目標を設定してサービスの改善を試みてください。
- 現在のWhaTapでは、デフォルトのしきい値を次のように適用します。ユーザー環境およびSLO目標値に合わせて変更できます。
 - **Jitter**: 30ms
 - **Latency**: 200ms
 - **bps**: 128Mib/s
 - **pps**: 50k
 - **session count**: 100
- トポロジーにしきい値を適用して、しきい値を超えた区間をすばやく把握し、改善するかどうかの判断をサポートします。



- 各指標がしきい値を超えた時点で通知を送り、そのイベント情報を確認できます。
- ネットワーク指標は、一般的な状況で平らな水平形態のチャートを期待しています。平均値から急激に変化する指標を通じて、トラブルの有無を判断し、区間を識別できます。

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

WhaTapのNPMを使用するためには、運用中のサーバーに**NPM Agent**をインストールする必要があります。**NPM**は、OSカーネルが提供するeBPF技術を使用しているため、OSバージョンの確認が必要です。

- ❗ 次の一覧以外のバージョンはテストが必要です。
- ・ 現在Windows、macOSには対応していません。

Linux

OS配布版ごとに適用できる最低バージョン

Red Hat

REHL	サポートの有無	基本カーネルバージョン
旧バージョン	X	-
REHL 7.x	X	3.10
REHL 8.x	O	4.18
REHL 9.x	O	5.14

CentOS	サポートの有無	基本カーネルバージョン
旧バージョン	X	-
CentOS 7.x	X	3.10
CentOS 8.x	X	4.18

CentOS	サポートの有無	基本カーネルバージョン
CentOS Stream 8.x	O	4.18
CentOS Stream 9.x	O	5.14

Amzon Linux	サポートの有無	基本カーネルバージョン
旧バージョン	X	-
Amazon Linux 2.0.20220121.0	X	4.14
Amazon Linux 2.0.20220207.0	O	5.10

Debian

Debian	サポートの有無	基本カーネルバージョン
旧バージョン	X	-
Debian 10 (buster)	X	4.19
Debian 11 (bullseye)	O	5.10
Debian 12 (bookworm)	O	6.1

Ubuntu	サポートの有無	基本カーネルバージョン
旧バージョン	X	-
Ubuntu 16.04	X	4.4.0
Ubuntu 18.04	X	4.15.0
Ubuntu 20.04	O	5.4

Ubuntu	サポートの有無	基本カーネルバージョン
Ubuntu 22.04	O	5.15

エージェントのインストール

Linux環境にインストールする

インストール案内セクションでLinuxタブを選択し、画面の案内に従ってエージェントのインストールを進めます。

エージェントダウンロード

モニタリング対象サーバーにエージェントファイルをダウンロードします。

- [Amazon Linux](#) [Debian/Ubuntu](#) [Red Hat/CentOS](#)

```
sudo rpm --import https://repo.whatap.io/centos/release.gpg
echo "[whatap]" | sudo tee /etc/yum.repos.d/whatap.repo > /dev/null
echo "name=whatap packages for enterprise linux" | sudo tee -a /etc/yum.repos.d/whatap.repo > /dev/null
echo "baseurl=https://repo.whatap.io/centos/9/\$basearch" | sudo tee -a /etc/yum.repos.d/whatap.repo > /dev/null
echo "enabled=1" | sudo tee -a /etc/yum.repos.d/whatap.repo > /dev/null
echo "gpgcheck=0" | sudo tee -a /etc/yum.repos.d/whatap.repo > /dev/null
sudo yum install whatap-npmd
```

インストール過程でgnupg2パッケージが必要です。

```
wget https://repo.whatap.io/debian/release.gpg -O - | sudo apt-key add -
wget https://repo.whatap.io/debian/whatap-repo_1.0_all.deb
sudo dpkg -i whatap-repo_1.0_all.deb
sudo apt-get update
sudo apt-get install whatap-npmd
```

```
sudo rpm -Uvh https://repo.whatap.io/centos/5/noarch/whatap-repo-1.0-1.noarch.rpm
sudo yum install whatap-npmd
```

ⓘ AWS EC2環境でリソース情報を収集するには、[次の文書](#)を参考にしてください。

互換性チェック及び設定

エージェントをダウンロードしたサーバーの互換性をチェックし、設定情報を入力します。

```
sudo /usr/whatap/whatap-npmd/install.sh {プロジェクトアクセスキー} {収集サーバーIPアドレス}
```

次のように互換性チェック結果と駆動メッセージを確認できます。

```
Supported kernel version : 5.15. OK.
Supported CPU Arch : aarch64, OK.
Move package files
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           Dload  Upload   Total   Spent    Left     Speed
100  14  100  14    0    0    65     0    0:00:00 0:00:00 0:00:00    65
Start service
Installation complete!
```

Kubernetes環境に設置する

インストール案内セクションでKubernetesタブを選択し、画面の案内に従ってエージェントのインストールを進めます。

❗ AWS EKS環境でリソース情報を収集するには、[次の文書](#)を参考にしてください。

エージェント生成

1. エージェントを生成するためにダウンロードボタンを選択してYAMLファイルをダウンロードします。
2. ダウンロードしたYAMLファイルをマスターNodeにアップロードしてください。
3. マスターノードで次のコマンドを実行し、エージェントを作成してください。

```
kubectl apply -f whatap_npmd.yaml
```

4. 次のコマンドを実行してエージェントの動作状態(STATUS)がRUNNING状態であることを確認します。

```
kubectl get pod -n whatap-npm-ns
```

```
ubuntu@Dev-Modules:~/YUH TESTS$ kubectl get pod -n whatap-npm-ns
NAME                                READY   STATUS    RESTARTS   AGE
whatap-npm-ds-6jgdr                1/1    Running   0           31s
whatap-npm-ds-br795                1/1    Running   0           31s
```

TagRuleオプション設定(選択事項)

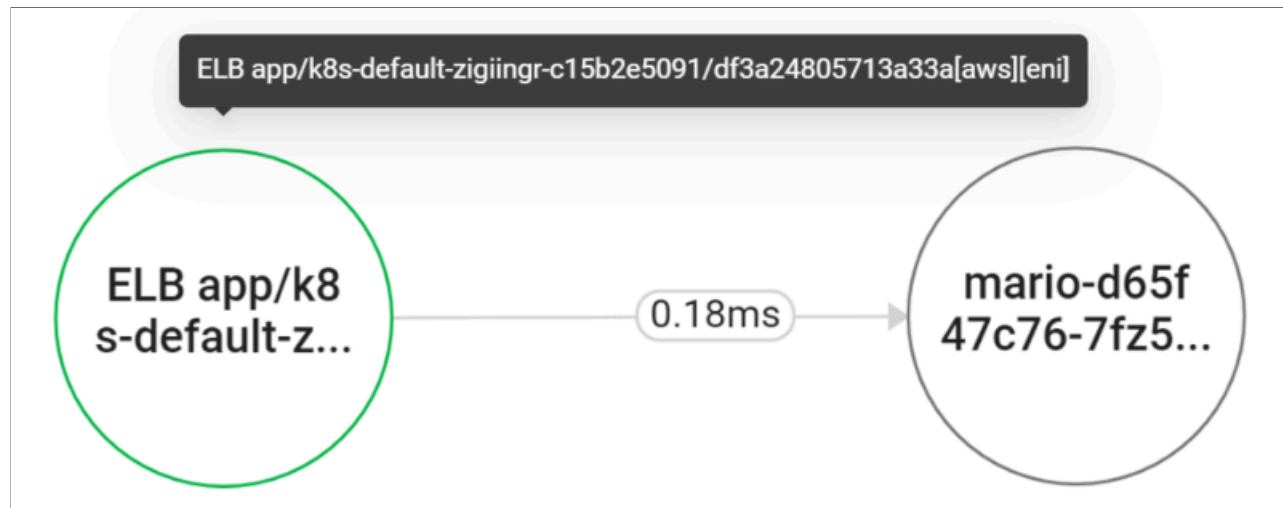
NPMでは、tagRule設定により、不規則で多様な情報を収集し、トポロジーまたはチャートで表示できます。詳細については、[次の文書](#)を参考にしてください。

モニタリングの開始

エージェントが正常に動作すると、モニタリング情報の収集を開始します。収集された情報をサービス画面で確認するには、画面右下のネットワークトポロジボタンを選択します。

AWSリソース情報の収集オプション

AWS環境でネットワーク性能モニタリングを使用するには、各リソース(LBなど)に接続されたEC2ネットワークインタフェース(ENI)情報を収集することでトポロジーに表現できます。AWS要素にエージェントを直接インストールすることができないため、収集されたIPを基準に関連付けられたENIを確認して表現します。



AWSのELBを通じて内部にトラフィックが認可される場合、図のようにENIに表現された情報を通じてトポロジーに表現できます。AWS ENIの `Description` フィールドを基準に表現します。もし `Description` フィールドに何の情報も残さない場合、`NetworkInterfaceId` 属性を提供します。

EC2環境でのAWSリソース権限設定

AWS EC2にNPMエージェントをインストールする場合、IAM設定を通じてAWSのリソース情報を収集できます。

ⓘ AWS設定ガイド

AWS IAMの設定

AWS IAMのポリシー設定プロセスで最小権限を指定するように求めています。NPMでAWSリソース情報を収集するに

は、`ec2:DescribeNetworkInterfaces` 権限が必要です。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        //...
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    }
  ]
}
```

❗ ポリシーと権限の詳細については、[次のリンク](#)を参考にしてください。

EKS環境でのAWSリソース権限設定

AWS EKSでNPMエージェントをインストールする場合、ServiceAccountを通じてAWSのリソース情報を収集できません。

❗ AWS設定ガイド

AWS IAMの設定

AWS IAMのポリシー設定プロセスで最小権限を指定するように求めています。NPMでAWSリソース情報を収集するには、`ec2:DescribeNetworkInterfaces` 権限が必要です。

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Effect": "Allow",
  "Action": [
    //...
    "ec2:DescribeNetworkInterfaces"
  ],
  "Resource": "*"
}
```

❗ ポリシーと権限の詳細については、[次のリンク](#)を参考にしてください。

Service Accountの設定

AWS EKSの設定過程では、IAMとKubernetesのService Accountを接続する必要があり、NPMでAWSリソース情報を収集するには、IAMとNPMのService Accountを接続する必要があります。

NPMでは、次のService Accountを使用してください。

```
namespace : whatap-npm-ns
service_account : whatap-npm
```

example

```
kubectl annotate serviceaccount -n whatap-npm-ns whatap-npm eks.amazonaws.com/role-arn=arn:aws:iam::$account_id:role/$role
```

tagRule設定

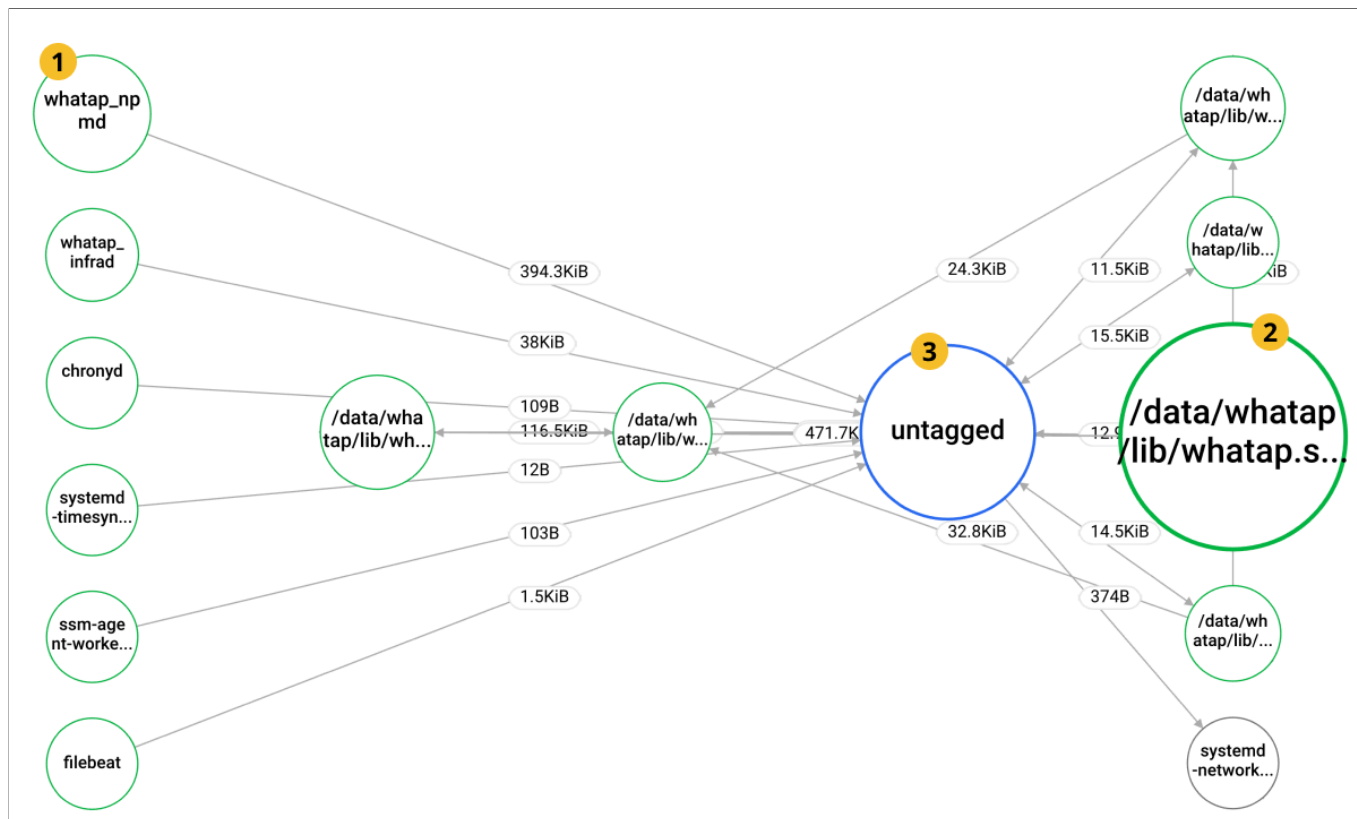
タグ(Tag)は、モニタリング対象サーバーから収集されたデータをグループ単位で表示するための基準になります。**NPM**で最も重要な要素は可視性です。グループ化は、ネットワークという不規則で多様な情報があふれる環境で、可視性を確保するための必須要素です。

エージェントは、Raw DataにIPアドレスおよびPort(Source、Target)、Host Name、Process Name情報をキーとして収集します。収集した情報に基づいて、トポロジーまたはチャートに表示するノードのグループ単位を設定できます。

エージェントから収集するデフォルトタグに加えて、ユーザーが簡単に変更して適用できるルールを作成しました。エージェントインストールパス(</usr/whatap/whatap-npmd>)の[tagRule.yaml](#)ファイルを修正して、オプションを適用します。

tagRuleオプション

tagRuleオプションが適用された結果は、[ダッシュボード](#) > [ネットワークトポロジー](#)メニューで確認できます。グループ化により、複雑で不規則なネットワーク間の関連性を単純化し、抽象化されたネットワークの性能を可視化してモニタリングできます。



- tagRuleオプションが適応された①ノードです。長くて複雑なプロセス名を簡素化してノードに表示します。
- tagRuleオプションが適応されていない②ノードです。どんなプロセスなのか把握しにくいので①ノードのようにtagRuleオプションを適用して、プロセス名を簡素化する必要があります。
- ③ノードは、そのホストおよびプロセス情報を取得できない場合です。詳細については、[次の文書](#)を参照してください。

プロセス名の簡素化

オプション： `processRegex`

名前が不明な長いプロセス名を簡素化します。例えばJava Processの場合、次のようにJVM実行オプションなどを含みます。

zookeeper process name

```
java -Dzookeeper.log.dir=/data/wpm/zookeeper/bin/../../logs -Dzookeeper.log.file=zookeeper-whatap-server-Dev-Modules.log
```

```
-XX:+HeapDumpOnOutOfMemoryError -XX:OnOutOfMemoryError=kill -9 %p -cp /data/wpm/zookeeper/bin/../zookeeper-server/target/classes:/data/wpm/zookeeper/bin/../build/classes:/data/wpm/zookeeper/bin/../zookeeper-server/target/lib/*.jar:/data/wpm/zookeeper/bin/../build/lib/*.jar:/data/wpm/zookeeper/bin/../lib/zookeeper-prometheus-metrics-3.8.0.jar:/data/wpm/zookeeper/bin/../lib/zookeeper-jute-3.8.0.jar:/data/wpm/zookeeper/bin/../lib/zookeeper-3.8.0.jar:/data/wpm/zookeeper/bin/../lib/snappy-java-1.1.7.7.jar:/data/wpm/zookeeper/bin/../lib/slf4j-api-1.7.30.jar:/data/wpm/zookeeper/bin/../lib/simpleclient_servlet-0.9.0.jar:/data/wpm/zookeeper/bin/../lib/simpleclient_hotspot-0.9.0.jar:/data/wpm/zookeeper/bin/../lib/simpleclient_common-0.9.0.jar:/data/wpm/zookeeper/bin/../lib/simpleclient-0.9.0.jar:/data/wpm/zookeeper/bin/../lib/netty-transport-native-unix-common-4.1.73.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-transport-native-epoll-4.1.73.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-transport-classes-epoll-4.1.73.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-transport-4.1.73.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-tcnative-classes-2.0.48.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-tcnative-2.0.48.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-resolver-4.1.73.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-handler-4.1.73.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-common-4.1.73.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-codec-4.1.73.Final.jar:/data/wpm/zookeeper/bin/../lib/netty-buffer-4.1.73.Final.jar:/data/wpm/zookeeper/bin/../lib/metrics-core-4.1.12.1.jar:/data/wpm/zookeeper/bin/../lib/logback-core-1.2.10.jar:/data/wpm/zookeeper/bin/../lib/logback-classic-1.2.10.jar:/data/wpm/zookeeper/bin/../lib/jline-2.14.6.jar:/data/wpm/zookeeper/bin/../lib/jetty-util-ajax-9.4.43.v20210629.jar:/data/wpm/zookeeper/bin/../lib/jetty-util-9.4.43.v20210629.jar:/data/wpm/zookeeper/bin/../lib/jetty-servlet-9.4.43.v20210629.jar:/data/wpm/zookeeper/bin/../lib/jetty-server-9.4.43.v20210629.jar:/data/wpm/zookeeper/bin/../lib/jetty-security-9.4.43.v20210629.jar:/data/wpm/zookeeper/bin/../lib/jetty-io-9.4.43.v20210629.jar:/data/wpm/zookeeper/bin/../lib/jetty-http-9.4.43.v20210629.jar:/data/wpm/zookeeper/bin/../lib/javax.servlet-api-3.1.0.jar:/data/wpm/zookeeper/bin/../lib/jackson-databind-2.13.1.jar:/data/wpm/zookeeper/bin/../lib/jackson-core-2.13.1.jar:/data/wpm/zookeeper/bin/../lib/jackson-annotations-2.13.1.jar:/data/wpm/zookeeper/bin/../lib/commons-io-2.11.0.jar:/data/wpm/zookeeper/bin/../lib/commons-cli-1.4.jar:/data/wpm/zookeeper/bin/../lib/audience-annotations-0.12.0.jar:/data/wpm/zookeeper/bin/../zookeeper-*.jar:/data/wpm/zookeeper/bin/../zookeeper-server/src/main/resources/lib/*.jar:/data/wpm/zookeeper/bin/../conf:/Xmx1000m -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.local.only=false org.apache.zookeeper.server.quorum.QuorumPeerMain /data/wpm/zookeeper/bin/../conf/zoo.cfg
```

上記の場合、トポロジーやチャート上に名前を表示しても把握しにくいいため、エージェントから最大限に必要な部分だけを抽出して使用します。次の例を参照してオプションを適用します。

tagRule.yaml

```
processRegex:
  - zookeeper
  - whatap_[a-zA-Z]*
```

processRegex オプションは、次のルールで名前を抽出します。

- **Java**は実行jarファイル名またはメソッド名
- **Python**は実行スクリプト名

ただし、抽出後もプロセス名が長い場合は、オプションで減らすことができます。正規表現で入力でき、ルールが複数ある場合、最も長くマッチングされた文字列が選択されます。

❗ 短いルールを多数使用する場合は、意図していないプロセスがタギングされます。固有の名前ルールを使用してください。

プロセスのホワイトリスト(White List)

オプション: `processWhiteList`

収集対象のプロセス一覧を設定します。該当オプションを通じて指定されたプロセス以外のデータはすべて無視します。

tagRule.yaml

```
processWhiteList:  
- zookeeper  
- whatap_[a-zA-Z]*
```

❗ `processRegEx` オプションを利用すると、OSの名前よりも正規表現にマッチングされた名前が優先されます。

プロセスのブラックリスト(Black List)

オプション: `processBlackList`

収集しないプロセス一覧を設定します。このオプションを通じて、指定されたプロセスデータを無視します。

`processWhiteList` と `processBlackList` がすべて利用中の場合、`processWhiteList` の一覧の中から `processBlackList` に該当する一覧を除いてデータを収集します。

例えば、`zookeeper`、`whatap_npmd`、`whatap_infrad`の3つのプロセスが動作しているときに、次のオプションを適用すると`zookeeper`と`whatap_infrad`の2つのプロセス情報を収集します。

tagRule.yaml

```
processWhiteList:
- zookeeper
- whatap_[a-zA-Z]*

processBlackList:
- whatap_npm
```

❗ processRegEx オプションを利用すると、OSの名前よりも正規表現にマッチングされた名前が優先されます。

プロセスタイプ(Type)指定

オプション: processType

processType オプションは、トポロジーとチャートに表示される最も基本的なグループであるタグです。主な目的は、同じプロセスがバージョンやOS環境によって少しずつ異なる場合、同じ役割を果たすプロセスらを1つのノードとして表示できます。

プロセス名(Process name)をキーとして使用し、マッチングされるタイプを指定する形で使用します。該当オプションで定義されていないプロセスは、プロセス名(Processname)がプロセスタイプ(Processtype)として自動的に適用されます。

```
processType:
apache:
- httpd
- apache2
mysql:
- mysqld
- mysqld_safe
```

❗ processRegEx オプションを利用すると、OSの名前よりも正規表現にマッチングされた名前が優先されます。

アプリ名(App Name) 指定

オプション: `appName`

`appName` は、トポロジーまたはチャートをユーザーにとって理解しやすい形で表現するためのオプションです。上記の `processType` オプションの場合は、ユーザーの意図とは関係なく、同じプロセスは1つのノードに統合(Merge)されて表示されます。`appName` はキーの組み合わせにより、分離して表示したいノードと合わせたいノードを指定し、ユーザーが希望する形のノードを表示できます。

proxy→msa_process→msa_process2→dbにつながるネットワーク構成が本番機/開発機/検証機の3つの目的に分かれている場合を可視化するケースを想定してみます。

- `processType` オプションで可視化する場合
- `appName` オプションで同じ名称のプロセスらを本番機/開発機/検証機に分離する場合

tagRule.yaml

```
appName:
  prod_proxy: # 本番機
    - host_tag: prod_proxy
      process_type: proxy
  prod_msa1: # 本番機
    - host_tag: prod_msa
      process_type: msa_process1
  prod_msa2: # 本番機
    - host_tag: prod_msa
      process_type: msa_process2
  dev_proxy: # 開発機
    - host_tag: dev_proxy
      process_type: proxy
  dev_msa: # 開発機
    - host_tag: dev_msa
      process_type: msa_process1
    - host_tag: dev_msa
      process_type: msa_process2
  test: # 検証機
    - host_tag: test
  db:
    - process_type: mysql
```

- ❗ `appName` 区分で使用可能なkeyは `host_tag`、`process_type`、`ip`、`port` です。
- 入力されていないKeyはワイルドカードとして適用されます。

`appName` オプションでマッチングされていない場合、`appNameDefault` オプションで定義されたフィールドまたは値が適用されます。

- `appNameDefault` にはデフォルトで使用するkeyを入力します。
- `host_tag`、`process_type` のいずれかを入力すると、そのフィールド情報が `appName` に自動付与され、もしその他のString値を入力すると、その値に固定されます。

tagRule.yaml

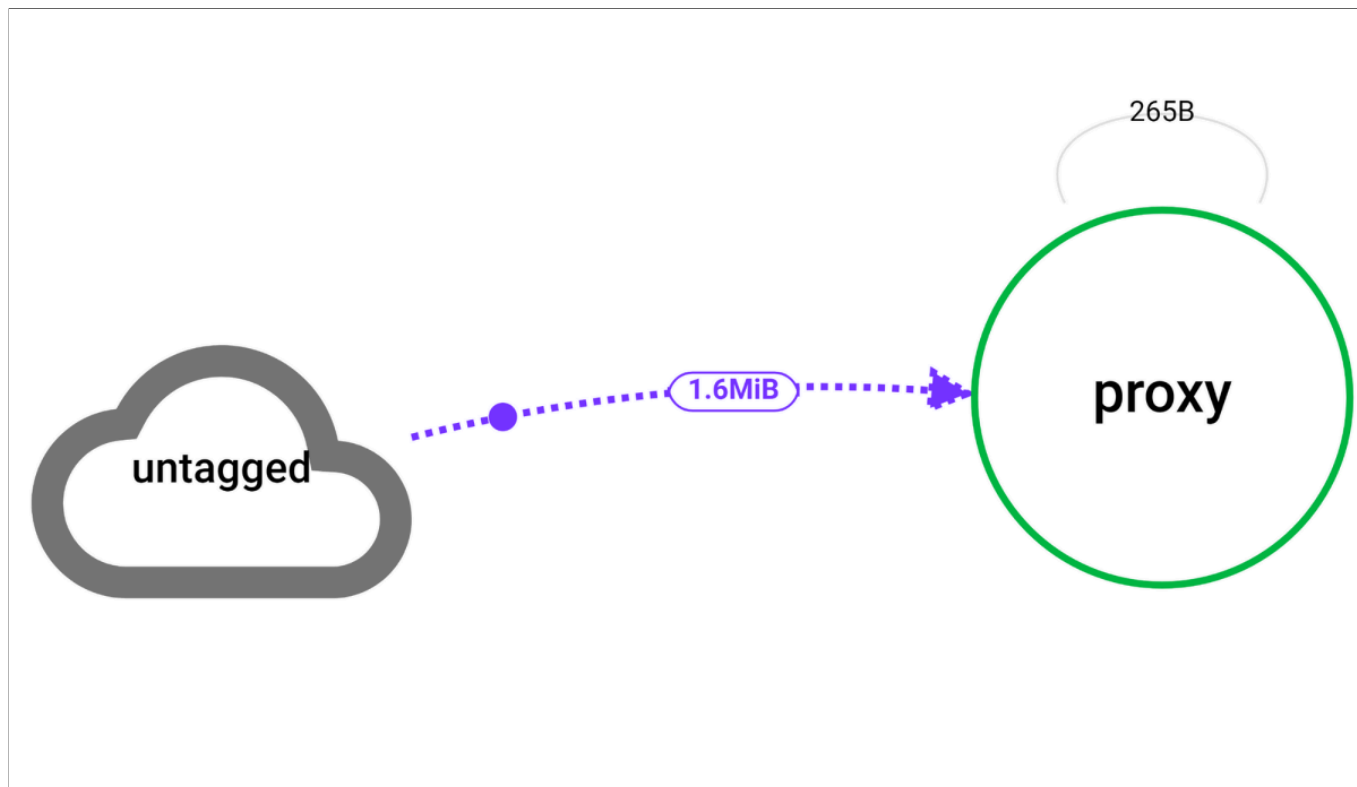
```
appNameDefault: host_tag # host_tag, process_type, custom string
```

不明な対象区分

オプション： `untagOption`

CSP(クラウドサービスプロバイダー)が提供するELB、外部に要請するAPIなどエージェントがインストールされていない対象と通信する場合、そのホストおよびプロセスの情報を取得できません。システム運用者が知っている(または管理対象の)IP、Port情報に基づいてタグを設定できます。

もしエージェントもインストールされず、そのオプションにも定義されていないIP、Portと通信する場合、トポロジーでは**untagged**というノードで表示されます。



IPとPortをキーでタグを設定し、Inboundトラフィックの場合、外部PortがClientポートで固定されたサービスポートではないため、'0'に設定して入力します。

tagRule.yaml

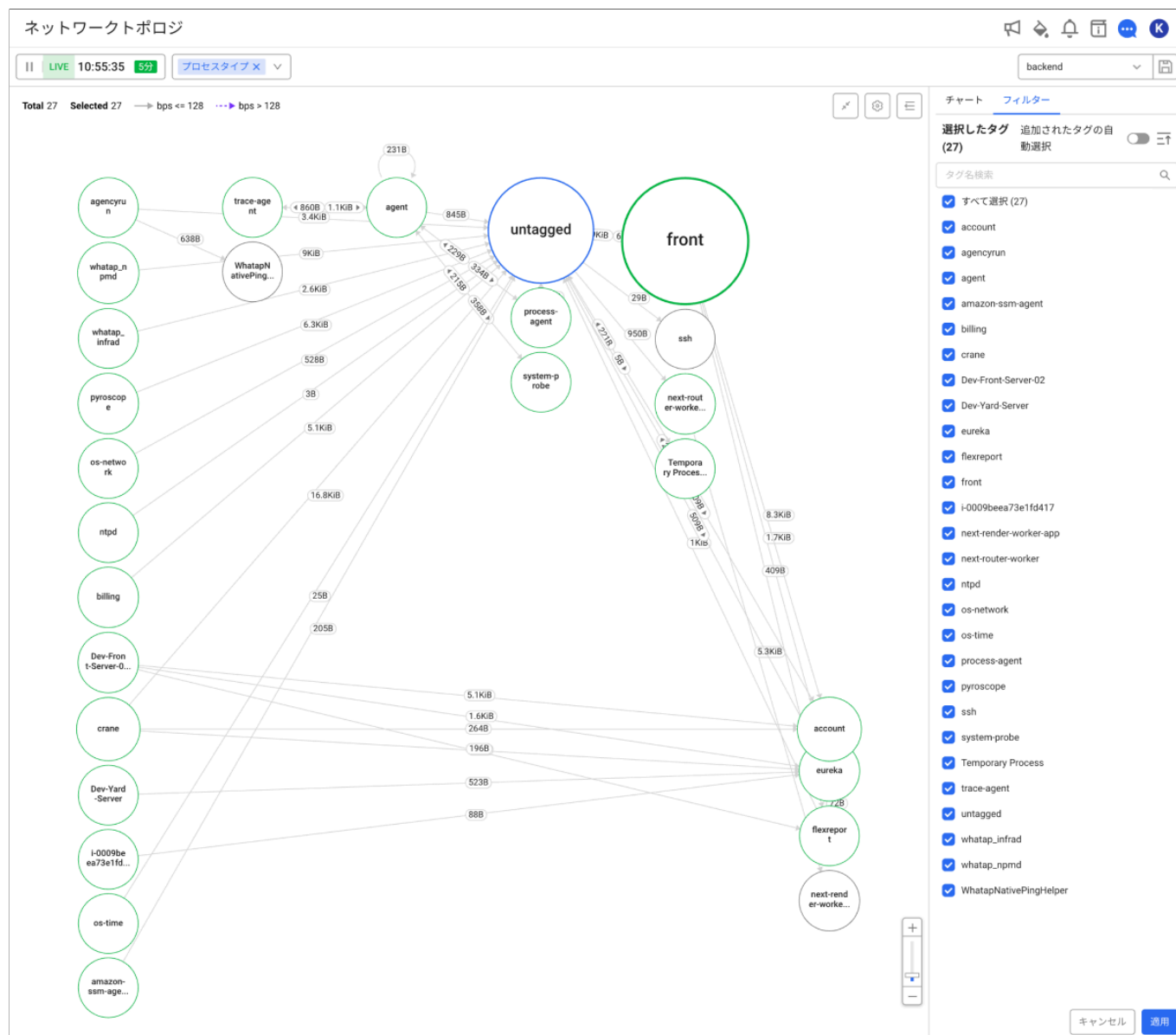
```
untagOption:
  15.165.146.117:
    6600: "whatap-proxy"
  172.31.24.16:
    0: "ELB app/New-gitlab/5fe1a93082df3495"
```

ネットワークトポロジー

ネットワークトポロジ ニューでは、トポロジーチャートを通じてノード間の関係を一目で把握し、指標別線形チャートと連携してネットワーク区間の性能を素早く確認できます。 リアルタイムおよび過去の特定期間のデータを照会することができ、「プロセスの種類」または「アプリケーション」タグに基づいて抽象化されたネットワークの性能を可視化してモニタリングします。

基本画面ガイド

トポロジーはノードとエッジでネットワークをイメージしたチャートです。 各性能指標に対する時系列チャートと並行して、ネットワークのボトルネックまたは障害区間をより簡単に特定できます。



- ・ノードは、モニタリング対象サーバーから収集したデータをグループ単位で表現する基準です。チャートで円形に表現された要素です。
- ・エッジは、ノードとノード間の通信を意味します。詳細については、[次の文書](#)を参照してください。
- ・トポロジーチャートの空欄をマウスでクリックし、ドラッグして希望する位置に画面を移動できます。
- ・マウスホイールをスクロールして、トポロジーチャートを拡大または縮小できます。右下のスライドツールを活用することもできます。
- ・それぞれのノードをマウスでクリックし、希望する位置にドラッグすると移動されます。

- すべてのノードが画面に表示されるようにキャンバス位置をソートするには、画面右上の  ボタンを選択します。タグフィルタリングおよびノードの位置変更を行った後、ノード全体をすばやく確認するのに役立ちます。
- トポロジーチャートの上には、全体ノードの数とタグでフィルタリングされたノードの数、しきい値設定などの要約情報を確認できます。

Total 25 Selected 24 → bps ≤ 128 ...▶ bps > 128

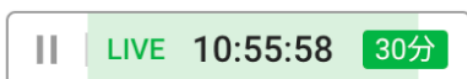
- 右フィルターパネルを非表示にしてトポロジー画面を大きく見るには、チャート右上の  ボタンを選択します。

トポロジーチャートオプション

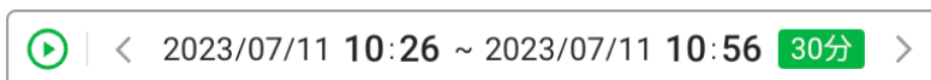
トポロジーチャートで表示する内容を設定して、データを照会および分析できます。

照会時間の設定

- **リアルタイムモード**：緑のボタンに設定された時間のデータをリアルタイムで照会します。

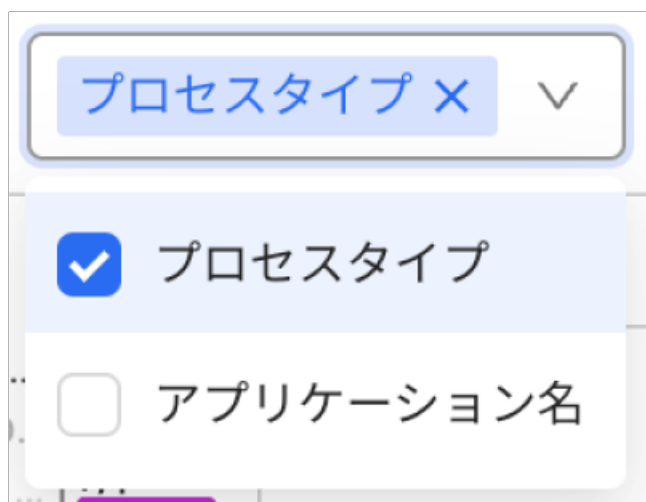


- 緑のボタンを選択して、ご希望の時間を選択できます。
- || ボタンを選択すると、リアルタイムのデータ照会を一時停止し、目的の時間を照会できます。
- **時間設定モード**：ユーザーが希望する時間を設定してデータを照会できます。



- 緑色のボタンを選択すると、様々な時間設定を選択できます。
- < または > ボタンを選択すると、緑のボタンの設定値の分、時間を移動できます。
- 日付と時間テキスト領域を選択すると、日付と時間を選択するオプションが表示されます。
- リアルタイムモードに切り替えるには  ボタンを選択します。

ノードの種類を選択する



チャートに表示されるノードの種類を選択できます。それぞれが1つのタグに対応します。タグは、[tagRule.yaml](#)ファイルに設定したtagRuleオプションに従って収集されます。**プロセスタイプ**、**アプリケーション名** または、2つの組み合わせでノードを表現できます。

ⓘ tagRuleオプションの詳細については、[次の文書](#)を参考にしてください。

チャートの時刻情報の設定

チャートに表示される情報のうち、必要な情報のみ視覚的に表示できます。画面右上の⚙️ボタンを選択します。



基本オプション

- **アニメーション**：しきい値を超えた場合に表示されるアニメーションの表示有無を設定します。
- **数値表示**：エッジのラベル情報の表示有無を設定します。

エッジラベルの設定

エッジは、ノードとノード間の通信を意味し、選択した性能指標をエッジのラベルで表示します。

エッジ

トポロジーチャートのエッジに表示する値の種類を選択できます。指標項目で希望する指標を選択することができます。エッジに表示可能な性能指標の種類は次のとおりです。

- [jitter](#)
- [latency](#)

- [bps](#)(bit per second)
- [pps](#)(packet per second)
- [sessions](#)

- ❗ • **エッジ**項目で選択した指標の種類によって、2番目の項目も同様に変更されます。2番目の項目では、選択した指標のしきい値を設定できます。

指標	jitter ▼
jitter	30
単位	ms ▼

- 指標項目で選択した値に応じて、ノードサイズのウェイトが自動的に設定されます。

しきい値の設定

2番目の項目と単位項目では、エッジとして設定された指標に対するしきい値と単位を設定できます。しきい値を超えるエッジは、トポロジー上で強調表示されます。

別途の設定プロセスがない場合、しきい値と単位は次のようにデフォルト設定されます。

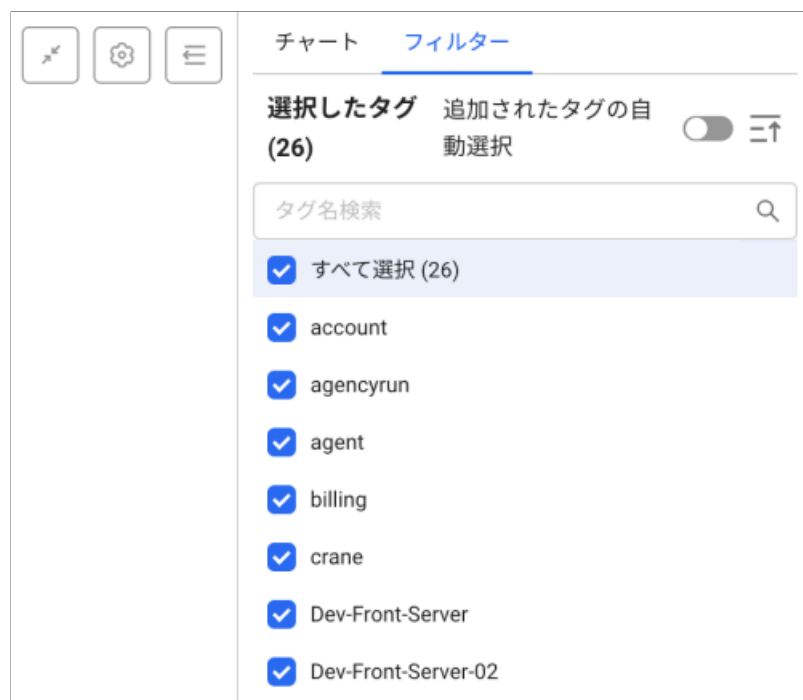
- latency > 200(ms)
- jitter > 30(ms)
- bps > 128(Mib/s)
- pps > 50(K/s)
- sessions > 100

設定された値は、チャートの左上にある要約情報から確認できます。

Total 25 Selected 24 → bps ≤ 128 ...▶ bps > 128

タグフィルタリング

ネットワーク構成が膨大で複雑な場合は、一部のノードのみを選択してサブネットワークビューを構成できます。画面右側のパネルでフィルタータブを選択します。**tagRule**オプションを通じて設定したタグ一覧が表示されたら、画面に表示させるノードを選択します。



選択されたタグノードとそのノード間での通信性能情報のみがトポロジーチャートに表示されます。**untagged**でマークされたノードを除外したり、その他に分析対象外のノードがある場合は、チェックを解除してトポロジーおよび時系列チャートから除外できます。

このように構成した情報は、**プリセット**機能を通じて保存し、読み込みして分析に活用できます。

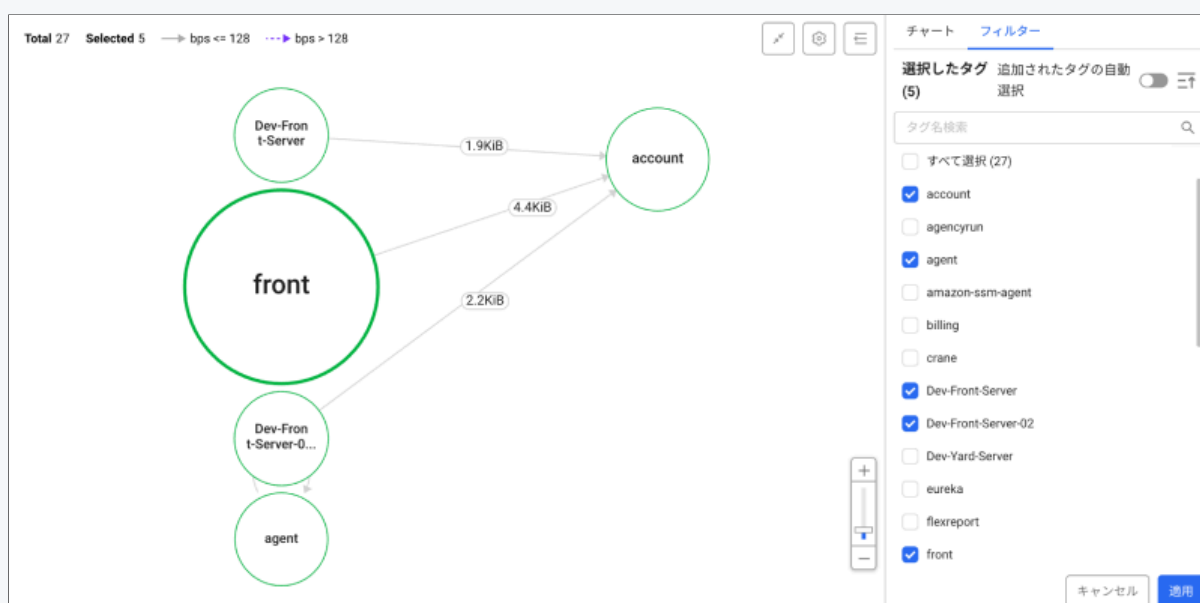
- ⓘ データ照会期間中に実際に収集されたタグのみが一覧に表示されます。
- 新しいノードが追加された場合、自動的に選択するには、**追加されたタグの自動選択**オプションを選択します。

- ❗ ・プリセット機能の詳細については、[次の文書](#)を参考にしてください。

活用例

- ❗ 全体ノードの中で一部のノードのみを表示または除外するには、次の手順を実行します。

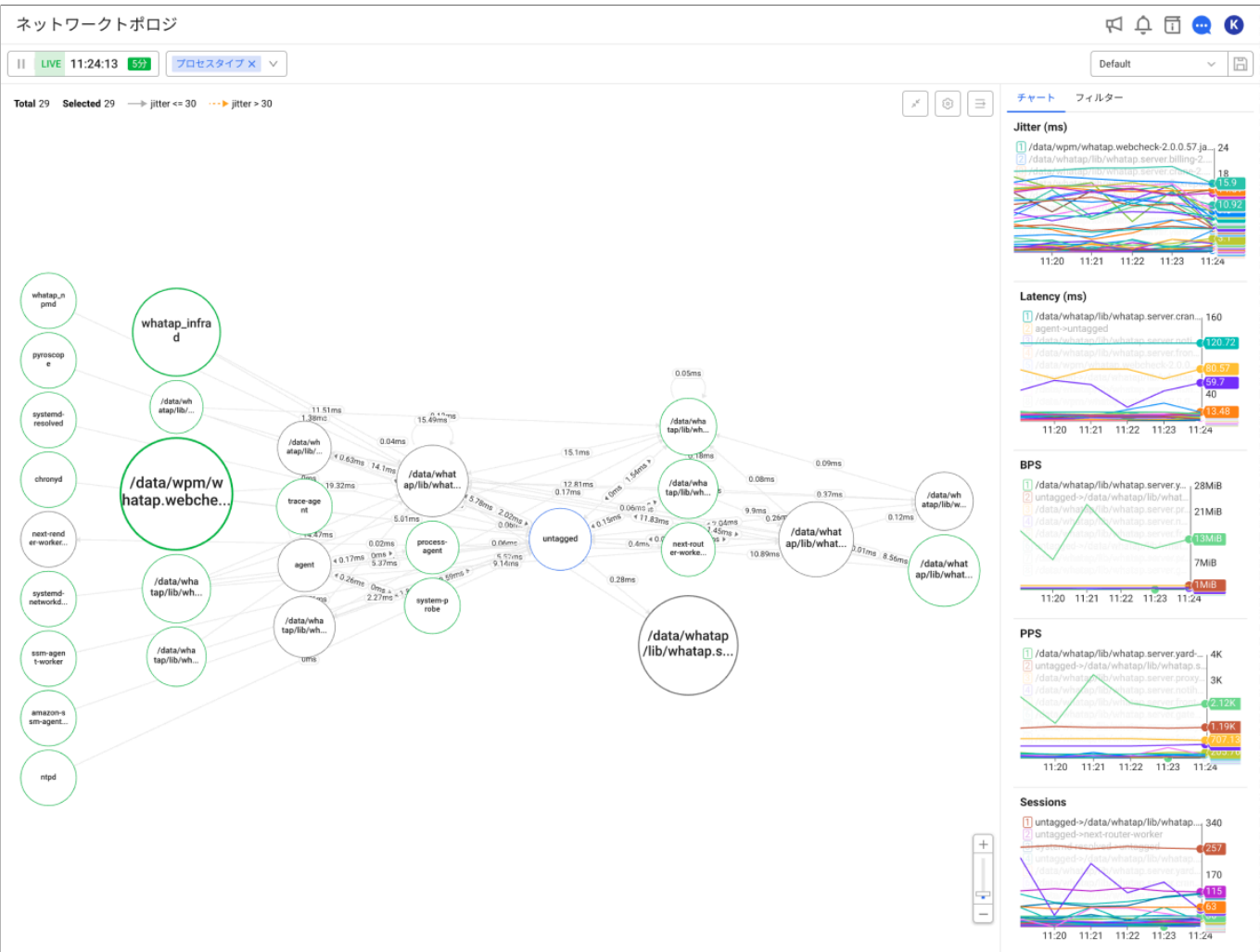
1. 画面右側のパネルで**フィルター**タブを選択します。
2. **すべて選択** オプションを解除して **追加されたタグの自動選択** オプションを非活性化します。
3. トポロジチャートにノードを選択または除外します。タグ名を検索するか、ソート機能(≡↑)を利用すると、ノードを簡単に見つけることができます。
4. 画面の下の**適用**ボタンを選択します。



- ❗ **すべて選択**オプションを選択し、一部のノードの選択を解除すると**すべて選択**は自動的に選択解除されます。

チャートの確認

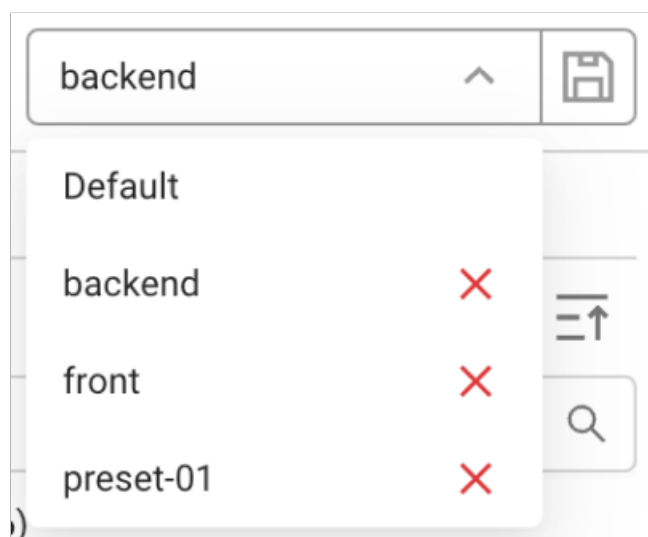
指標別の線形チャートと連携して、ネットワーク区間の性能を迅速に確認できます。右のパネルで**チャートタブ**を選択します。



- **チャート** タブに表示される項目は、**フィルター** タブで選択したタグです。
- チャートの上にマウスオーバーすると、タイムゾーンごとのネットワークの性能を確認できます。

プリセット

プリセット機能を通じて、タグフィルター機能と連携してサブネットワークトポロジービューを保存できます。トポロジービューで表示するノードとエッジが多すぎる場合は、ネットワーク部分を分けてフィルタリングし、プリセットに保存して、分析に活用するのを推奨します。



プリセットに保存される情報は次のとおりです。

- ノード表示の種類
- エッジ表示の種類
- しきい値
- タグフィルター
- トポロジー上の各タグの位置

プリセット作成

1. トポロジーチャートオプションを自由に設定します。
2. 画面の右上に  ボタンを選択します。
3. プリセット保存ウィンドウが表示されたら、保存したいプリセットの名前を入力します。



Default

プリセット保存

現在設定された指標とレイアウトを保存します。

Default

キャンセル 保存

4. プリセット名の入力を完了すると**保存** ボタンを選択します。

プリセット修正

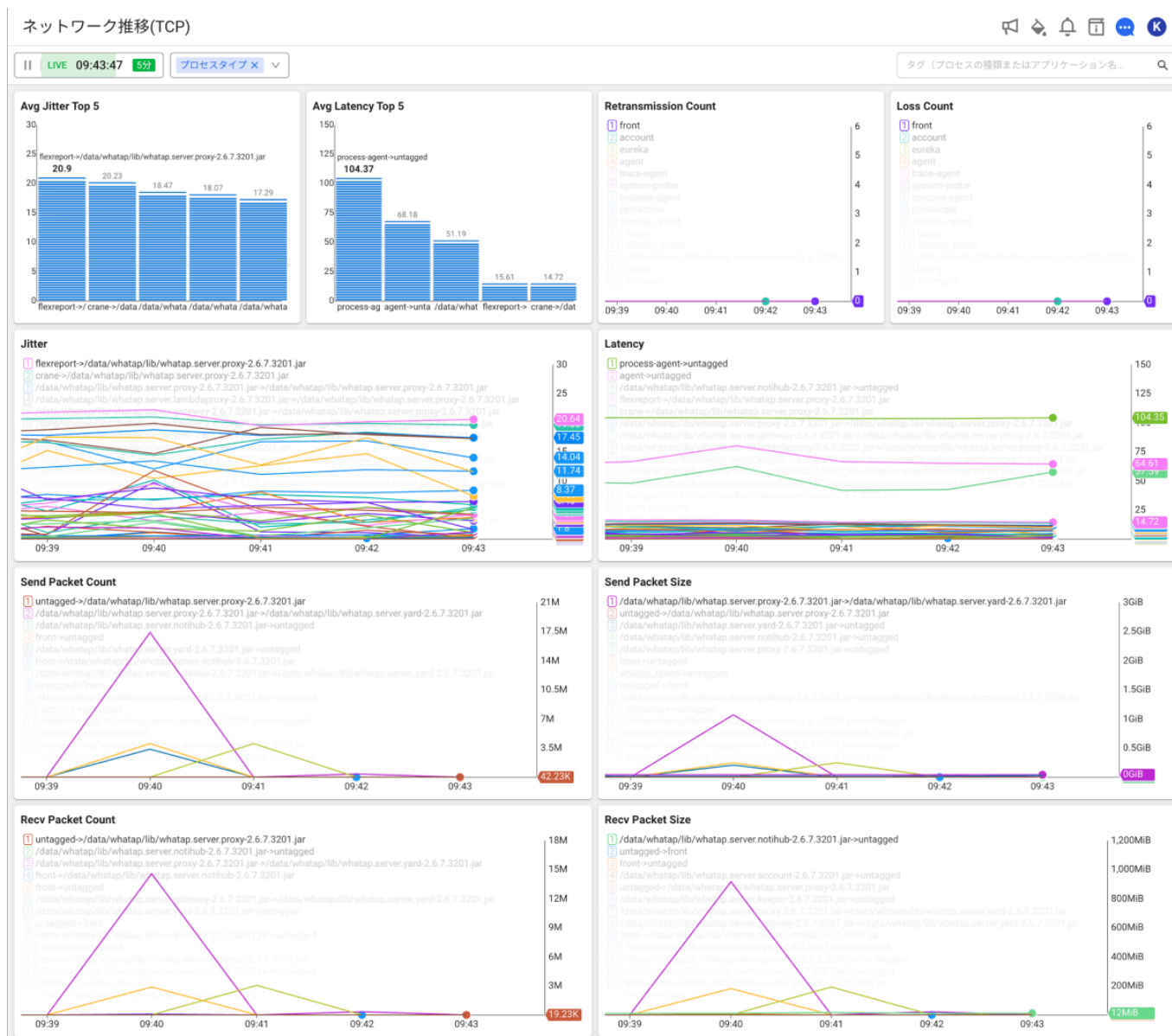
プリセットを選択した状態でトポロジチャートのオプションを変更し、現在のプリセット状態を更新するには、既存のプリセットと同じ名前で保存します。

❗ 別の名前でプリセットを保存することもできます。

プリセット削除

使用しないプリセットを削除するには、プリセット一覧で✖ ボタンを選択します。

ネットワーク推移(TCP)

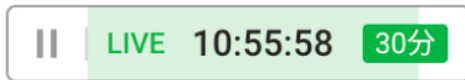


ネットワークトポロジメニューは、ネットワーク構成情報に対する可視化が主な目的であったが、**ネットワーク推移(TCP)**メニューは、数値で表示される性能情報を一つの画面で詳しく確認できます。時間に応じた性能情報をリアルタイムで確認でき、過去時点の性能情報を確認して問題点を分析する資料として活用できます。

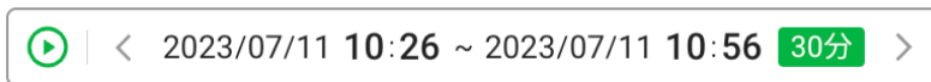
チャートオプションの設定

照会時間の設定

- リアルタイムモード：緑のボタンに設定された時間のデータをリアルタイムで照会します。

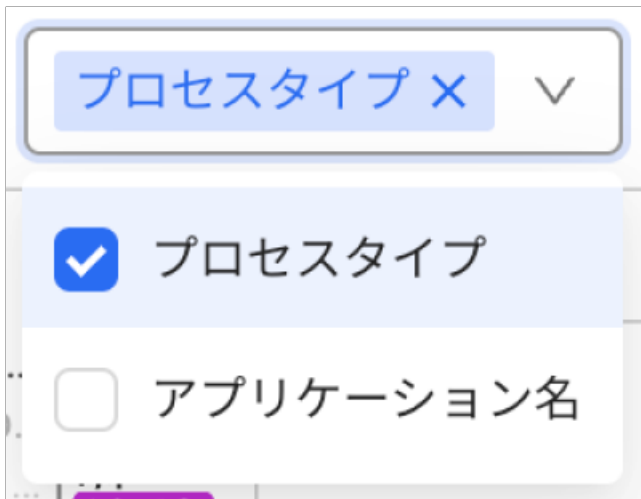


- 緑のボタンを選択して、ご希望の時間を選択できます。
- II ボタンを選択すると、リアルタイムのデータ照会を一時停止し、目的の時間を照会できます。
- 時間設定モード：ユーザーが希望する時間を設定してデータを照会できます。



- 緑色のボタンを選択すると、様々な時間設定を選択できます。
- < または > ボタンを選択すると、緑のボタンの設定値の分、時間を移動できます。
- 日付と時間テキスト領域を選択すると、日付と時間を選択するオプションが表示されます。
- リアルタイムモードに切り替えるには▶ボタンを選択します。

ノードの種類を選択する



チャートに表示されるノードの種類を選択できます。それぞれが1つのタグに対応します。タグは、[tagRule.yaml](#)ファイルに設定したtagRuleオプションに従って収集されます。**プロセスタイプ**、**アプリケーション名** または、2つの組み合わせでノードを表現できます。

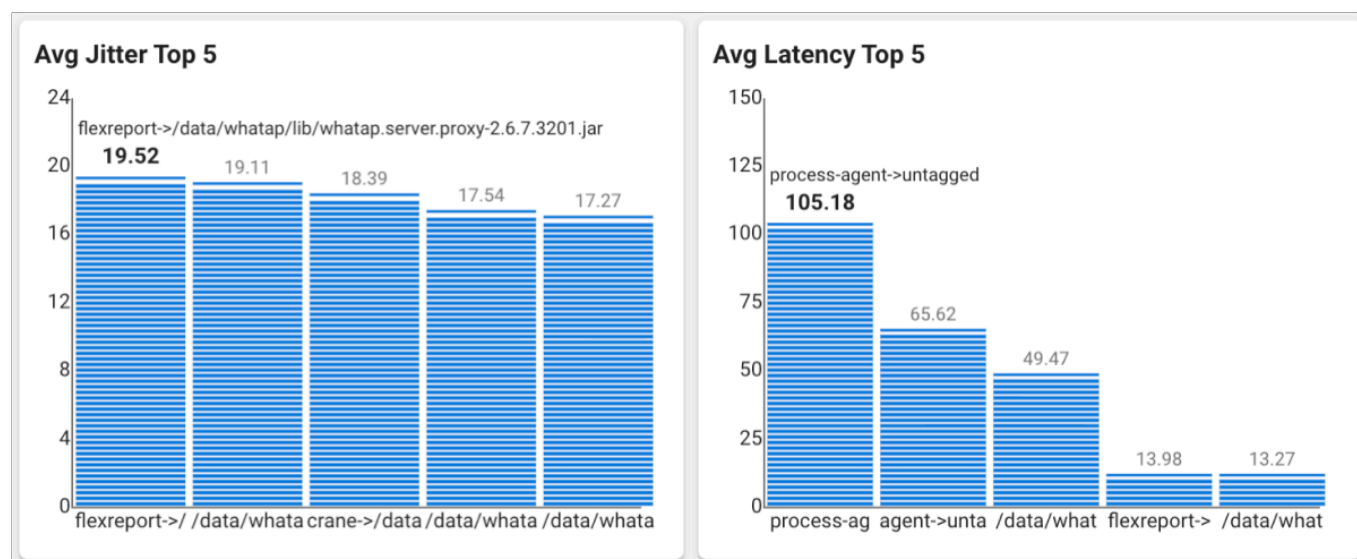
ⓘ tagRuleオプションの詳細については、[次の文書](#)を参考にしてください。

ノード検索

画面右上の検索入力ボックスを通じてチャートに表示されるノードをフィルタリングできます。キーワードにマッチしない項目は、チャート上で表示されません。

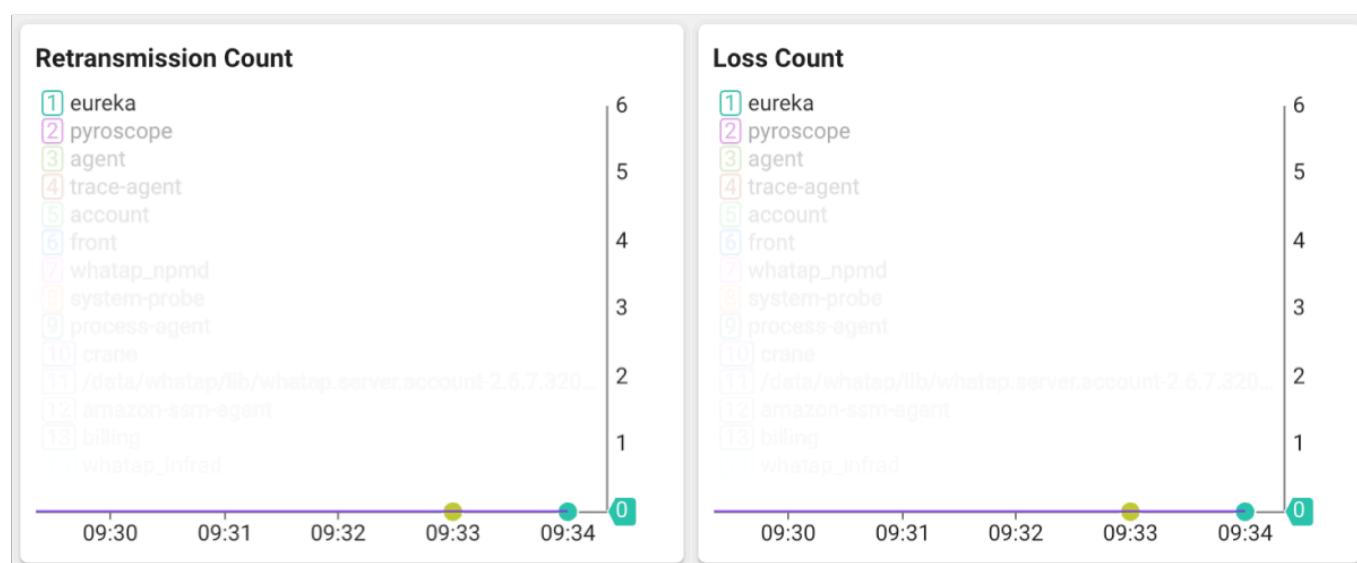
ⓘ この機能は、特定のタグに関連する通信のみを確認する場合に役立ちます。

Top 5チャート



ネットワークの通信障害を意味する主要指標である**jitter**と**latency**を上位5位までをチャートとして表示します。

再送信およびパケット損失指標



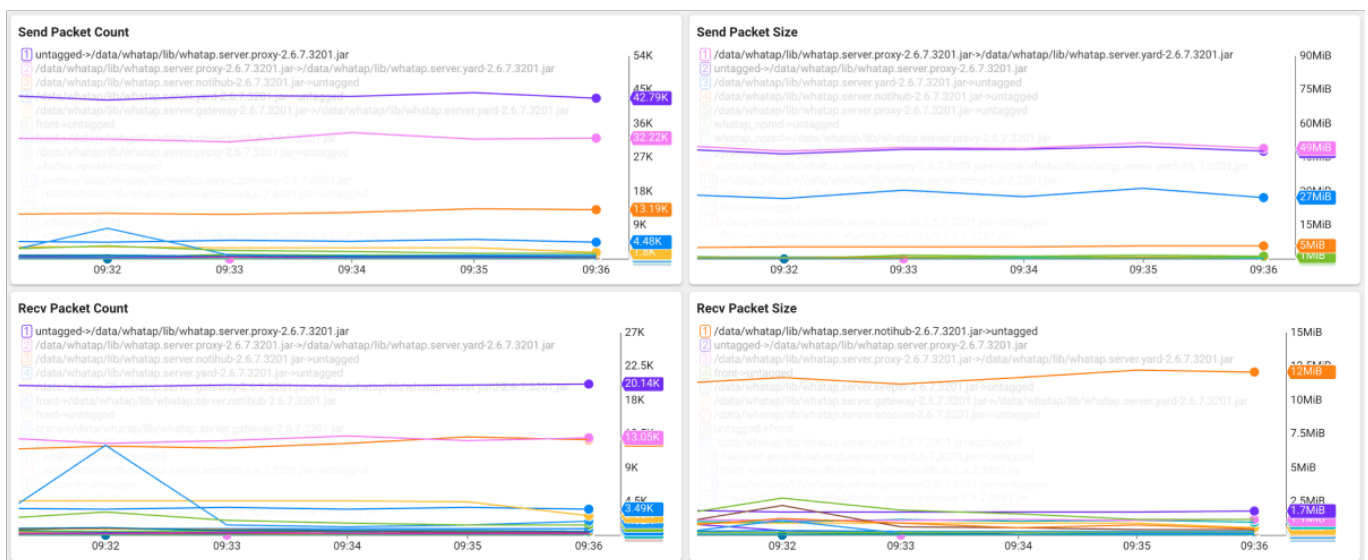
特定の通信区間で再送信が行われた回数をチャートとして表示します。再送信の原因となり得るパケット損失回数と連携して、再送信状況に対するパケット損失有無を確認できます。

遅延率指標



遅延率と遅延率の変化に基づいてネットワークの安定性を評価できます。

データ処理量指標



データ処理量(bpsまたはpps) 指標を送受信の観点から分けてチャートで表示します。

TCPセッション / UDPセッション

エージェントを通じて収集したセッションデータを利用して、障害ポイントを詳細に識別できます。データを効率的に識別できるように、複数の条件を設定できるフィルター機能を提供します。識別したデータを別の資料として活用できるように、csvファイル形式でダウンロードできます。

< 2024/01/05 14:14 ~ 2024/01/05 15:14 60分 >

方向InOut

フィルターを入力してから「Enter」キーを押して追加

🔍

📄 カラム設定

📄 CSV

No	Time	Local Process Type	Local App Name	IP	Port	Foreign IP	Foreign Port	PID	Foreign Process Type
1	24/01/05 15:13:05	front	dev-front-02	10.21.1.187	0	10.21.13.216	6379	3487158	untagged
2	24/01/05 15:13:05	eureka	dev-front-02	10.21.1.187	0	10.21.1.99	6761	3417880	eureka
3	24/01/05 15:13:05	front	dev-front-02	10.21.1.187	0	10.21.1.187	18080	3487158	account
4	24/01/05 15:13:05	process-agent	dev-front-02	10.21.1.187	0	34.160.150.109	443	1875864	untagged
5	24/01/05 15:13:05	front	dev-front-02	10.21.1.187	0	10.21.1.99	6761	3487158	eureka
6	24/01/05 15:13:05	agent	dev-front-02	10.21.1.187	0	35.190.51.116	443	1875863	untagged
7	24/01/05 15:13:05	front	dev-front-02	10.21.1.187	0	75.2.37.199	443	3487158	untagged
8	24/01/05 15:13:05	trace-agent	dev-front-02	127.0.0.1	0	127.0.0.1	5001	1875865	agent
9	24/01/05 15:13:05	agent	dev-front-02	10.21.1.187	0	35.244.255.175	443	1875863	untagged
10	24/01/05 15:13:05	agent	dev-front-02	127.0.0.1	0	127.0.0.1	5001	1875863	agent
11	24/01/05 15:13:05	account	dev-front-02	10.21.1.187	0	10.21.1.187	6761	3559745	eureka
12	24/01/05 15:13:05	process-agent	dev-front-02	127.0.0.1	0	127.0.0.1	5001	1875864	agent
13	24/01/05 15:13:05	eureka	dev-front-02	10.21.1.187	0	15.165.146.117	6600	3417880	proxy
14	24/01/05 15:13:05	eureka	dev-front-02	10.21.1.187	0	10.21.1.187	6761	3417880	eureka
15	24/01/05 15:13:05	front	dev-front-02	10.21.1.187	0	10.21.1.187	6761	3487158	eureka
16	24/01/05 15:13:05	system-probe	dev-front-02	127.0.0.1	0	127.0.0.1	5001	1875862	agent
17	24/01/05 15:13:05	front	dev-front-02	10.21.1.187	0	10.21.1.99	18080	3487158	account
18	24/01/05 15:13:05	pyroscope	dev-front-02	10.21.1.187	0	10.20.1.185	4040	1057434	untagged
19	24/01/05 15:13:05	front	dev-front-02	10.21.1.187	0	15.165.146.117	6600	3487158	proxy
20	24/01/05 15:13:05	whatap_npmd	dev-front-02	10.21.1.187	0	3.39.176.158	6600	3564600	untagged
21	24/01/05 15:13:05	account	dev-front-02	10.21.1.187	0	15.165.146.117	6600	3559745	proxy
22	24/01/05 15:13:05	account	dev-front-02	10.21.1.187	0	10.21.1.183	3306	3559745	untagged
23	24/01/05 15:13:05	front	dev-front-02	10.21.1.187	0	10.21.1.55	8800	3487158	gateway

①時間を設定してから②🔍ボタンを選択します。テーブルに、ユーザーが設定した条件と一致する条件のデータが表示されます。

- ① データ照会条件を新たに設定した後は、🔍ボタンを選択し、新たにマッチングされるデータを出力します。
- この機能はRawデータを活用します。エージェント1台に対し、1分あたり最大5件を提供します。

照会時間の設定

ユーザーが希望する時間を設定してデータを照会できます。

< 2023/10/12 09:03 ~ 2023/10/12 10:03 60分 >

- 緑のボタンを選択して時間設定を選択できます。最大1時間まで選択できます。
- < または > ボタンを選択すると、緑のボタンの設定値の分、時間を移動できます。
- 日付と時間テキスト領域を選択すると、日付と時間を選択するオプションが表示されます。

セッションデータの照会設定

インバウンド / アウトバウンド条件設定

方向

In Out

検索条件を設定する前に、次のトラフィックの種類を選択します。

- In: インバウンドトラフィック
- Out: アウトバウンドトラフィック

フィルター条件の設定

タグ値とフィールド値に対するマルチフィルター条件を設定してデータを照会できます。追加される条件は、&& (and) 条件に設定されます。例えば、ProcessType タグ値が'agencyrun'で Session Count フィールド値が2より大きいセッション データを照会するには、次のように設定してください。

▼ Local Process Type == agencyrun × Session Count > 2 × フィルターを入力してから「Enter」キーを押して追加
 Q

- ✓ 次のようにタグまたはフィールド値を重複設定すると、|| (or) 条件としてフィルタリングできます。

Local Process Type == agencyrun, Local Process Type == dev-yard

1. フィルター入力欄を選択すると、選択できるフィルターの一覧を確認できます。

文字列形式(text)と数字形式(numeric)を選択できます。

▼ フィルターを入力してから「Enter」キーを押して追加

Text	
Local IP	text
Local Port	text
Foreign IP	text
Foreign Port	text
Local Process Type	text
Local App Name	text
Foreign Process Type	text
Foreign App Name	text
Numeric	
Send Packet Count	numeric
Send Packet Size	numeric
Receive Packet Count	numeric
Receive Packet Size	numeric
Session Count	numeric

key == value 一致 key != value 除外 key + value 無効なタグです。 ガイド表示

2. フィルターを選択した後 **オペレーター**(Operator)を選択します。

文字列形式と数字形式によって選択できる演算子は異なる場合があります。

- 文字列形式

▼ Local Process Type フィルターを入力してから「Enter」キーを押して追加

Operator	
==	operator
!=	operator

key == value 一致 key != value 除外 key + value 無効なタグです。 ガイド表示

- 数字形式

3. 文字列形式を選択した場合は、選択できる値の一覧が表示されます。希望する値を選択します。数字の形式は自分で入力する必要があります。

4. 希望する値を選択するか、数字を入力します。フィルター項目を追加するには、Enterキーを入力して上記のプロセスを繰り返します。
5. フィルター入力欄の右側の  ボタンを選択します。

演算子の種類

型式(type)	オペレータ	説明
文字 (text) / 数字 (numeric)		選択したフィルターの値と一致する文字列をフィルタリングします。

型式(type)	オペレータ	説明
	<code>!=</code>	選択したフィルターの値を除いた文字列をフィルタリングします。
数字(numeric)	<code>></code>	選択したフィルターの値よりも大きい場合をフィルタリングします。
	<code>>=</code>	選択したフィルターの値よりも大きい、同じ場合をフィルタリングします。
	<code><</code>	選択したフィルターの値よりも小さい場合をフィルタリングします。
	<code><=</code>	選択したフィルターの値よりも小さい、同じ場合をフィルタリングします。

付加機能

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。
カラム設定ボタンを選択します。

カラム設定

表示するカラムを選択してください。

1

すべて選択 初期化

- ☒ Avg Jitter
- ☒ Avg Latency
- ☒ Foreign IP
- ☒ Foreign Port
- ☒ IP
- ☒ Lost Count
- ☐ Max Jitter
- ☐ Max Latency
- ☒ PID
- ☒ Port
- ☒ Recv Bytes
- ☒ Recv Packet
- ☒ Retransmissions
- ☒ Send Bytes
- ☒ Send Packet
- ☒ Session

3

カラム名で検索

表示項目(20) 2

- No
- Time YY/MM/DD 00:00:00
- Local Process Type
- Local App Name
- IP
- Port
- Foreign IP
- Foreign Port
- PID
- Foreign Process Type
- Foreign App Name
- Session
- Retransmissions
- Lost Count

キャンセル 確認

- 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。

カラムの追加

- 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。

照会データのダウンロード

テーブルに出力されたデータをCSVファイル形式でダウンロードするには、 CSV ボタンを選択します。

メトリクス

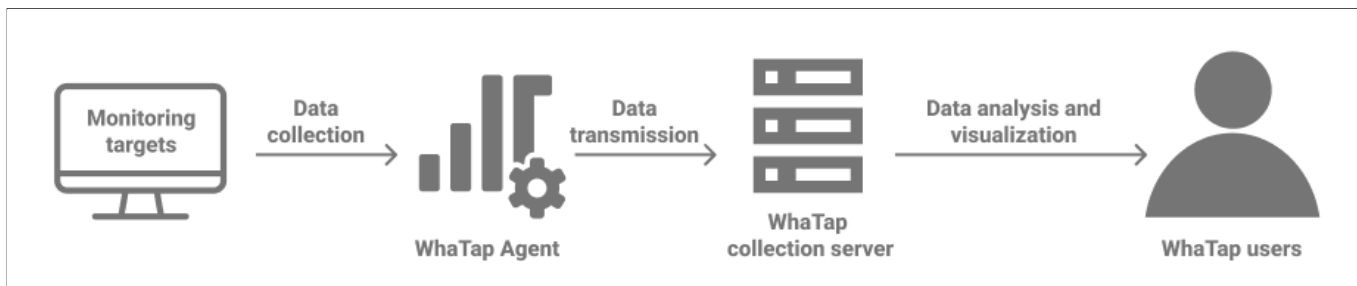
メトリクスとは?

メトリクスはモニタリング対象の状態を数値で表現した時系列データです。WhaTapエージェントがモニタリング対象から収集した性能データを加工した結果です。

メトリクスを通じてサーバー別メモリ使用率、DB接続時間などの主要な性能指標を一目で把握できます。問題を発見した後はログとトレースを通じて詳細分析ができます。

リソース使用量の統計を基に必要なリソース量を算定でき、コスト効率化にも活用できます。

WhaTapのメトリクス収集方式



メトリクスを収集するにはモニタリング対象にWhaTapエージェントをインストールする必要があります。エージェントが送信したメトリクスはWhaTap収集サーバーに保存されます。

WhaTapのメトリクス構成要素

WhaTapのメトリクスは以下の要素で構成されています。

- **Category:** 関連メトリクスを1つのグループにまとめる名前です。
- **Tags:** 収集対象を識別するラベルです。IP、Oname、Hostなど変更の少ない固有情報を保存します。1つのメトリクスに複数のタグが含まれることがあります。

- **Fields:** 実際の測定値を保存します。1つのメトリクスに複数のフィールドが含まれることがあります。
- **Time:** メトリクスを収集した時間です。
- **Oid:** メトリクスを収集したエージェントの固有番号です。
- **Oname:** メトリクスを収集したエージェントの名前です。

メトリクスデータの照会と可視化

WhaTapはユーザーが指定した条件に従って収集した原本データを照会できる**メトリクス照会メニュー**を提供します。

ネットワーク性能指標

ネットワーク性能モニタリングで収集しているメトリクス指標の説明です。 **サイトマップ>メトリクス照会**メニューで検索できます。 ネットワーク性能指標は、すべての情報をフィールド(Field)にフォールディングされた形で提供します。

npm_process_tag_data

セッションごとのネットワーク通信指標を意味します。 収集間隔は1分です。

フィールド名	単位	説明	備考
app_name	-	通信主体に対するカスタム識別タグ	既定値 - 単一のLinuxサーバー環境：ホスト名 - Kubernetesコンテナ環境：Pod名
avg_jitter	ms	パケット往復遅延時間の変動値の平均	収集間隔の平均jitter
avg_latency	ms	パケット往復遅延時間の平均	収集間隔の平均latency
container_name	-	Kubernetes環境で収集されたコンテナ名	Kubernetes環境でなければ空値
direction	-	セッションの連結方向	IN (Inbound)、 OUT (Outbound)
foreign_container_id	-	Kubernetesコンテナ内にローカル通信の場合、当該コンテナのid	ローカル通信の場合、IPアドレス、ポート番号では対象を知ることができず、正確な対象を探すために追加情報を提供
foreign_ip	-	通信対象のIPアドレス	-
foreign_pod_id	-	Kubernetes Pod内にローカ	ローカル通信の場合、IPアドレス、ポート番号では

フィールド名	単位	説明	備考
		ル通信の場合、当該Podのid	対象を知ることができず、正確な対象を探すために追加情報を提供
foreign_port	-	通信対象のポート番号	Client Port(Inbound - foreign_port)は、0でデータが併合されて表記
host_tag	-	通信主体のホスト名	-
local_ip	-	通信主体のIPアドレス	-
local_port	-	通信主体のポート番号	Client Port(Outbound - local_port)は0でデータが併合されて表記
lost_count	count	失われたパケットの数量	-
max_jitter	ms	パケット往復遅延時間の変動値の最大値	一定周期で収集されたjitterの中で最大値
max_latency	ms	パケット往復遅延時間の最大値	一定周期で収集されたlatencyの中で最大値
min_jitter	ms	パケット往復遅延時間の変動値の最小値	一定周期の間に収集されたjitterの中で最小値
min_latency	ms	パケット往復遅延時間の最小値	一定周期の間に収集されたlatencyの最小値
pod_name	-	通信主体がKubernetes環境であれば、Pod名	Kubernetes環境でなければ空値
process_type	-	通信を行ったプロセスの種類	-
protocol	-	通信プロトコルの種類	TCP / UDP
recv_packet_count	count	受信したパケットの数量	-

フィールド名	単位	説明	備考
recv_packet_size	バイト	受信したパケットのデータサイズ	-
retransmission_count	count	再送信が起きたパケット数量	-
row_count	count	重複を含むセッション数	-
send_packet_count	count	送信したパケットの数量	-
send_packet_size	バイト	送信したパケットの全体サイズ	-
session_count	count	重複を除いたセッション数	ユニークセッションの数は、単純合算ではなくHyperLogLog ⁽¹⁾ で算出
session_count_lobits	-	ユニークセッション計算のためのbit情報 (HyperLogLog)	統計処理のためのHyperLogLog関連情報です。
tot_jitter	ms	パケット往復遅延時間の変動値の合計値	-
tot_latency	ms	パケット往復遅延時間の合計値	-

① (1) : HyperLogLogは、非常に少ないメモリ内の要素の数を推定する確率的なデータ構造を意味します。

npm_process_tag_list

セッションのForeign領域タギング処理のための内部データです。

- セッションごとのタグ一覧
- 収集周期：1分

フィールド名	説明	備考
foreign_app_name	通信対象のカスタム識別値	既定値 - 単一のLinuxサーバー環境：ホスト名 - Kubernetesコンテナ環境：Pod名
foreign_container_id	Kubernetesコンテナ内にローカル通信の場合、当該コンテナのid	-
foreign_container_name	通信対象がKubernetes環境であればコンテナ名	-
foreign_host_tag	通信対象のホスト名	-
foreign_ip	通信対象のIPアドレス	-
foreign_pod_id	Kubernetes Pod内にローカル通信の場合、当該Podのid	-
foreign_pod_name	通信対象がKubernetes環境であれば、Pod名	-
foreign_port	通信対象のポート番号	-
foreign_process_type	通信対象のプロセス名	識別できなければ <code>untagged</code>
local_inbound_check	内部ネットワークで発生したInbound通信有無の確認	-

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

1

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

2

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 resp15 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

3

	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。

❗ ネットワーク性能指標の中で npm_process_tag_data 、 npm_process_tag_list カテゴリは**タグ**アイテムを提供していません。

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。
- ネットワーク性能指標の中で npm_process_tag_data 、 npm_process_tag_list カテゴリは**タグ**アイテムを提供していません。

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

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。

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。

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



JSON フィールド	説明
└ 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
ProcessPrivateMemory	PrivateMemory	count	field	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いすぎる、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`)

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

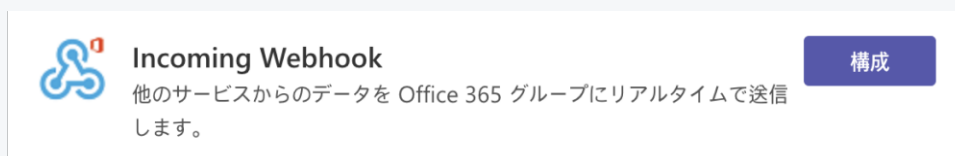
- `/help` : コマンド一覧を表示
- `/license` : 通知を受信するプロジェクトのアクセスキーを入力
- `/remove` : 通知を受信しないプロジェクトコードを入力
- `/list` : 現在通知を受信しているプロジェクトコードを表示

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。

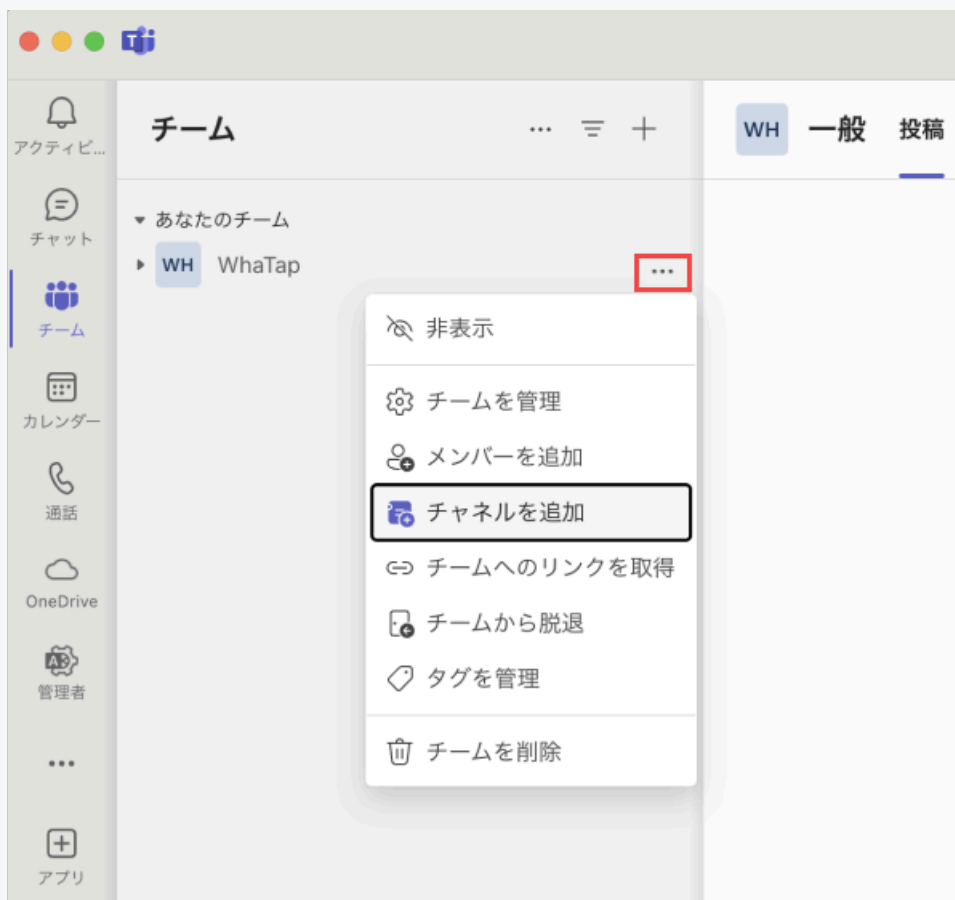
6. [登録] ボタンをクリックします。

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。

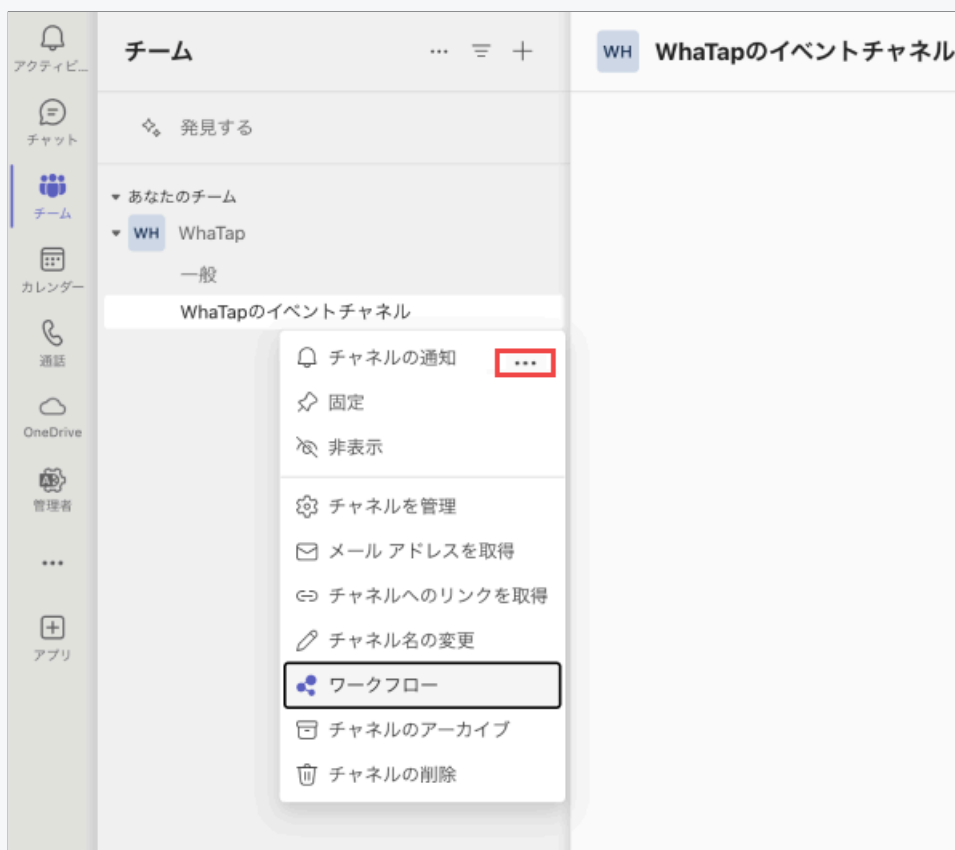
☐ 共有
組織内または組織外の人またはチームがアクセスできます。

☐ 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。

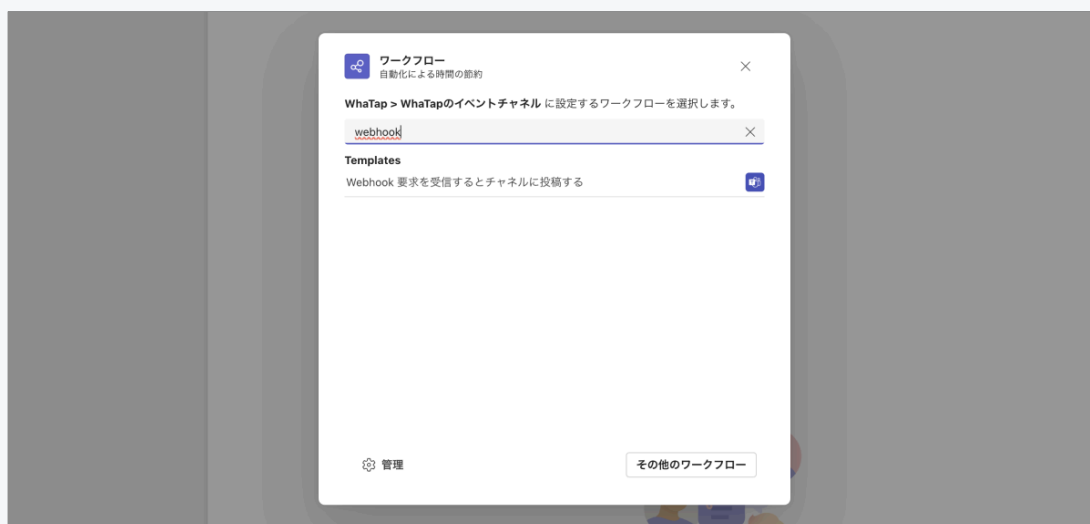
キャンセル 作成

❗ 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。



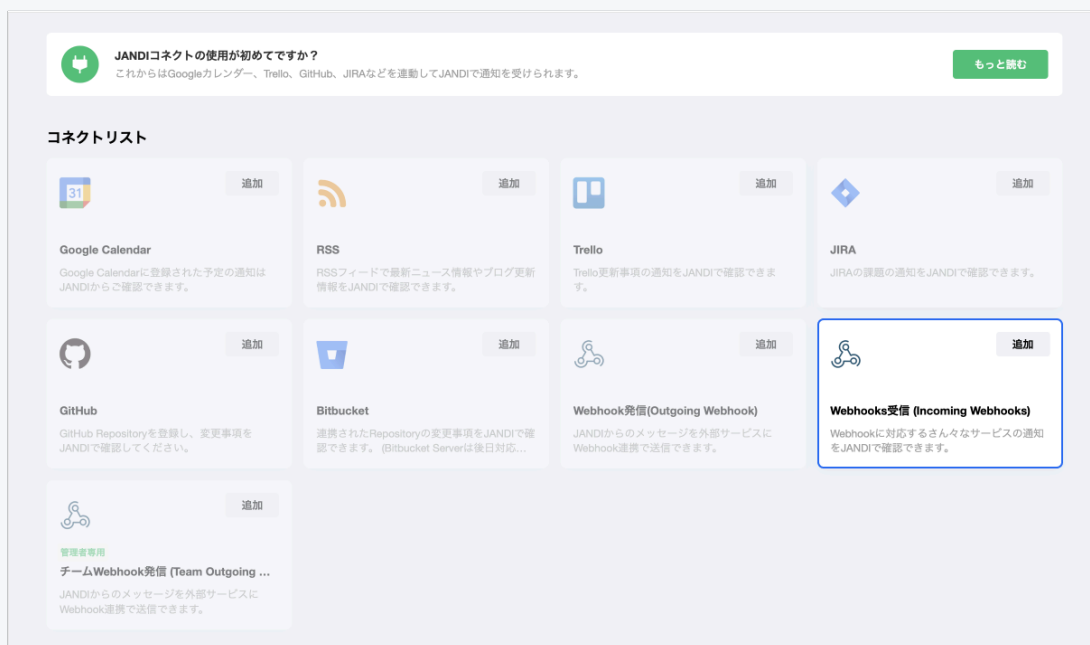
8. WhaTap モニタリングサービスの **サードパーティプラグインの追加** 画面に戻り、Teams **Workflows** タブ下部の **Channel Name** と **Webhook URL** に値を入力します。
9. [登録] ボタンをクリックします。

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}"
}
```

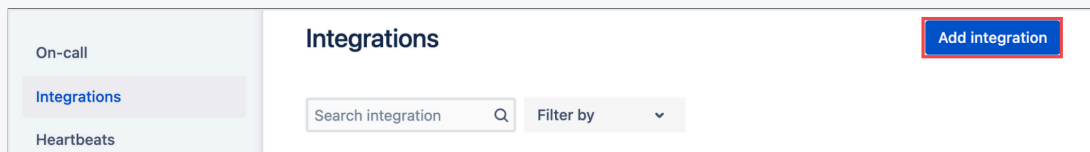
2. **Webhook Name** と **Webhook URL** の項目を入力してください。
3. ヘッダー値を追加するには、**ユーザーヘッダー値の追加** の + 追加 をクリックし、key と value を入力してください。
4. [登録] ボタンをクリックしてください。

Opsgenie

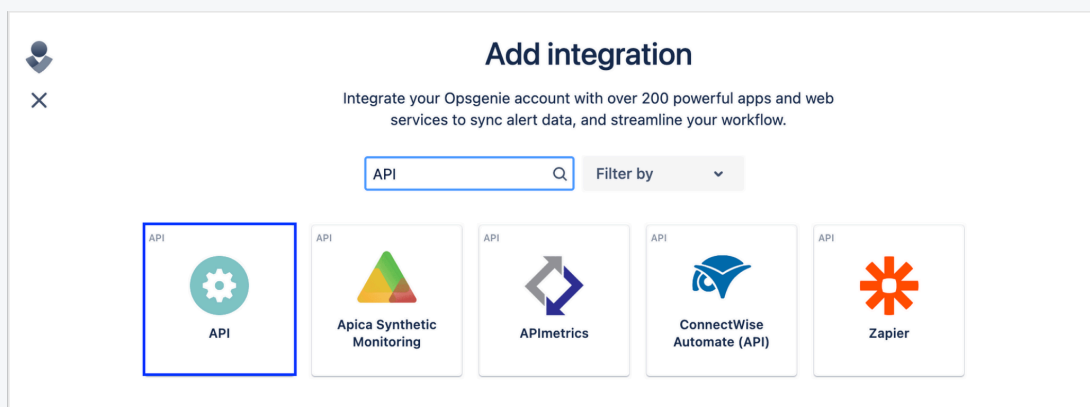




1. **Teams** メニューで、アラートを受信するチームを選択してください。
2. **Integration** メニューに移動し、[Add Integration] ボタンをクリックしてください。



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



- 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。
- 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。
- 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。
- 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。
- 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。
- 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	"zf3uoj0fv4v7"

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。

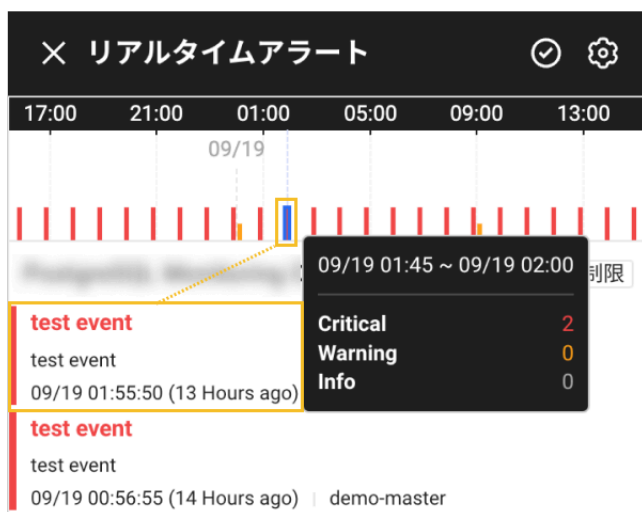
リアルタイムアラートを確認する

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。

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。

メトリクス照会

フィールドとタグに基づいて、カテゴリのごとに特定のメトリクスを照会できます。詳しくは、[次のドキュメント](#)をご参照ください。