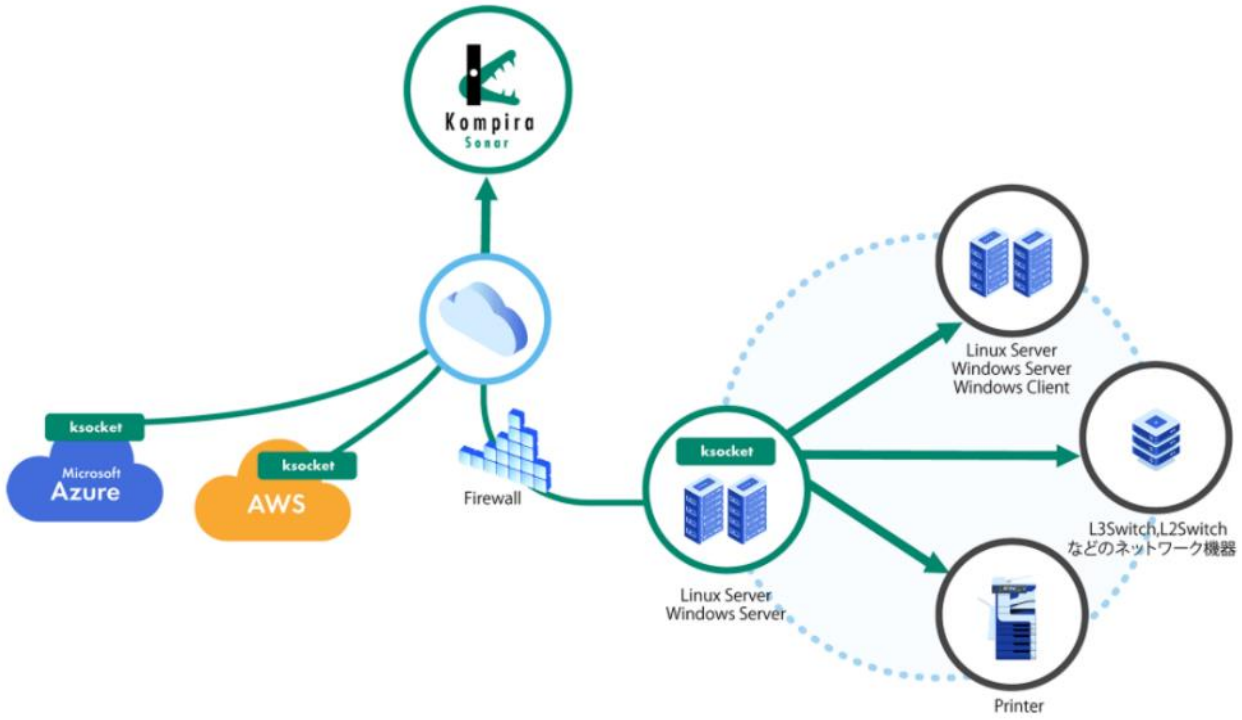
Kompira Sonar　基本マニュアル

もくじ

Kompira Sonar とは

エージェントレスで構成情報を自動収集

Kompira Sonar は、オンプレ/クラウド問わずに構成情報の収集が可能なサービスです。



Kompira Sonar とは

Kompira Sonar活用例

資産管理・構成管理連携

各種ITSMツールや資産管理ツールとKompira Sonar連携する事で、最新の構成情報と資産管理を紐づけることができます。

詳しく見る

監視ツールとの連携による自動設定

Zabbixなどの監視ツールとKompira Sonarを連携することで、構成管理と監視設定を紐づけることができます。

詳しく見る

自社サービスとして構成管理を連携

自社サービスを提供している場合は、構成管理機能を自社サービスの一つとして提供することも可能です。

詳しく見る

用語

Kompira Sonar（以下、Sonar）および本マニュアルで登場する用語について説明します。

用語	説明
スキャン	ネットワーク（同じサブネット内）の特定の時点での構成情報を探索し取得すること
Ksocket	スキャンを行うためにネットワーク内に配置するソフトウェア
スナップショット	スキャンで取得した特定の時点での構成情報
ノード	スナップショットで得られた機器の情報を集約したもの
新規ノード	7日以内に新たに検知されたノード
既存ノード	新規ノードに該当せず消失していないノード
消失ノード	最後に検知されてから30日以上経過したノード
許可アドレス・ネットワーク	スキャンを許可する IPアドレスのリスト
除外アドレス・ネットワーク	スキャンから除外する IPアドレスのリスト
Ksocket トークン	一定期間だけ有効な Kscoket 接続用の認証用文字列
Search query	検索ワードの入力欄

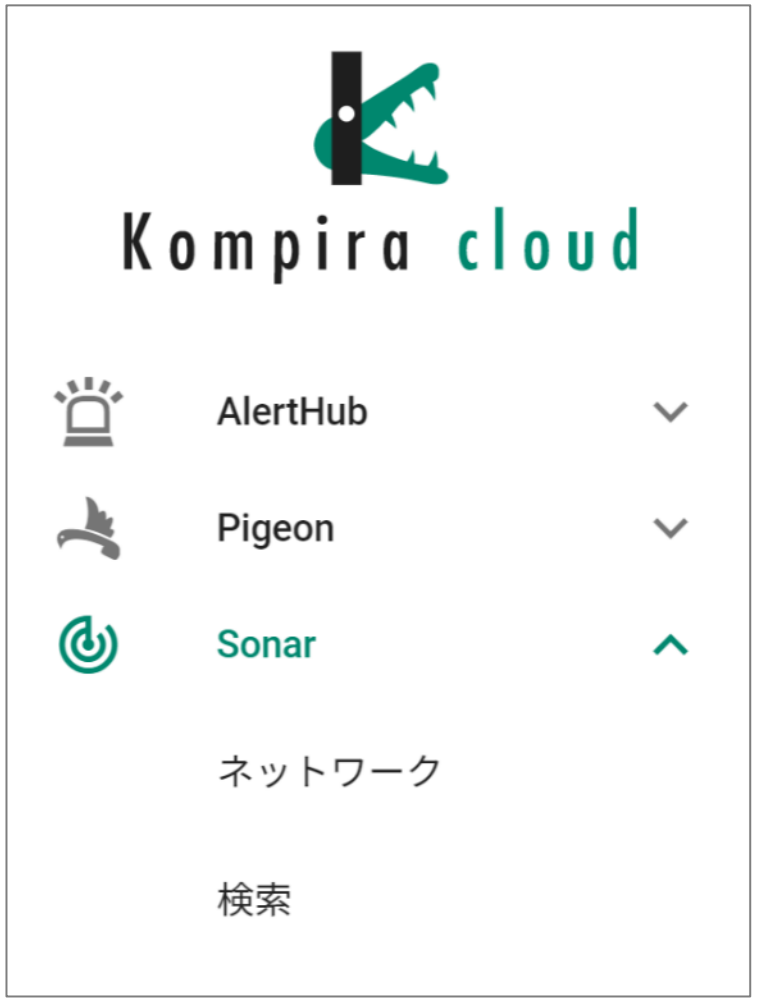
Sonarの機能

Sonarの機能は、

・ネットワーク
・検索

に分かれています。
ネットワークでは Ksocket によってスキャンされた構成情報の確認とスキャンの設定が行えます。
検索ではスキャンで得られたノードの詳細条件による検索が行えます。

スキャンを行うための設定手順



Sonar で構成情報のスキャンを行うためには、以下の流れで設定を行います。

1. **Ksocket のセットアップ**
2. **ネットワークの作成**
3. **スキャンの設定と実行**

設定手順１：Ksocket のセットアップ

1-1 Ksocket の導入要件

Ksocket とはネットワーク上の構成情報をスキャンによって取得するためのソフトウェアです。
スキャンしたいネットワーク内のサーバー上に、最低1台の Ksocket をインストールする必要があります。
インストールを行うサーバーのシステム要件は以下となります。

サポート対象OS

- Red Hat Enterprise Linux 6, 7, 8
- CentOS 6, 7, 8
- Ubuntu 16.04, 18.04
- Windows Server 2012
- Windows Server 2012 R2
- Windows Server 2016

推奨ハードウェア構成

- CPU: 1GHz以上
- メモリ: 1GB以上
- ストレージ: 20GB以上

※ 対象とするネットワーク規模や検出デバイスにより推奨スペックは変動します。

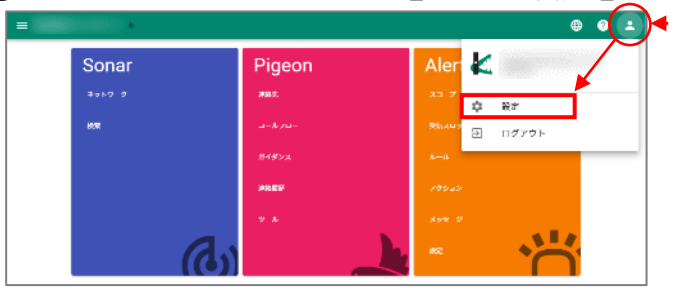
設定手順１：Ksocket のセットアップ

1-2 Ksocket トークンを発行する

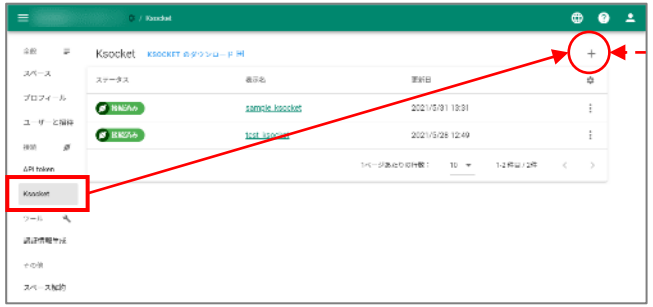
Ksocket と Sonar を接続するにあたって必要な Ksocket トークンを発行します。

Kompira cloud の管理画面にログイン後、以下の作業を実施してください。

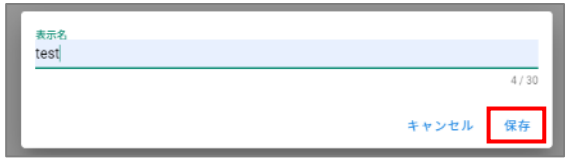
① 『アカウントメニュー』-»『設定』の順にクリック



② 『Ksocket』-»『 ⊕（新規作成ボタン）』の順にクリック



③ 任意の表示名を入力 -»『保存』をクリック

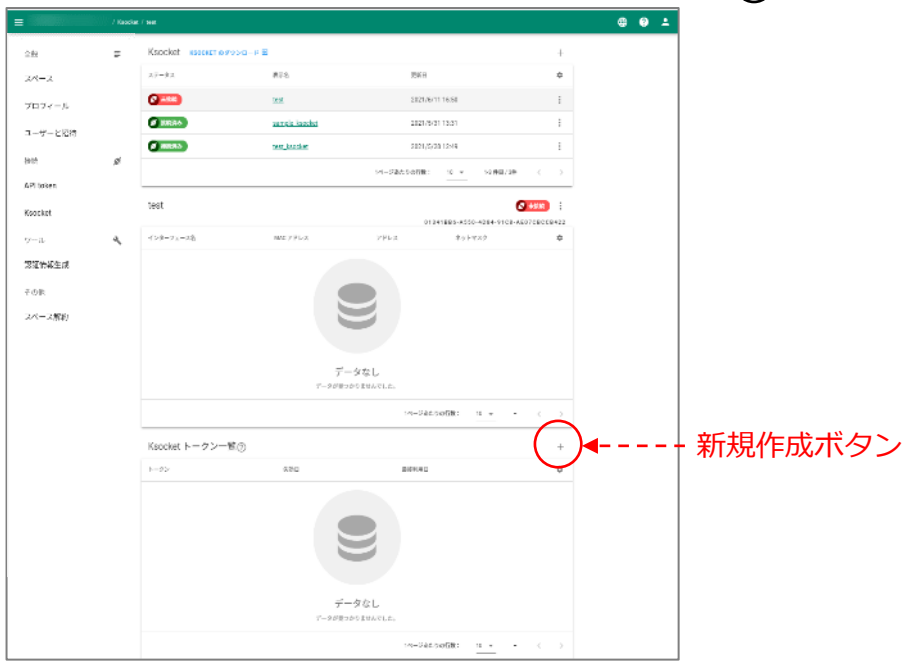


この段階では、後述する「1-3 サーバーへ Ksocket をインストールする」でサーバーにインストールする Ksocket の表示名を設定しています。
分かりやすい表示名を入力してください。

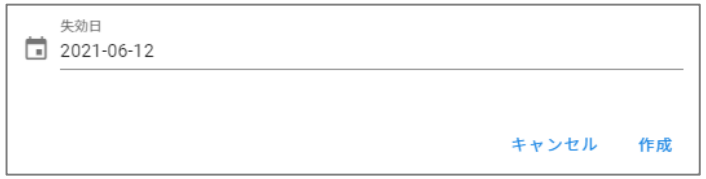
設定手順１：Ksocket のセットアップ

1-2 Ksocket トークンを発行する

④ 「Ksocket トークン一覧」の右側にある『⊕（新規作成ボタン）』をクリック



⑤ トークンの失効日を入力し『作成』をクリック

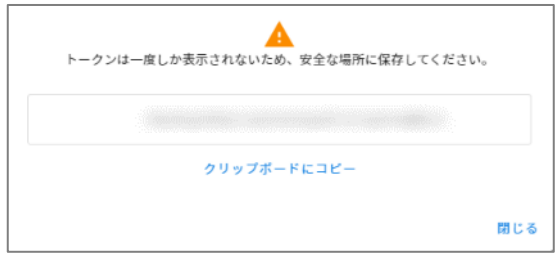


トークンの失効日は、Ksocket を使ってスキャンできなくなる日を指しています。失効日を超えると新しいトークンの発行が必要になりますので、必要な長さを持つ日付を指定してください。

設定手順１：Ksocket のセットアップ

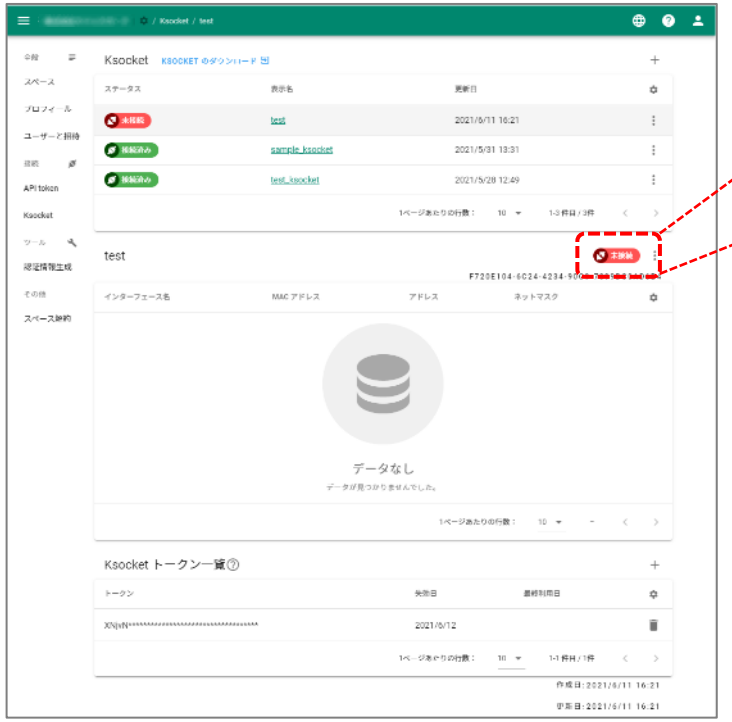
1-2 Ksocket トークンを発行する

⑥ トークンの発行



トークンの文字列は発行直後しか表示されないため、安全な別の場所に保存しておいてください。『クリップボードにコピー』をクリックすることでトークンの文字列がコピーされます。

トークンの発行後の画面



Sonar と Ksocket がまだ接続されていない状態であることを表します。
この状態では、スキャンを行うことは出来ません。

設定手順１：Ksocket のセットアップ

1-3 サーバーへ Ksocket をインストールする
スキャンを行うネットワーク上のサーバーに Ksocket をインストールします。
Ksocket はスキャン対象の端末へのインストールは不要です。

先ずはお客様のOSに適したソフトウェアをダウンロードしてください。

Ksocketダウンロード

Windows OS

・ Windows x86-64（※Windows 版を利用する場合は別途Npcapが必要となります。 詳細はユーザーマニュアルをご参照ください。）

Linux OS

・ Linux x86-64（※一般的なLinuxサーバーではこちらのx86-64版をご使用ください。）
・ Linux armv7l(arm32bit)
・ Linux aarch64(arm64bit)

次に、ダウンロードした Ksocket のインストールと Ksocket の接続を行います。
接続方法については下記URLをご覧ください。

➢**Ksocketユーザーマニュアル**
https://ksocketarchive.z11.web.core.windows.net/manual/ksocket-v2.2.1.pdf

※Ksocket をインストールしたサーバーに障害が発生した場合、スキャンができなくなる可能性があります。
予め Ksocket をインストールしたサーバーを複数台用意することで、予備機としてもご使用いただけます。

設定手順１：Ksocket のセットアップ

1-4 詳細情報をスキャンするための認証情報を設定する

Ksocket は、検知した機器の IPアドレスなどの簡易な情報が取得できますが、
スキャン対象の端末にログインするための認証情報を準備することで、端末の詳細情報をスキャンすることが可能です。

対応プロトコル

Ksocket は以下のプロトコルでの認証に対応しています。

- SNMP(v2c/v3)
- SSH
- WinRM

設定ファイル

設定は toml 形式のファイルに記載し、以下のパスに格納します。

アカウント設定ディレクトリの構成

```
/opt/fixpoint/ksocket/etc/ksocket/credentials/
├── snmp
│   ├── 999-default.toml
│   └── 999-example.toml.skeleton
├── ssh
│   └── 999-example.toml.skeleton
└── winrm
    └── 999-example.toml.skeleton
```

1つのアカウント情報ごとに1つのtomlファイルを作成します。また.skeletonファイルは記載方法のテンプレートとして用意されていますので、.tomlにコピーして利用してください。各プロトコルごとに記載し、ファイルのソート順の早い順で認証情報を適用してアクセスを試みます。

設定手順１：Ksocket のセットアップ

1-4 詳細情報をスキャンするための認証情報を設定する

SNMP v2c の場合

例）

```
includes = ["10.10.0.0/24",
            "10.20.0.0/24"]

port = 161

[authData]
community = "public“
```

ここで "includes“ ではアカウント情報を適用する対象の IPアドレスです。Ksocket が検知した際に、ここに含まれている IPアドレスであった場合は記載されているアカウント情報を利用してアクセスを試みます。
ここに記載した IPアドレス全てに対して Ksocket からアクセスを試みるわけでは無い点にご注意ください。

SNMP v3 の場合

例）

```
includes = ["10.10.0.0/24",
            "10.20.0.0/24"]

port = 161

 [authData]
username = “snmp-user”

authProtocol = "usmHMACMD5AuthProtocol“
authKey = "your-password“
privProtocol = "usmAesCfb128Protocol“
privKey = "priv-password"
```

"authprotocol“ には以下のいずれかを記載します。
- ・認証をしない場合： "usmNoAuthProtocol“
- ・MD5を使用する場合： "usmHMACMD5AuthProtocol“
- ・SHAを使用する場合： "usmHMACSHAAuthProtocol"

"privProtocol“ には暗号化方式を記載します。
- ・暗号化をしない場合： "usmNoAuthProtocol“
- ・DESを使用する場合： "usmDESPrivProtocol“
- ・AESを使用する場合： "usmAesCfb128Protocol"

設定手順１：Ksocket のセットアップ

1-4 詳細情報をスキャンするための認証情報を設定する

SSH の場合

パスワードログインの場合
例）

```
includes = ["10.10.0.0/24"]

port = 22

 [account]
 username = "john“
 password = "passw0rd"
```

鍵認証でのログインの場合
例）

```
includes = ["10.10.0.0/24"]
port = 22

[account]
username = "john“
[[account.clientKeys]]
filename = "../../id_rsa.common“
passphrase = "secret_credential"
```

WinRM の場合

Ksocket にて Windows 機の詳細情報を取得するためにはスキャン対象ノード側の WinRM 接続を許可する必要があります。
設定方法は「WinRM 接続の有効化の方法」（P.18）にてご案内しております。

例）

```
includes = ["10.10.0.0/24"]

port = 5985

authMethod = "ntlm"

[account]
username = "john“
password = "passw0rd“
```

" authMethod"には認証形式を記載します。
"basic", "ntlm", "credssp"のいずれかから選択できます。

設定手順１：Ksocket のセットアップ

1-4 詳細情報をスキャンするための認証情報を設定する

複数のアカウント情報が存在する場合

```
ssh
├── 100-test1.toml
├── 200-test2.toml
├── 300-test3.toml
└── 999-example.toml
```

上記のように複数のアカウント情報ファイル(tomlファイル）が存在する場合には、以下の順序で処理されます。

1. 100-test1.toml のアカウントを使ってsshアクセス試行する
2. 200-test2.toml のアカウントを使ってsshアクセス試行する
3. 300-test3.toml のアカウントを使ってsshアクセス試行する
4. 999-example.toml のアカウントを使ってsshアクセス試行する

あるネットワークのゾーン内のアカウント情報を共通化させたい場合には、“includes”で指定する事ができます。
言い換えれば、各接続先ごとに認証情報が異なる場合には、各接続先ごとにアカウント情報ファイルを用意する必要があります。

設定手順１：Ksocket のセットアップ

1-4 詳細情報をスキャンするための認証情報を設定する

設定ファイルを暗号化/復号化する

toml 形式で保存されたアカウント情報は平文で保存されていますので、
必要に応じて暗号化を行います。

```
$sudo /opt/fixpoint/ksocket/bin/ksocket encrypt (認証情報のtomlファイル名)

例
$sudo /opt/fixpoint/ksocket/bin/ksocket encrypt 999-sample.toml
```

拡張子“.toml”が “.toml.kscrypt” という暗号化ファイルになります。
（Ksocket 本体は kscrypt ファイルを読み取り、内部的に復号処理を
行って対象の機器にアクセスを行います。）

暗号化したアカウント情報ファイルを復号する際は以下のように行い
ます。

```
$sudo /opt/fixpoint/ksocket/bin/ksocket decrypt (暗号化認証情報の.kscryptファイル名)

例
$sudo /opt/fixpoint/ksocket/bin/ksocket decrypt 999-sample.toml.kscrypt
```

拡張子".toml.kscrypt"ファイルが平文の".toml"に復号されます。
複合後はテキストエディターで編集できるようになりますので、
必要に応じて再度暗号化を行ってください。

暗号化設定ファイルの保存・バックアップ

Ksocket は初回起動時に RSA鍵ペアを作成保存します。暗号化・復号化では、ここで作成したRSA鍵が利用されます。 暗号化されたファイルを別の Ksocket 用サーバーに移動させるなど、元の RSA鍵にアクセスできない場合には復号化は出来ません。このためアカウント情報ファイルのバックアップや移動を行う際には、必ず復号化してから行ってください。
特に（コールドスタンバイ用などで）Ksocket用サーバを複数運用してアカウント情報ファイルを共有する場合、kscryptファイルのコピーは利用できませんのでご注意ください。

設定手順１：Ksocket のセットアップ

1-4 詳細情報をスキャンするための認証情報を設定する

WinRM 接続の有効化の方法

Ksocket にて Windows 機の詳細情報を取得するためにはスキャン対象ノード側の WinRM 接続を許可する必要があります。これを有効化するには以下の手順を対象の Windows PowerShell コンソール(管理者権限)で実行してください。

スキャン対象側 Windows 機の設定画面

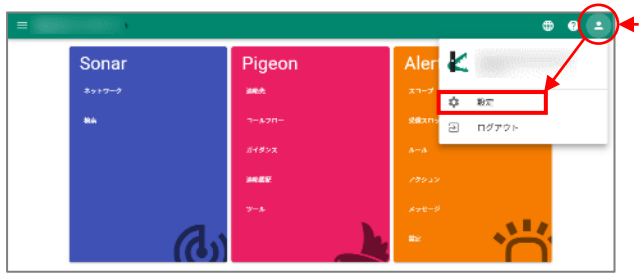
```
# ExecutionPolicyがRestrictedだった場合、RemoteSignedに変更する
> Get-ExecutionPolicy
Restricted
> Set-ExecutionPolicy RemoteSigned
> Get-ExecutionPolicy
RemoteSigned

# WinRMサービスを実行できるようにする
> winrm qc

# Basic認証で接続する場合は、Basic認証での接続を許可する
> winrm set winrm/config/service/auth ‘@{Basic=“true”}’
> winrm set winrm/config/service '@{AllowUnencrypted="true"}’

# ユーザに対して読み取り権限を付与する
# 以下コマンド実行によって表示されたウィンドウで、
# 該当するユーザに読み取り権限と実行権限を許可して適用
> winrm configSDDL default

# WMIリソースのアクセス権限設定
# 以下コマンド実行によって表示されたウィンドウで、
# [操作]>[プロパティ]>[セキュリティ] を選択
# - Root¥CIMV2 から[セキュリティ]を選択し、
# 該当するユーザにメソッドの実行とリモートの有効化を許可して適用
# - Root¥StandardCimv2 から[セキュリティ]を選択し、
# 該当するユーザにメソッドの実行とリモートの有効化を許可して適用
> wmimgmt.msc
```

設定手順１：Ksocket のセットアップ

1-5 Sonar と Ksocket の接続を確認する

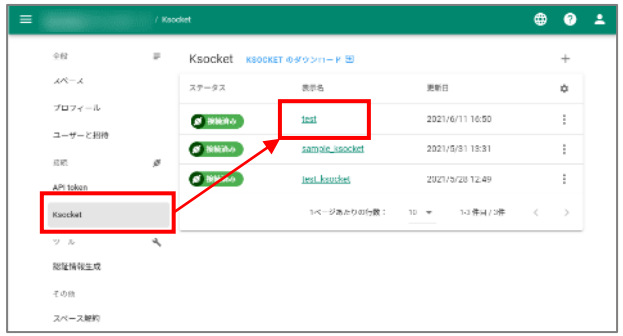
Ksocket トークンの認証が完了し、Sonar と Ksocket が接続したことを管理画面で確認します。

① 『アカウントメニュー』-»『設定』の順にクリック



---アカウントメニュー

② 『Ksocket』-»『（表示名） 』の順にクリック



「1-2 Ksocketトークンを発行する」で作成した Ksocket 表示名をご選択ください。（左図の場合は「test」を選択しています。）

③ 接続を確認

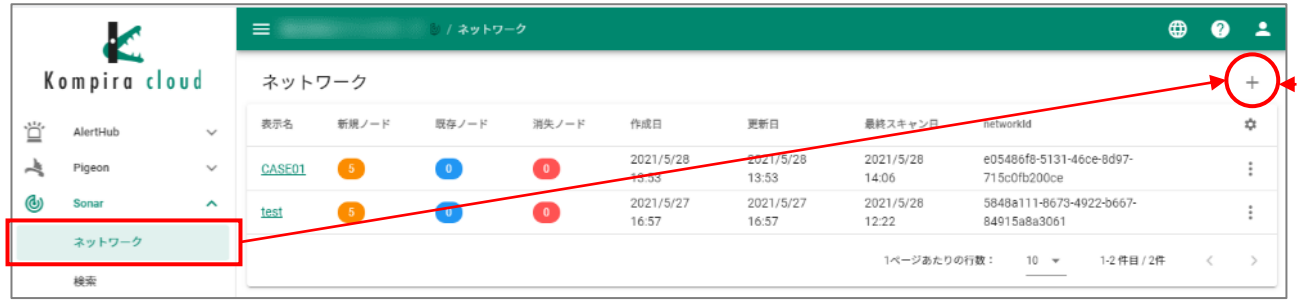
Ksocket が接続されている
ステータスを表します。

設定手順２：ネットワークの作成

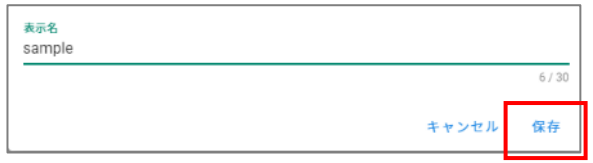
2-1 ネットワークの作成を行う

ネットワークの設定をおこないます。
ネットワークの表示名はスキャンを行う範囲に則した分かりやすい表示名を入力してください。

①『ネットワーク』-»『 ⊕（新規作成ボタン）』の順にクリック



②任意の表示名を入力 -»『保存』をクリック



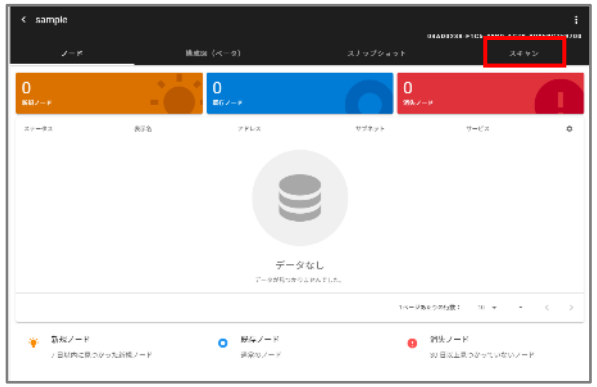
③新規ネットワークが作成される

設定手順３：スキャンの設定と実行

3-1 スキャンの設定

Ksocket がインストールされているネットワーク内を探索し、探索時点の構成情報をスナップショットとして取得することを「スキャン」と呼びます。

①『スキャン』のタブに移動



②スキャンに使用する Ksocket にチェックを入れる

Ksocketにチェックを入れると
スキャンの設定ボックスが表示されます。

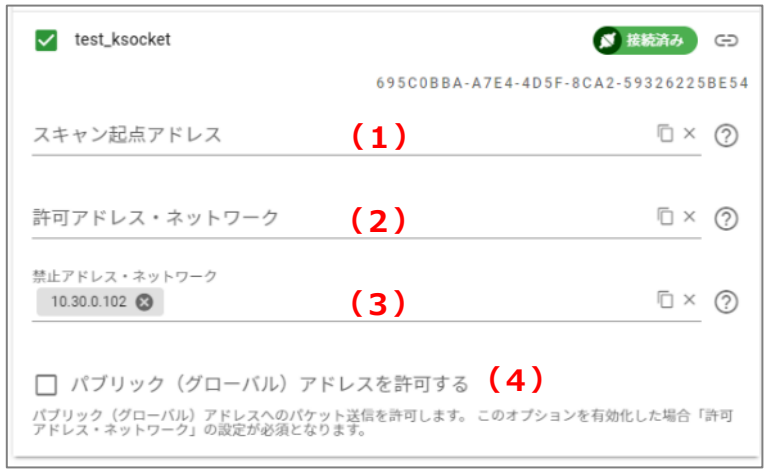
設定手順３：スキャンの設定と実行

3-1 スキャンの設定

Ksocketはアクセス可能なIPアドレスに対してパケットの送信やログインの試行を行います。
そのため、必要に応じてKsocketがアクセスしても良いIPアドレス範囲を指定してください。

③スキャンの設定ボックスの入力
（任意の設定です。不要な場合はP.23-①までおすすみください。）

設定ボックス



※IPアドレス範囲が不明な場合は、お客様環境のネットワーク管理者かネットワーク事業者にご確認ください。

（1）スキャン起点アドレス
スキャンの起点となるIPv4アドレスです。主に任意のサブネットに存在する機器を検知する際に必要となります。

（2）許可アドレス・ネットワーク
スキャン時のホワイトリストとして利用されるIPv4アドレス・ネットワークです。ここに記載されたIPアドレスにのみパケットを送信します。未指定時は 0.0.0.0/0 が指定されたものとして動作します。

（3）禁止アドレス・ネットワーク
スキャン時のブラックリストとして利用されるIPv4アドレス・ネットワークです。ここに記載されたIPアドレスにはパケットを送信しません。この指定は許可アドレス・ネットワークより優先されます。

（4）パブリック（グローバル）アドレスを許可する
初期状態のSonarでは、パブリックIPに対するスキャンを行わないように設定されているため、必要な場合にはチェックを入れます。チェックを入れる場合、（2）許可アドレス・ネットワークの欄に1つ以上のアドレス・ネットワーク指定が必要です。

設定手順３：スキャンの設定と実行

3-2 スキャンの実行

①『保存してスキャン』をクリックする

スキャンに関する各種機能については、下記ページでご紹介しております。

- 「スキャンのオプション１：スケジュール機能」
- 「スキャンのオプション２：通知機能」

『保存してスキャン』
スキャンに使用する Ksocket やスキャン範囲の設定などを保存し、そのままスキャンを実行します。

『デフォルトとして保存』
スキャンに使用する Ksocket やスキャン範囲の設定などの保存のみを行います。

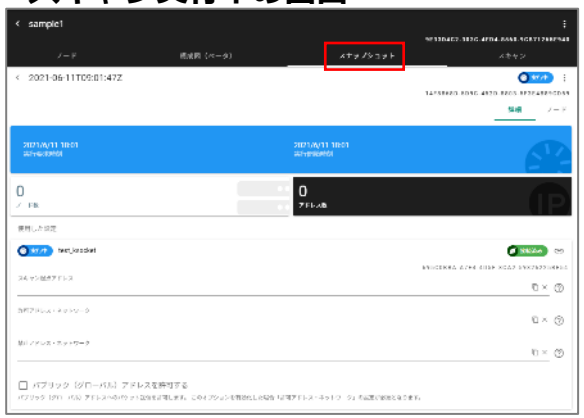
『スキャン開始』
保存されているスキャン設定に従ってスキャンを実行します。

設定手順３：スキャンの設定と実行

3-3 スキャン実行後の確認

①スキャンが開始されると、画面表示が自動的に『スナップショット』のタブに移行

スキャン実行中の画面



スキャン完了の画面

対象機器の数の違いなどにより、スキャンにかかる時間は異なります。

②『ノード』をクリックすると、スキャンされたノード一覧が確認可能

スキャンのオプション１：スケジュール機能

スキャンに関するオプション機能のご紹介です。
スケジュール機能を使い、スキャン実施のスケジューリングが可能です。

設定済のスケジュールの変更は『⋮（編集ボタン）』から可能です。

※スケジューリングは最大3つまで行えます

編集ボタン

１回のみ

毎日

毎週

毎月

スキャンのオプション２：通知機能

スキャンに関するオプション機能のご紹介です。
通知機能を使うと、『メール』もしくは『Webhook』での通知が可能です。

通知例：「メール」の場合

メール

通知を受けたいメールアドレスを設定します。
アドレスの設定は5件まで可能です。

Webhook

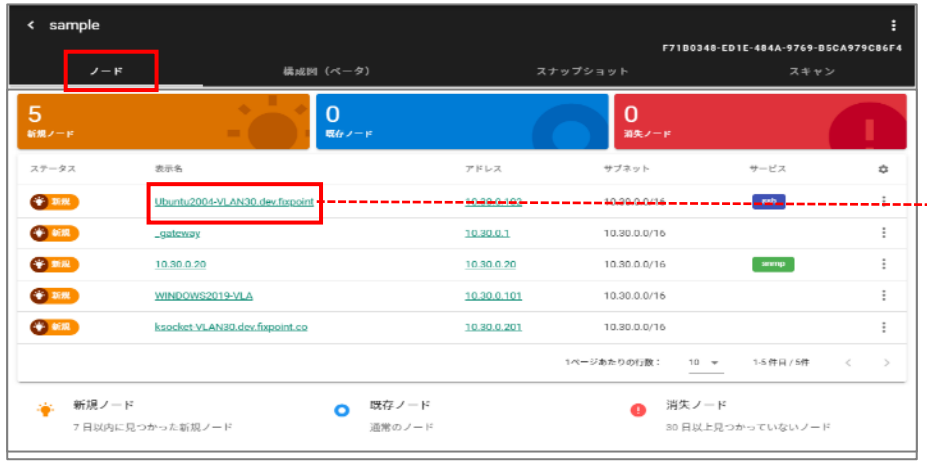
任意の Web サービスなどに対して Webhook で通知を送信できます。

設定画面の確認１：スナップショット

スキャンが実行されると、スナップショットが作成されます。
スナップショットにはスキャンした時点のネットワーク・各ホストの状態 （例えばアドレス情報やアドレスに紐づいたホストの持つパッケージ等の情報） が含まれています。
『スナップショット』のタブで確認できます。

設定画面の確認２：ノード

スナップショットの情報を集約し、各機器の情報や状態は管理ノードとして扱われます。
管理ノードは『ノード』のタブで確認できます。



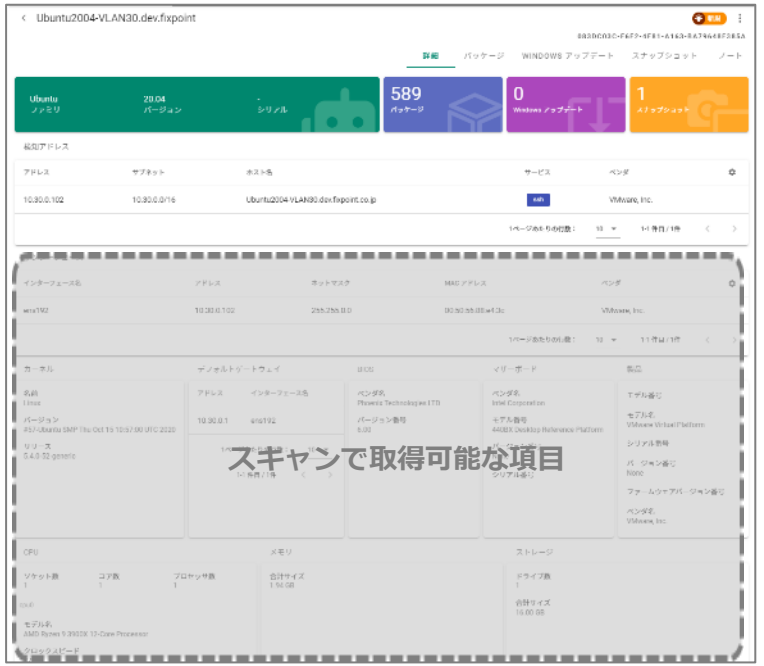
スキャンで取得可能な項目

Windows
- 製品OS
- シリアル
- マザーボード
- BIOS
- Memory
- CPU
- Disk インターフェイス
- デフォルトゲートウェイ
- インストール済みパッケージ
- Windows Update

Linux
- 製品OS
- シリアル
- マザーボード
- BIOS
- Memory
- CPU
- Disk インターフェイス
- デフォルトゲートウェイ
- インストール済みパッケージ

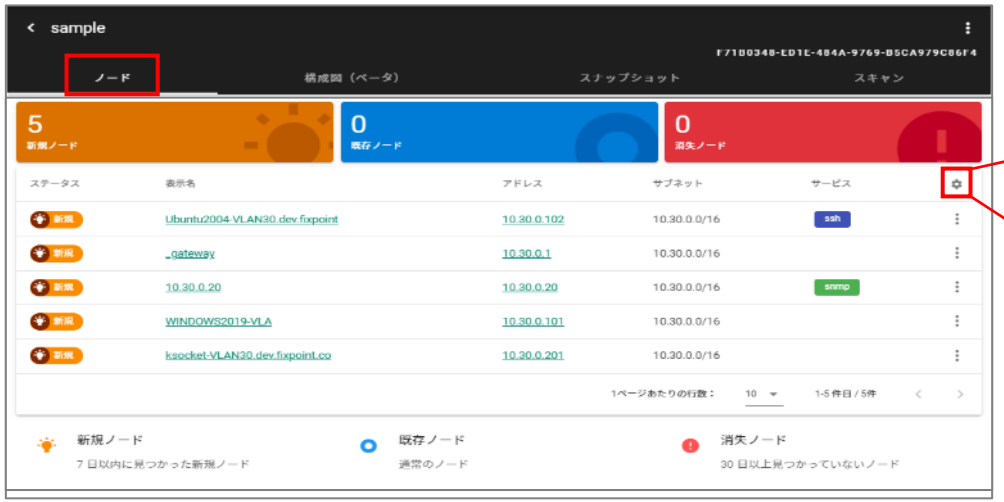
NW機器
- 製品情報
- シリアル

ノードの詳細例



設定画面の確認２：ノード

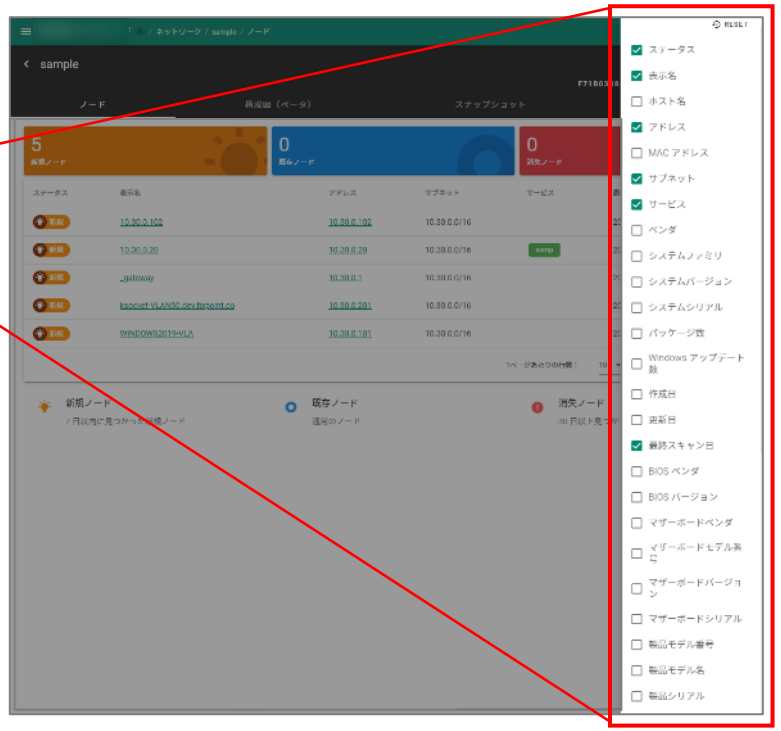
『ノード』には、最新のスキャン結果だけではなく、過去のスキャンで取得された情報が表示されます。
対象機器の停止等の理由により、最新のスキャンでノードが検知されなかった場合でも、手動削除しない限りノード一覧から自然に消えることはありません。



新規検知からの経過日数により、ノードのステータスが移行します。

新規ノード：7日以内に新たに検知されたノード
既存ノード：新規ノードに該当せず消失していないノード
消失ノード：最後に検知されてから30日以上経過したノード

表示項目の設定

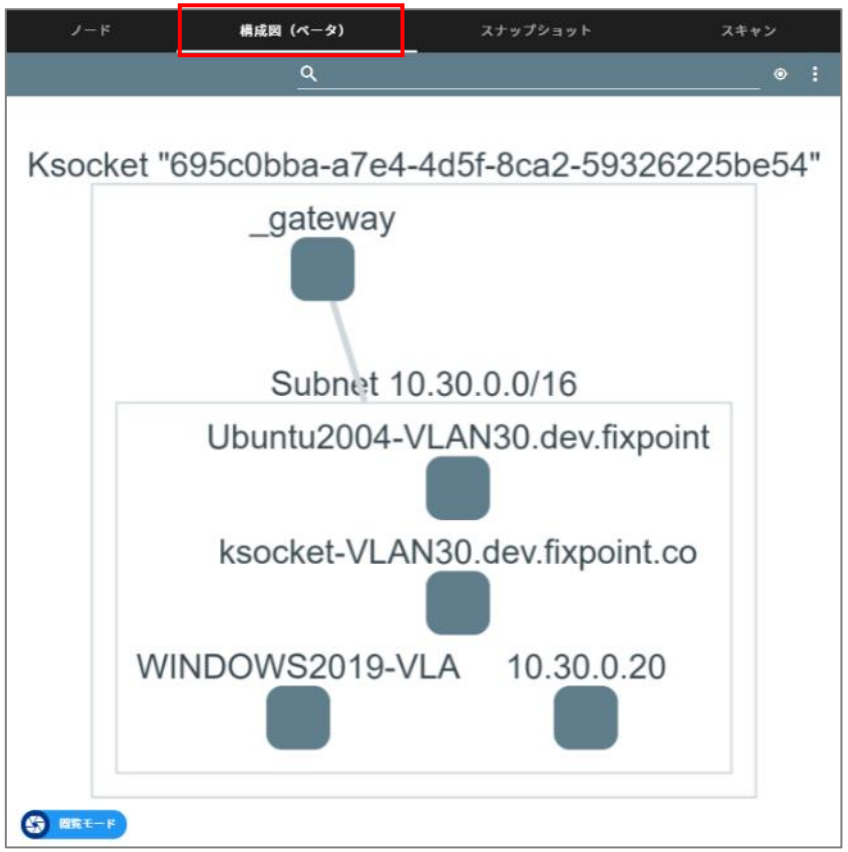


表示項目を選択することが可能です。
例）・ホスト名 ・MACアドレス ・サブネット
・システムバージョン ・パッケージ数
・Windowsアップデート数
・作成日 ・更新日 ・最終スキャン日　　他

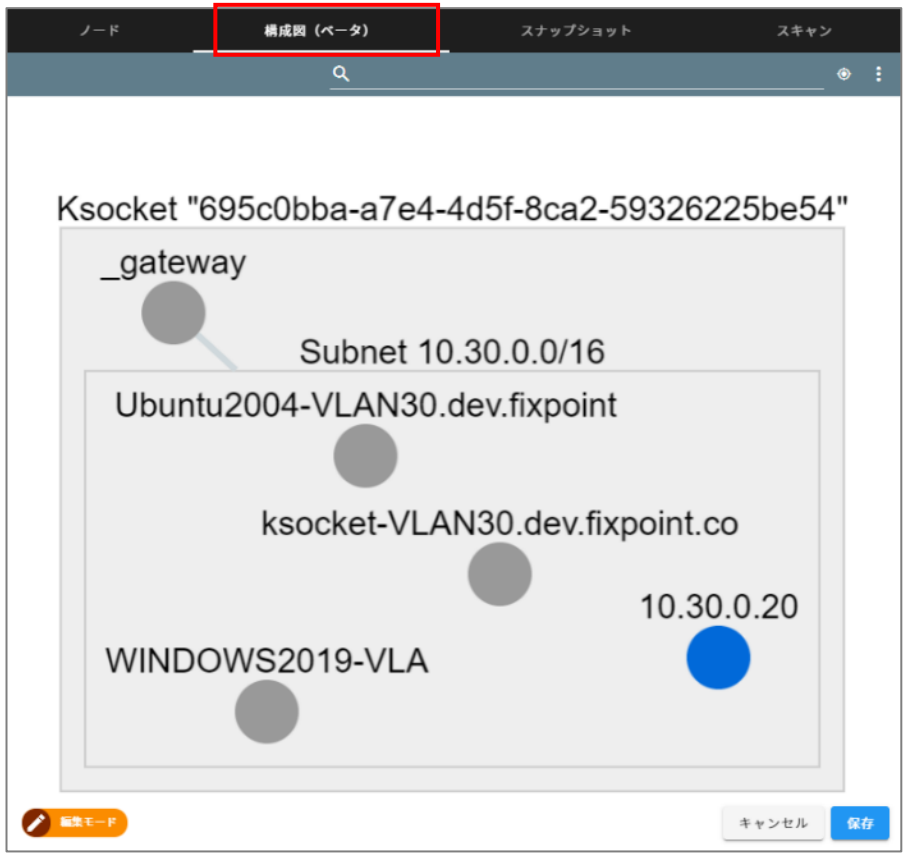
設定画面の確認３：構成図(ベータ)

『構成図（ベータ）』のタブでは、スキャンした各機器をネットワークアドレス単位でまとめて視覚的に確認できます。

閲覧モード



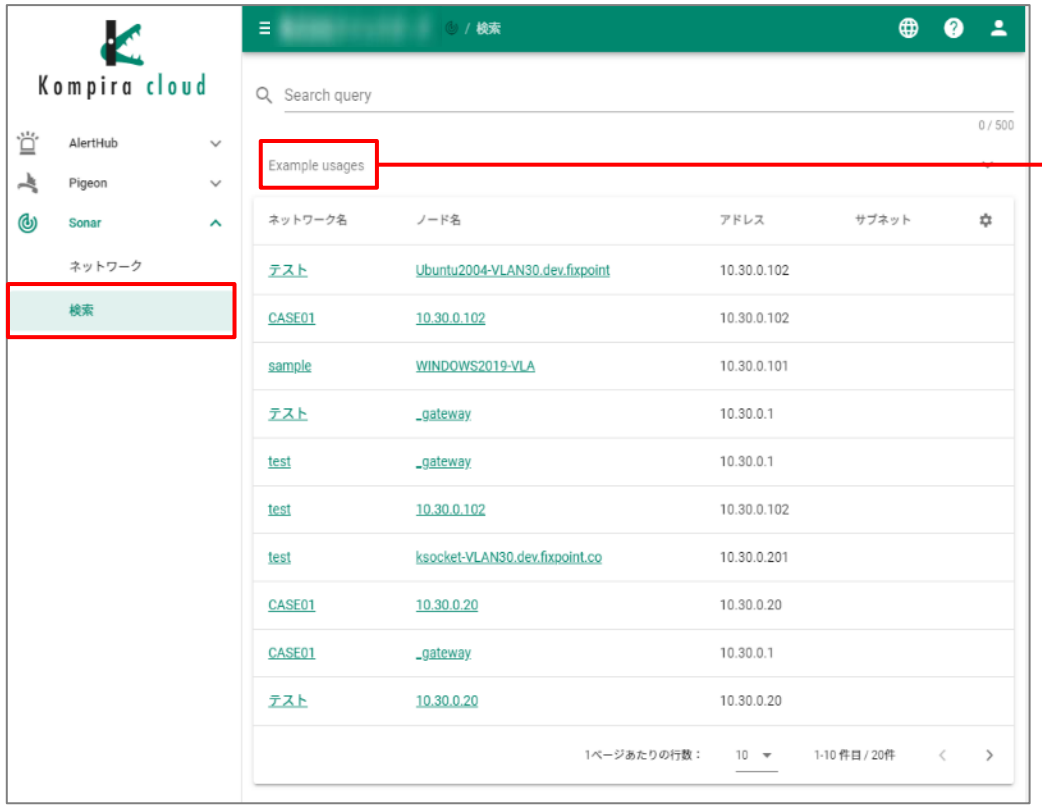
編集モード



設定画面の確認４：検索

スキャンで取得したノード情報を、様々な条件で検索することができます。

① 『検索』-» 『Example usages』をクリック



検索項目の例の表示から、検索したい情報に該当する例文を確認

設定画面の確認４：検索

②『Search query』に検索ワードを入力　-»【Enter】キーを押すことで検索が開始

検索例１
対象ネットワークに “テスト” を含むノードを
検索したい場合...

検索ワード　network:“テスト“

検索例２
対象ネットワークに “test” を含む、かつ、
システム名に CentOS を含むノードを
検索したい場合...

検索ワード　network:"test" AND system:CentOS

Kompira Sonar コミュニティ

本マニュアルでご紹介した内容以外で知りたい項目などがございましたら、
株式会社フィックスポイントの公式コミュニティにお問い合わせください。
https://kompira.zendesk.com/hc/ja/community/topics/360000013382-Sonar%E9%96%A2%E9%80%A3

また、Kompira 製品サイトでは「Kompira cloud マニュアル」やSonarの
機能に関する各種情報を公開しております。ぜひこちらもご参照ください。

・Kompira cloud 製品利用ガイド
https://kompira.zendesk.com/hc/ja/articles/4403090973209

・Sonar 製品情報
https://kompira.zendesk.com/hc/ja/sections/900001026646-Kompira-Sonar