# Alibaba Cloud
# 日志服#

## クイックスタート

**Document Version20191122**

# 目次

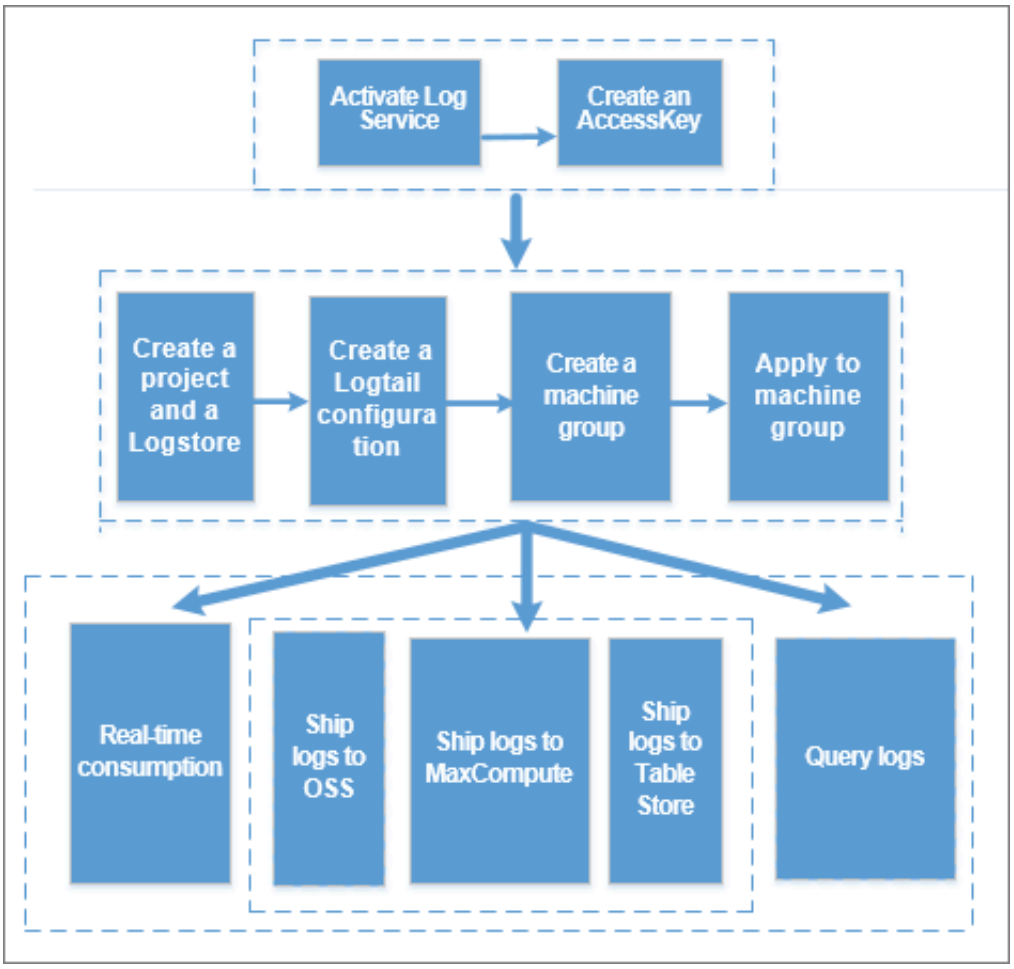
# 15分でクイックスタート

**Log Service** は、膨大なログを収集、格納、および照会するためのプラットフォームです。**Log Service** を使用することで、運用中のクラスター内のすべてのログを一元管理とができます。また、リアルタイムに読み取り、照会することもできます。

本ドキュメントでは、**Windows** 環境で **Elastic Compute Service (ECS)** のログを収集するための **Logtail** 設定の基本的な流れについて説明します。**Log Service** が初めての方でもご利用いただけるよう、ログ収集、リアルタイムなログ照会といった **Log Service** の基本機能について説明しています。

## Log Service の操作手順

**図** 1-1 : **手順**



### ステップ1. はじめに

#### 1. Log Service の有効化

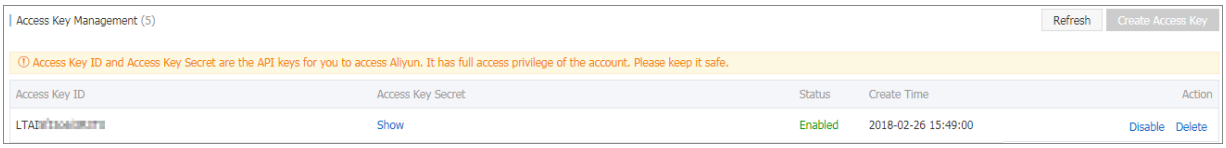
登録済みの Alibaba Cloud アカウントで Log Service プロダクトページにログインし、有効化をクリックします。

#### 2. AccessKey の作成 (オプション)

> 注：
> SDK でデータを書き込む場合は、プライマリアカウントやサブアカウントに AccessKey を作成する必要があります。ログの収集に、AccessKey の作成は必須ではありません。

Log Service コンソールの右上隅にあるアバターの上にマウスを乗せ、表示されるドロップダウンリストよりAccessKey 管理をクリックします。ダイアログボックスで、AccessKey の管理をクリックして、AccessKey 管理ページに移動します。その上で AccessKey を作成します。ステータスが有効になっていることを確認します。

**図 1-2 : AcessKey の有効化**



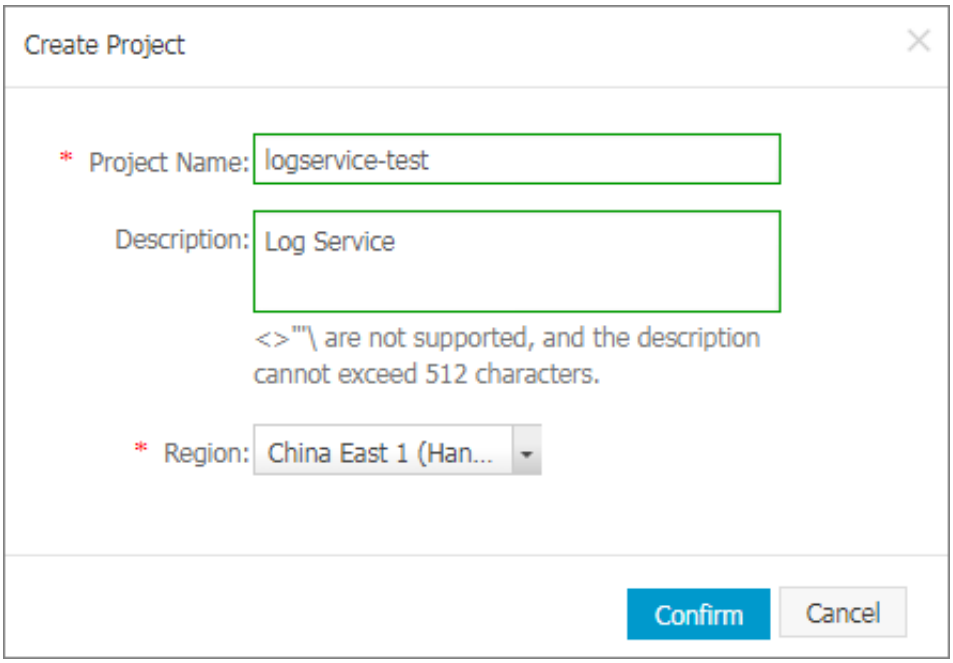

#### 3.プロジェクトの作成

Log Service コンソールに初めてログインする場合、プロジェクト作成プロンプトが表示されます。右上にあるプロジェクトの作成ボタンをクリックすることでも、プロジェクトは作成できます。

プロジェクトを作成する際は、プロジェクト名とリージョンを指定します。cn-shanghai-internal-prod-1およびcn-hangzhou-internal-prod-1は、Log Service のシステム内部に

より使用されるリージョンです。その他のリージョンは、パブリッククラウド上のリージョンです。

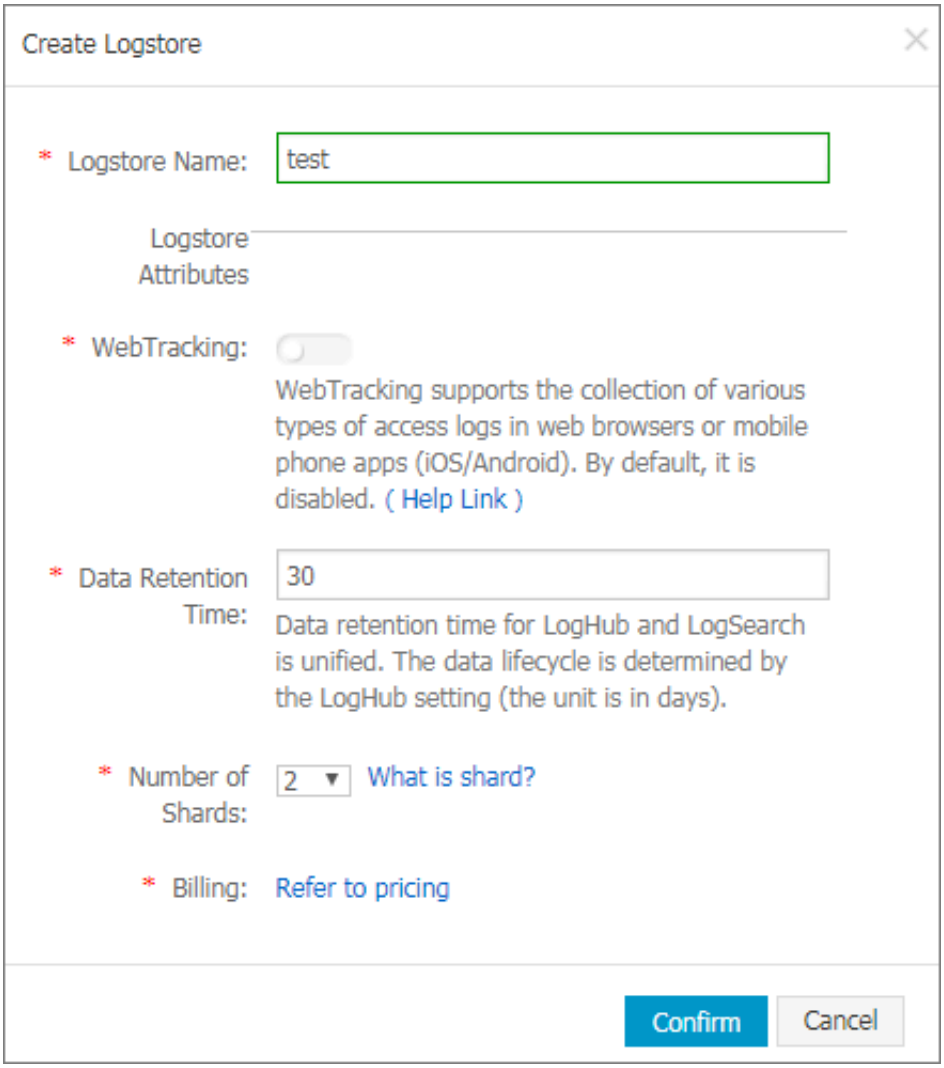
**図 1-3 : プロジェクトの作成**



4.Logstore **の作成**

プロジェクトを作成すると、**Logstore** 作成プロンプトが表示されます。プロジェクトに移動し、右上隅の作成をクリックすることでも作成できます。**Logstore** を作成する際に、ログの運用方法も指定します。

**図** 1-4 : Logstore **の作成**




**ステップ**2. ECS **インスタンスに** Logtail **クライアントをインストール**

1. **インストールパッケージをダウンロード**

**ECS** インスタンスに、**Logtail** のインストールパッケージをダウンロードします。**Windows** 用のインストールパッケージをダウンロードするには、こちらをクリックします。

### 2. Logtail をインストール

現行ディレクトリにインストールパッケージを解凍し、`logtail_installer`ディレクトリに移動します。管理者アカウントで **cmd** を実行し、インストールコマンド `.\logtail_installer.exe install cn_hangzhou`を実行します。

注：

ネットワーク環境と **Log Service** リージョンによって、実行するインストールコマンドは異なります。 本ドキュメントの例では、中国東部1 (杭州) の **ECS** クラシックネットワークを使用します。その他エリアについては、「*Windows に Logtail をインストール*」をご参照ください。

その他のリージョンにインストールする場合のコマンドについては、「*Windows に Logtail をインストール*」および「*Linux に Logtail をインストール*」をご参照ください。

## ステップ3. データインポートウィザードの構成

**Log Service** コンソールでプロジェクト名をクリックして**Logstore** リストページに移動します。対象**Logstore** 名の横にあるデータインポートウィザードアイコンをクリックして**Logtail** の設定に移動します。 また **Logstore** 設定の横にある管理をクリックして、**Logtail** 設定リストに新規設定を作成します。

**Logtail** 設定には、次のステップがあります。データソースの選択、データソースの構成、検索、分析、および可視化、送信/ETL。最後の **2** つのステップはオプションです。

### 1. データソースを選択

**Log Service** は、多くのクラウド製品、独自のソフトウェア、カスタムデータのログ収集をサポートしています。本ドキュメントでは、テキストログの収集を例に取り上げます。詳細については、「テキストログの収集」をご参照ください。

その他のソースのテキストをクリックし、 次へ をクリックします。

### 2. データソースを構成

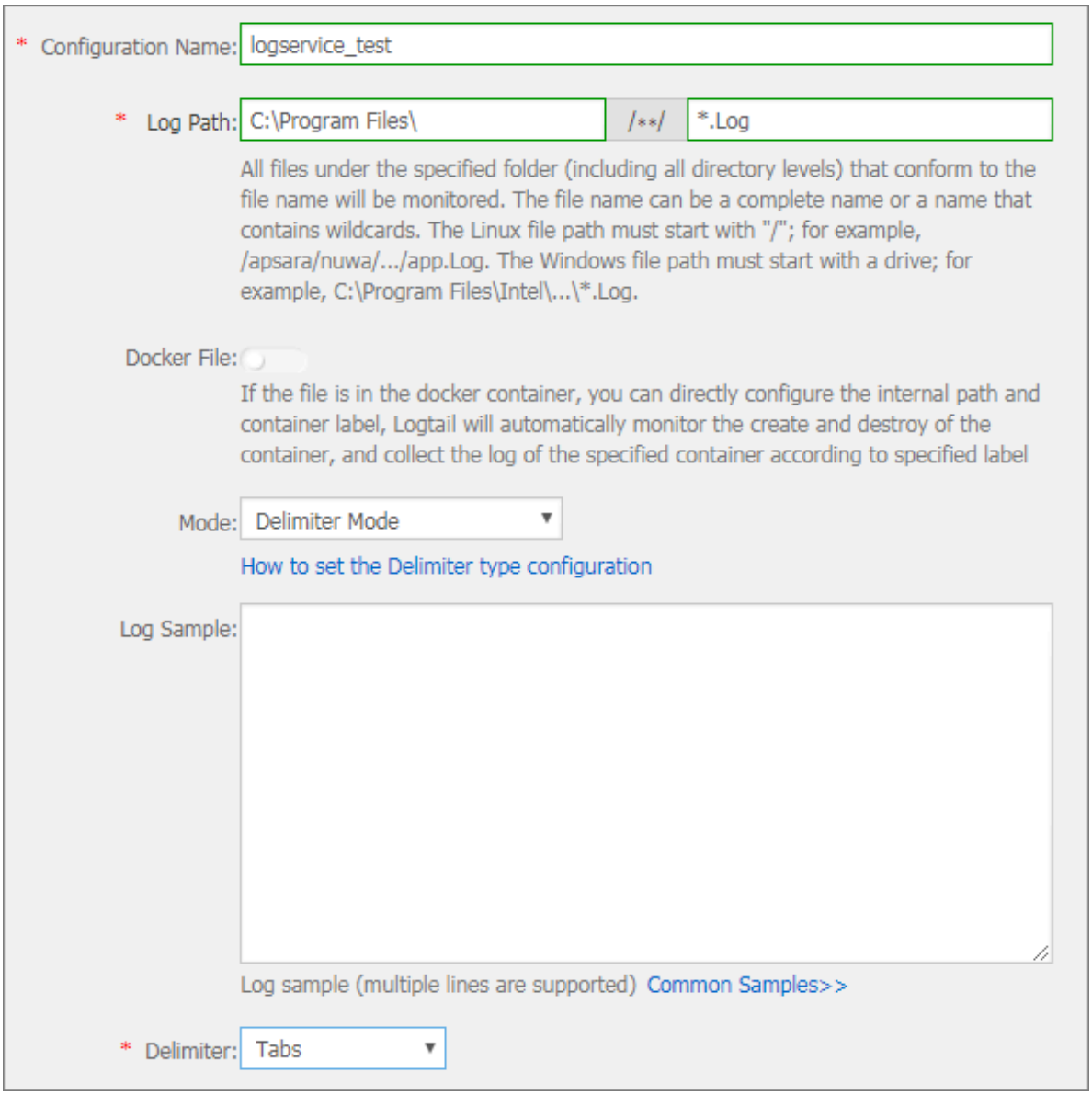
- 構成名とログパスを指定します。

  ページの指示に従って、構成名、ログパス、およびログファイル名を入力します。 ログファイル名はフルネームにすることができ、また、ワイルドカードマッチングをサポートします。
- ログ収集モードを指定します。

  現時点において、**Log Service** ではシンプルモード、デリミタモード、**JSON**モード、フルモード、または **Alibaba Cloud** カスタムモードでの解析ログをサポートしています。 このド

キュメントでは、デリミタモードを例として使用しています。 収集モードの詳細については、「テキストログの収集」および「テキストログの設定と解析」をご参照ください。

**図 1-5 : データソースの構成**




- ログサンプルを入力します。

  デリミタモードまたはフルモードがログ収集モードとして選択されている場合は、ログサンプルを入力する必要があります。 **Log Service**は、**Logtail**の設定時に選択した設定に従ってログサンプルの解析をサポートします。 ログサンプルを解析できなかった場合は、区切り文字の設定や正規表現を変更する必要があります。 ログサンプルフィールドに、解析するログサンプルを入力します。

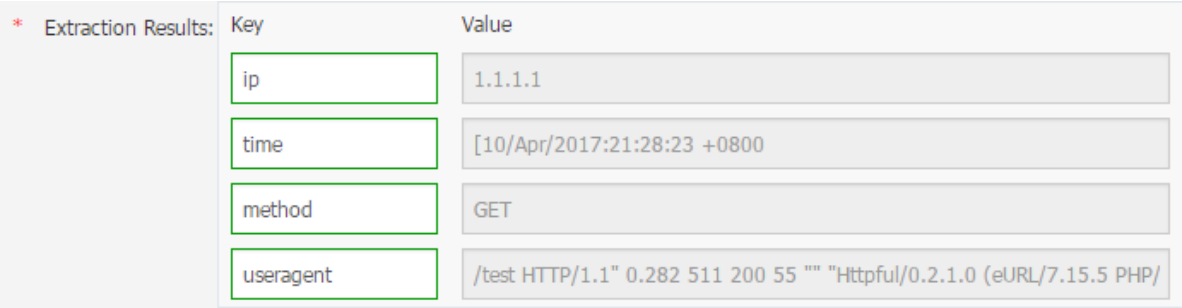
- 区切り文字を指定します。

  タブ、縦線、またはスペースを区切り文字に指定できます。区切り文字をカスタマイズすることもできます。ログフォーマットに合った区切り文字を選択します。フォーマットに合ったものでない場合は、ログの解析に失敗します。

- ログ抽出結果にキーを指定します。

  ログサンプルを入力して区切り文字を選択すると、選択した区切り文字でログフィールドが値として自動抽出されます。値のキーを指定します。

  **図 1-6 : ログコンテンツの抽出結果**




- 必要に応じて詳細オプションを構成します。

  通常は、詳細オプションをデフォルト設定で十分です。詳細オプションを設定する場合は、「テキストログの収集」をご参照ください。

- マシングループに適用します。

  初めてマシングループを作成する場合は、ページの指示通りにマシングループをご作成ください。 それから、**Logtail** 構成をマシングループに適用します。

  > 注 :
  > **Armory** を作成してマシングループに関連付けるには、ページの指示に従って指定した内部リンクにジャンプします。

上記のステップをすべて終了するとすぐに **Log Service** によって **Alibaba Cloud ECS** インスタンスのログ収集が開始します。収集されたログは、コンソールや **API/SDK** を使用してリアルタイムに読み込むことができます。

ログを照会/分析、また、送信するには、次へをクリックします。

> 注 :
>
> - **Logtail** 設定が反映されるまでに最大 **3** 分かかります。

- **IIS** アクセスログを収集する場合は、「*Logstash* を使用した *IIS* ログの収集」をご参照ください。
- **Logtail** の収集エラーについては、「ログ収集エラーの診断」をご参照ください。

### 検索と分析および可視化

収集の設定をすると、リアルタイムに **ECS** ログが収集されます。収集ログを照会および分析するには、データインポートウィザードで次のようにインデックスを設定します。

**Logstore** リストページの検索をクリックすると、検索ページに移動します。右上隅の有効化をクリックし、表示される検索と分析ページでインデックスを設定します。

- フルテキストインデックス属性

  フルテキストインデックス属性を有効にします。大文字と小文字の区別を有効にするかを確認し、トークンの内容を確認します。
- キー/値インデックス属性

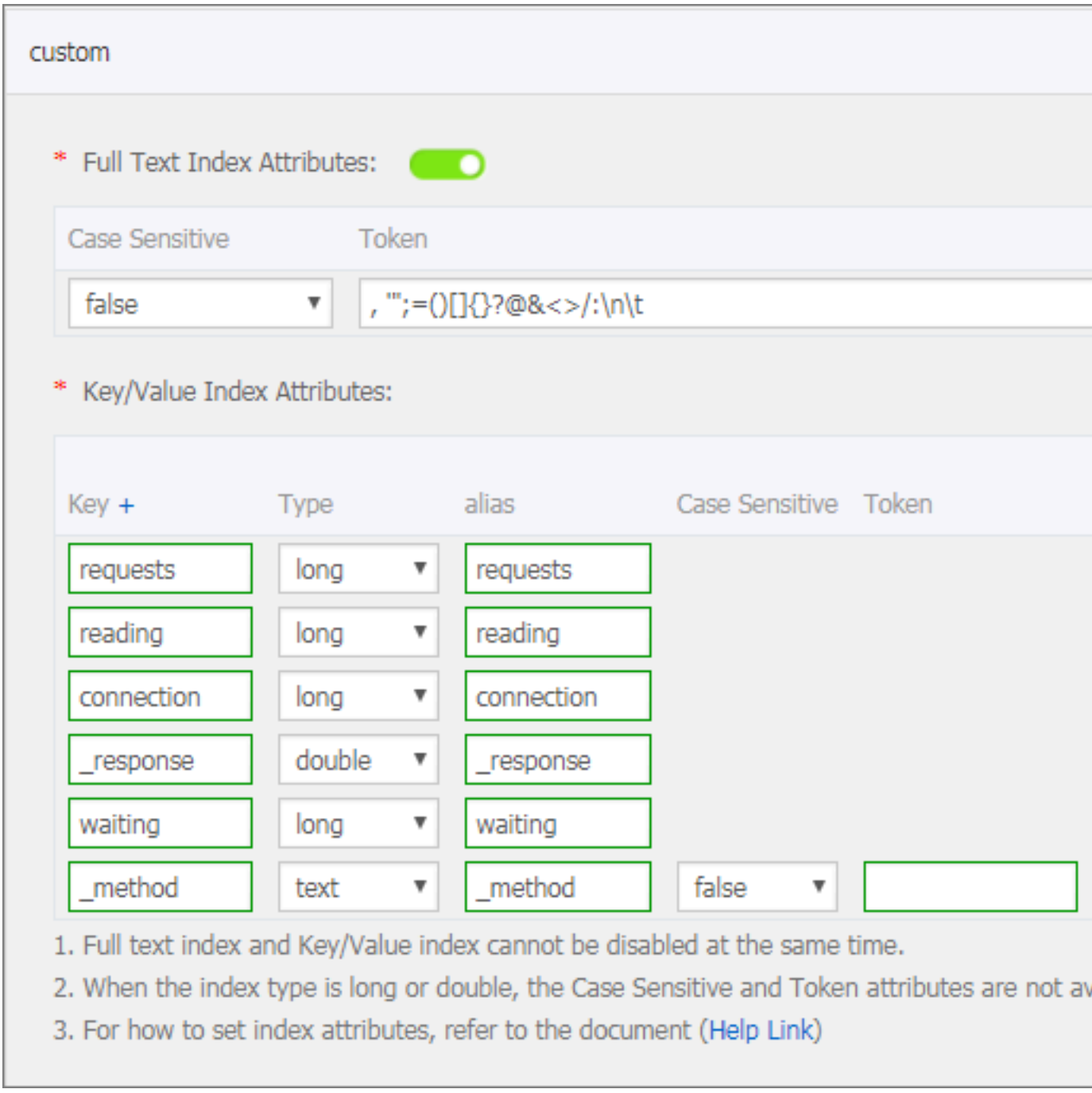
  キーの右側にあるプラスアイコンをクリックして行を追加します。 キー、タイプ、エイリアス、大文字と小文字の区別、トークンを設定し、分析を有効にするかどうかを選択します。

 注：

1. フルテキストまたはキー/値インデックス属性の内、**1** つは有効にする必要があります。両方とも有効にした場合は、キー/値インデックス属性が優先されます。
2. インデックスが **long** 型または **double** 型の場合、大文字小文字の区別およびトークン属性は利用できません。
3. インデックスの設定方法については、「概要」をご参照ください。

4. **Nginx** テンプレートまたは **MNS** テンプレートを使用するには、クエリページで有効化をクリックして、クエリ/分析ページの各設定を指定します。

**図 1-7 : 照会 (検索)と分析**




クエリ/分析を設定したら、ログの送信設定をする場合は、次へをクリックします。 照会分析を実行するには、**Logstore** リストページに戻り、検索をクリックして検索ページに移動します。キーワード、トピック、または検索と分析文を入力し、ログ期間を指定して検索します。 **Log Service** では直観的なヒストグラムを提供し、検索結果表示されます ヒストグラムをクリックす

ると、より詳細なにログ期間検指定して索することができます。 詳細については、「概要」をご参照ください。

**Log Service** は、クイッククエリや統計図表といった、さまざまな方法でログを照会分析できます。詳細については、「その他の機能」をご参照ください。

たとえば、直近 **15** 分間のすべてのログを照会するには、空の検索条件を指定し、期間に **15** 分を選択します。

4. **転送**

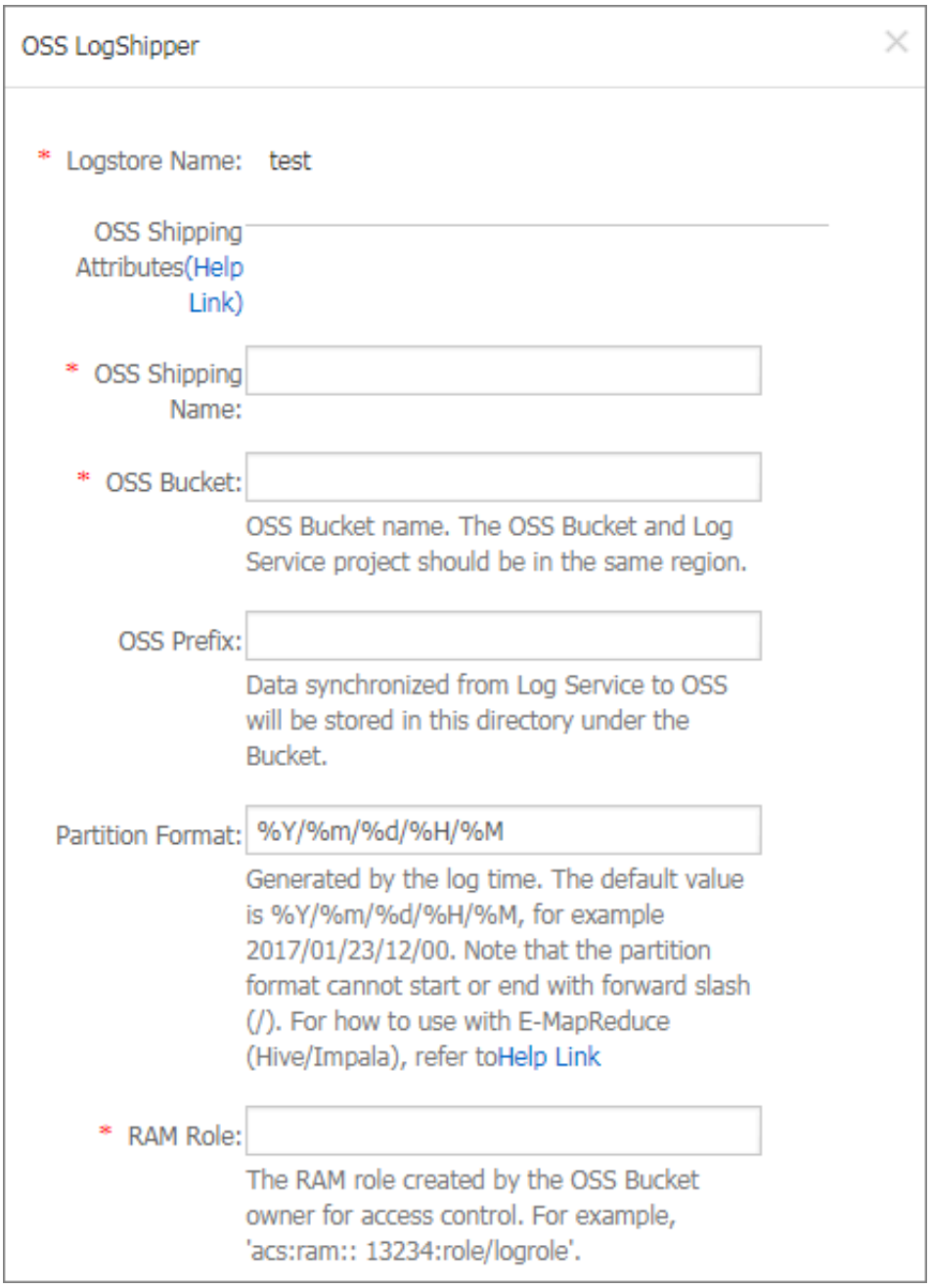
**Log Service** では、さまざまなデータソースやさまざまな形式のデータをまとめて収集、管理、保守できるだけでなく、 **Object Storage Service (OSS)** といったクラウドプロダクトにログデータを転送して処理と分析が行えます。

**OSS** にログを転送するには有効化をクリックします。

本ドキュメントでは、**OSS** ストレージを例に取り上げています。 「*OSS* へのログの転送」をご参照の上、認証まで完了させてください。

有効化をクリックすると、**OSS LogShipper** ダイアログボックスが表示されます。設定についての詳細は、「*OSS* へのログの転送」をご参照ください。設定し終わったら、確認をクリックして転送を完了します。

**図** 1-8 : LogShipper **の設定**




アクセス検索、およびログ分析といった基本機能に加え、 **Log Service** にはログを活用するさまざまな方法があります。詳細については、ユーザーガイドをご参照ください。

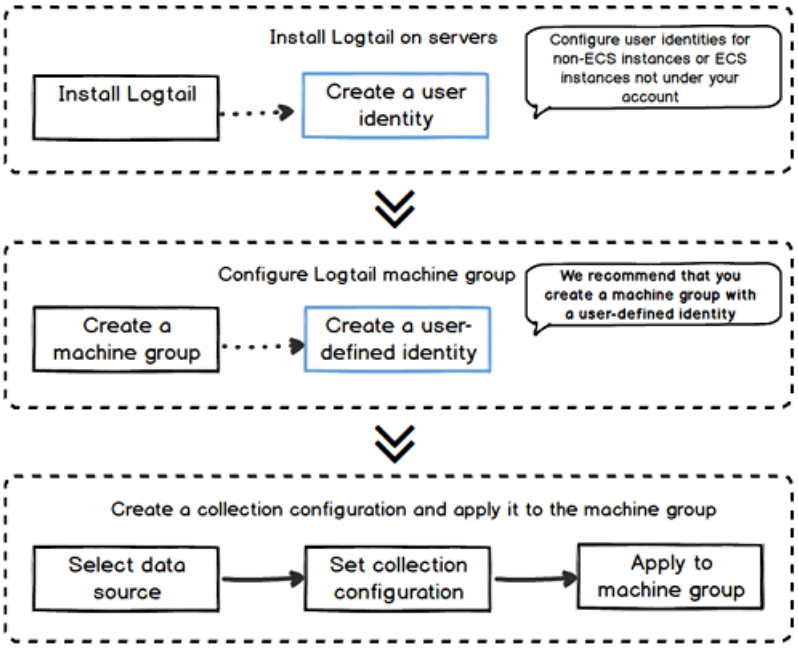
# 2 ECSログの収集

このページでは、**Log Service** コンソールで **Elastic Compute Service (ECS)** のログを収集するよう**Logtail** の設定方法について説明します。

**設定手順**

1. サーバーに **Logtail** をインストールします。
2. **Logtail** マシングループを設定します。
3. **Logtail** を作成し、マシングループに適用します。

**図 2-1 : 設定手順**




**前提**

- **ECS** と**Log Service** が有効化されていること。
- **Project** と **Logstore** を作成していること。 詳細については、「準備」をご参照ください。

> 注 :
> **ECS** インスタンスがクラシックネットワークまたは 仮想プライベートクラウド **(VPC)** に接続されている場合、**ECS** インスタンスと **Log Service** プロジェクトは同じリージョンに属している必要があります。

- **ECS** インスタンスが別の **Alibaba Cloud** アカウントで作成されている場合、**ECS** インスタンスの所有者情報は自動的に取得できません。 この場合、*ECS インスタンスに AliUid を設定* する必要があります。

### ステップ 1：Logtail のインストール

1. インストールコマンドを実行します。

   **ECS** インスタンスリージョンに合った **Logtail** インストールスクリプトを選択します。 詳細については、「*#unique_3*」および「*#unique_2*」をご参照ください。

   たとえば、中国 (杭州) リージョンのクラシックネットワークで稼働中の **Linux ECS** インスタンスに **Logtail** をインストールするには、次のコマンドを実行します。

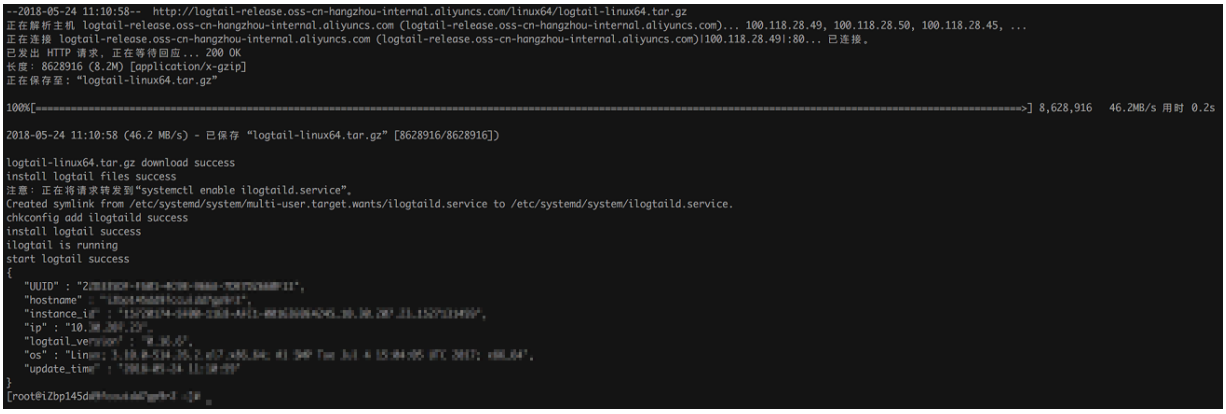
   ```
   wget http://logtail-release.oss-cn-hangzhou-internal.aliyuncs.com
   /linux64/logtail.sh; chmod 755 logtail.sh; sh logtail.sh install
   cn_hangzhou
   ```

2. **Logtail** の稼働状況を確認するには、次のコマンドを実行します。

   ```
   /etc/init.d/ilogtaild status
   ```

   **Logtail** の稼働状況をチェックする際、`ilogtail is running` が表示されれば、インストールに成功したことになります。

**図** 2-2 : Logtail **のインストール**



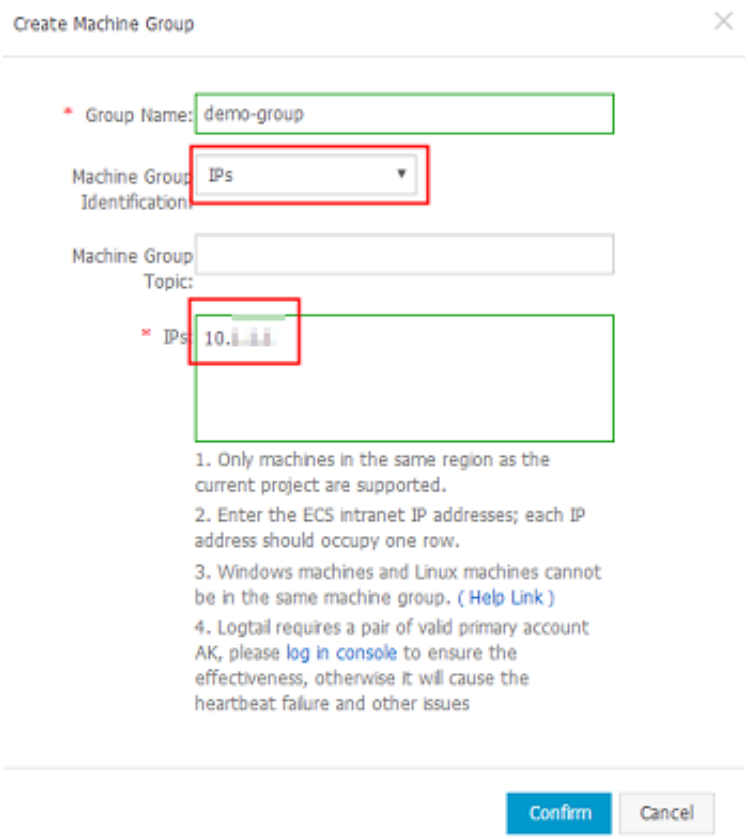

### ステップ 2：マシングループの設定

1. **Log Service** コンソールで、プロジェクトを作成します。
2. **Logstores** で、左側のナビゲーションウィンドウの **[Logtail Machine Group]** をクリックします。
3. マシングループ ページで **[マシングループの作成]** をクリックします。

4. ダイアログボックスが表示されたら、ご使用の **ECS** イントラネット**IP** アドレスとカスタム **ID** を入力して、[確認] をクリックします。

> 注：
>
> - **Log Service** プロジェクトと同一リージョンの **ECS** インスタンスのみがサポートされます。
> - **Log Service** プロジェクトと同一リージョンの **ECS** インスタンスのみがサポートされます。

**図 2-3 : マシングループの設定**



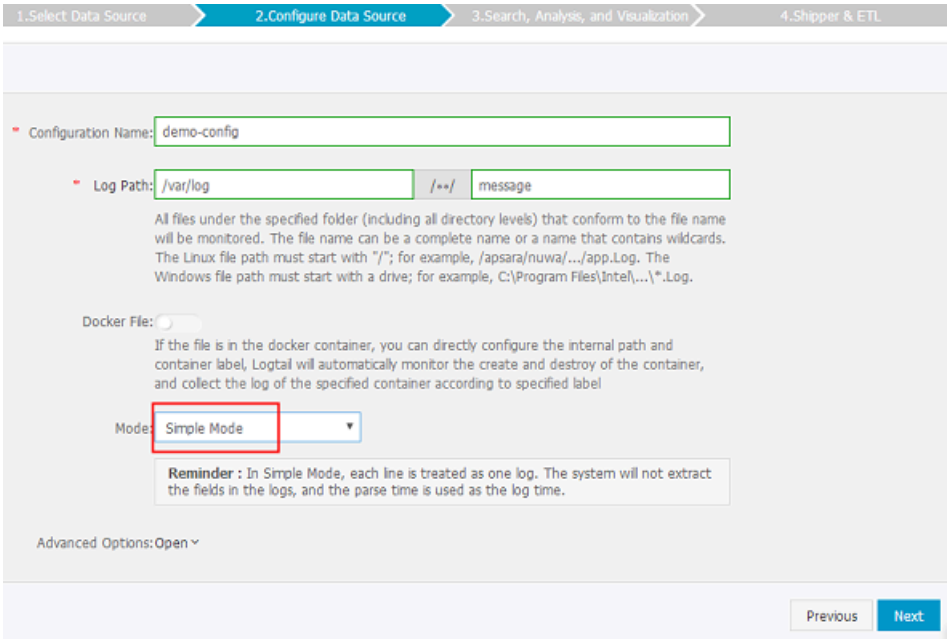

**ステップ 3：Logtail Config の作成**

1. **Logstores** ページで、目的とする **Logstore** を見つけて [データインポートウィザード] アイコンをクリックします。
2. [データソースの選択] タブページで、[カスタムデータ] の [テキストファイル] をクリックします。

3. データソースを設定します。 詳細については、「テキストログの収集」をご参照ください。 このページでは、例として シンプルモードを使用します。

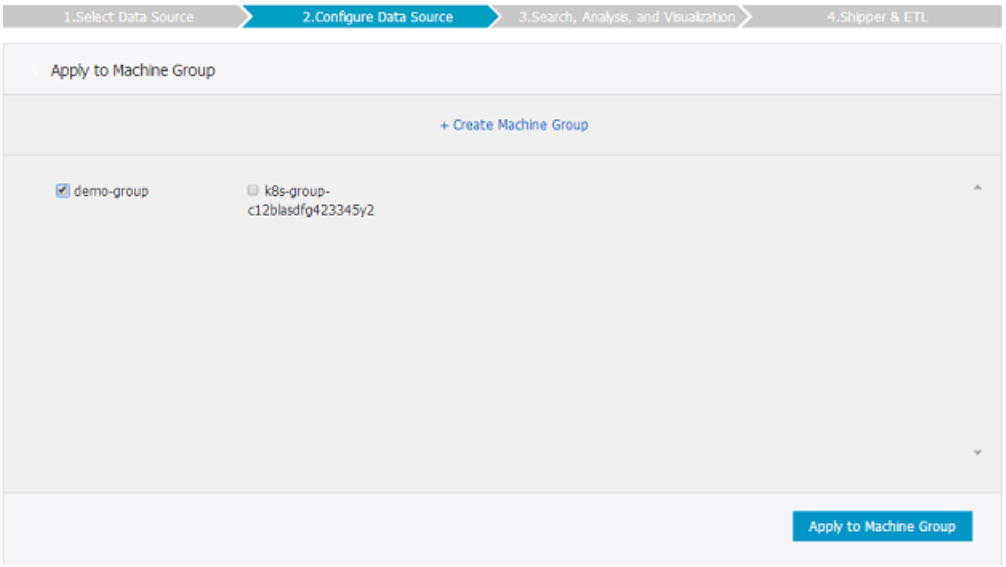
   [ログのパス] テキストボックスに **ECS** ログのパスを入力して [次へ] をクリックします。

   **図** 2-4 : **シンプルモード**




4. ステップ **2** で作成したマシングループを選択し、[マシングループに適用] をクリックします。

   **図** 2-5 : **作成したマシングループへのデータソースの適用**




**Logtail** を使用して **ECS** ログを収集できるようになります。 以下は、収集ログのインデックス設定およびログ送信のための手順です。

### ログの表示

**ECS** コンソールにログインするか、`echo "test message" >> /var/log/message` コマンドを実行します。 新しいログはローカルディレクトリ `/ var / log / message` に生成されます。 **Logtail** は新しいログを収集し、**Log Service** に送信します。

**Logstores** ページで目的とする **Logstore** を見つけて **[検索]** または **[プレビュー]** をクリックして **Logtail** による収集ログを表示します。

**図 2-6 : ログの表示**

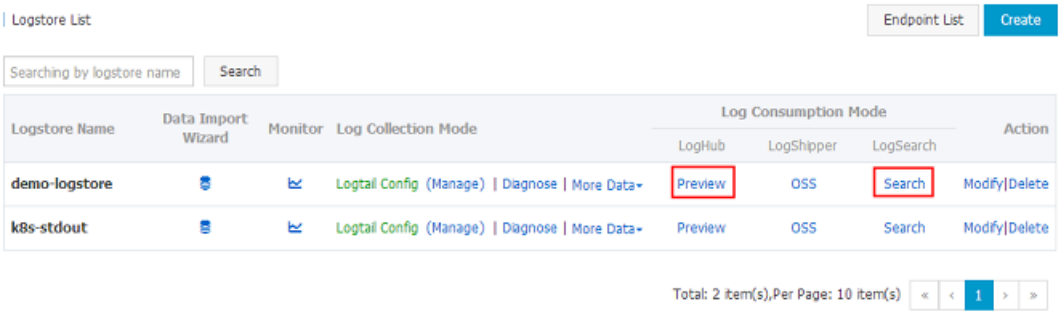



**図 2-7 : ログのプレビュー**

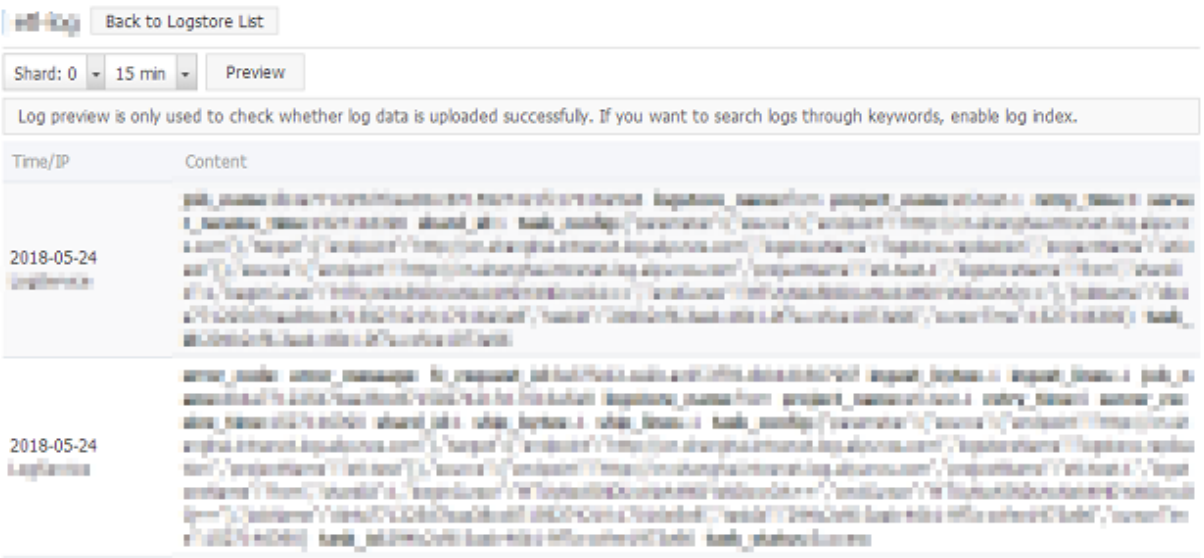
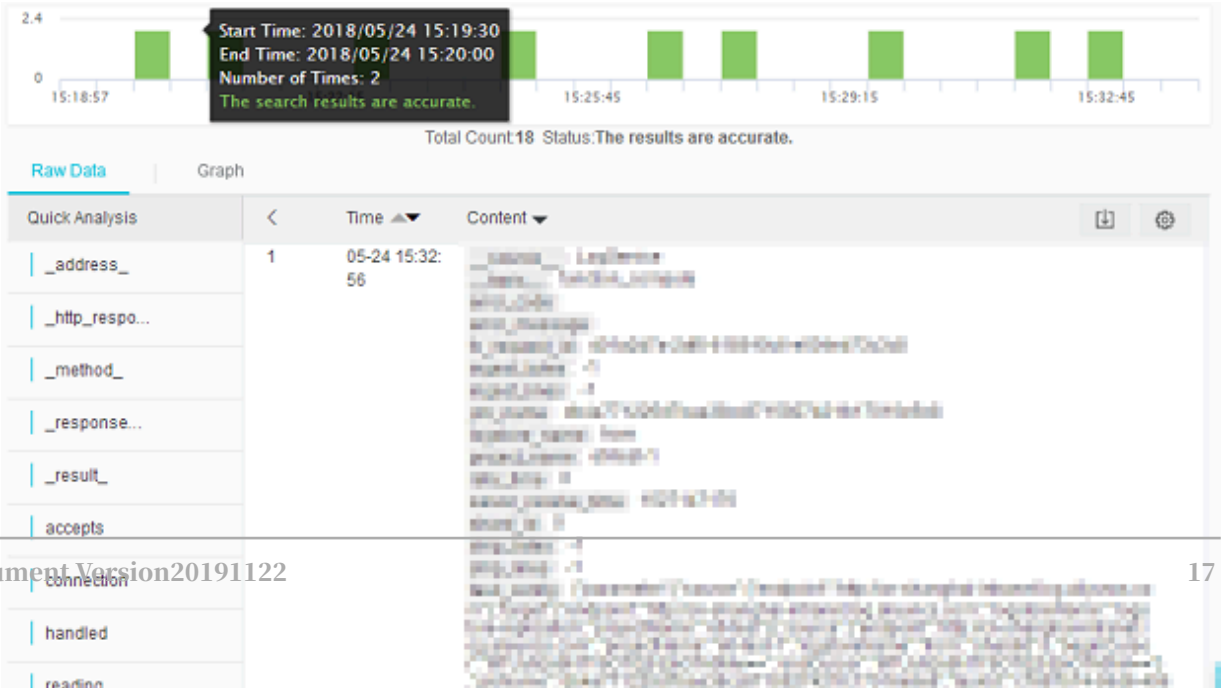



**図 2-8 : ログの取得**



# 3 Kubernetes ログの収集

**Log Service** では、**Logtail** で **Kubernetes** クラスタのログを収集することができます。ログの収集条件は、**CustomResourceDefinition (CRD) API** で設定します。本ドキュメントでは、**Logtail** をインストールし、**Logtail** で **Kubernetes** クラスタログを収集する方法について説明します。

**手順**

1. **alibaba-log-controller Helm** パッケージをインストールします。
2. 収集条件を設定します。

   **Log Service** コンソールや **CDR API** を使用して収集条件を設定します。次の手順に従い、コンソールで収集条件を設定します。

**図** 3-1 : **手順**

1.**パッケージをインストール**

1. **Container Service for Kubernetes** のマスターノードにログインします。

   ログイン方法については、「*Kubernetes クラスターへの SSH キーペアを用いたアクセス*」をご参照ください。
2. パラメータを置き換え、次のコマンドを実行します。

   次のインストールコマンドの `${your_k8s_cluster_id}`をお客様の **Kubernetes** クラスタ **ID** に置き換えて実行します。

```
wget http://logtail-release.oss-cn-hangzhou.aliyuncs.com/linux64/
alicloud-log-k8s-install.sh -O alicloud-log-k8s-install.sh; chmod
744 ./alicloud-log-k8s-install.sh; sh ./alicloud-log-k8s-install.sh
 ${your_k8s_cluster_id}
```

インストールの例

インストールコマンドを実行すると、次のように表示されます。

```
[root@iZbp******biaZ ~]# wget http://logtail-release.oss-cn-hangzhou.
aliyuncs.com/linux64/alicloud-log-k8s-install.sh -O alicloud-log-k8s-
install.sh; chmod 744 ./alicloud-log-k8s-install.sh; sh ./alicloud-log
-k8s-install.sh c12ba20**************86939f0b
....
....
....
alibaba-cloud-log/Chart.yaml
```

```
alibaba-cloud-log/templates/
alibaba-cloud-log/templates/_helpers.tpl
alibaba-cloud-log/templates/alicloud-log-crd.yaml
alibaba-cloud-log/templates/logtail-daemonset.yaml
alibaba-cloud-log/templates/NOTES.txt
alibaba-cloud-log/values.yaml
NAME: alibaba-log-controller
LAST DEPLOYED: Wed May 16 18:43:06 2018
NAMESPACE: default
STATUS: DEPLOYED


RESOURCES:
==> v1beta1/ClusterRoleBinding
NAME AGE
alibaba-log-controller 0s


==> v1beta1/DaemonSet
NAME DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR AGE
logtail 2 2 0 2 0 0s


==> v1beta1/Deployment
NAME DESIRED CURRENT UP-TO-DATE AVAILABLE AGE
alibaba-log-controller 1 1 1 0 0s


==> v1/Pod(related)
NAME READY STATUS RESTARTS AGE
logtail-ff6rf 0/1 ContainerCreating 0 0s
logtail-q5s87 0/1 ContainerCreating 0 0s
alibaba-log-controller-7cf6d7dbb5-qvn6w 0/1 ContainerCreating 0 0s


==> v1/ServiceAccount
NAME SECRETS AGE
alibaba-log-controller 1 0s


==> v1beta1/CustomResourceDefinition
NAME AGE
aliyunlogconfigs.log.alibabacloud.com 0s


==> v1beta1/ClusterRole
alibaba-log-controller 0s




[SUCCESS] install helm package : alibaba-log-controller success.
```

`helm status alibaba-log-controller`のように「**helm status**」でポッドのステータスを確認します。「**Running**」ステータスであれば、インストールに成功していることになります。

**Log Service** には、名前が**k8s-log**で始まるプロジェクトが作成されます。作成されたプロジェクト**k8s-log**を **Log Service** コンソールでキーワード検索します。

2.**収集条件を設定**

**Logstore** を作成し、すべての **K8** コンテナから標準出力 **(stdout)** を収集する手順は、次のとおりです。

1. **Logstore** リストページに移動

   ステップ **1** で作成したプロジェクトをクリックして **Logstore** リストページに移動します。

2. **Logstore** の作成

   右上にある作成をクリックし、表示されるダイアログボックスで **Logstore** を作成します。

   **図** 3-2 : Logstore **の作成**

3. 収集情報の設定

   **a.** データインポートウイザードページへ移動します。

   **b.** サードパーティー製のソフトウェアから**Docker Stdout**を選択します。

   設定ページのマシングループに適用をクリックします。全コンテナの全 **stdout** ファイルが収集されます。

   **図** 3-3 : Docker stdout

4. 設定をマシングループに適用

   マシングループに適用するページで、マシングループを選択して、次へをクリックします。

   **図** 3-4 : **設定をマシングループに適用**

以上で収集条件の設定の完了です。 続けて、インデックス作成とログ送信の設定を行います。なお、ページを終了して設定を完了することもできます。

**収集ログの表示**

設定した収集条件に基づいて、クラスタ内のコンテナが **stdout** 入力を受信すると、**1** 分後に**stdout** ログを収集します。 **Logstore** リストページで、表示をクリックすると収集されたログ

はすばやく表示されます。また検索をクリックして検索条件を指定して、ログを検索し分析することができます。

**図 3-5 : 表示と検索**

下図の検索ページのように、ログの任意のキーワードをクリックすると、すばやく検索できます。特定のログを検索するには、検索ボックスにキーワードを指定します。

**図 3-6 : ログの検索**

### 他の方法による収集条件を設定

他の方法で収集条件を設定するには、以下をご参照ください。

### コンソールで設定

コンソール設定の詳細については、以下をご参照ください。

- コンテナのテキストログ *(推奨)*
- コンテナの標準出力 *(推奨)*
- *host* テキストファイル

  デフォルトでは、**Logtail** コンテナの「*/logtail_host*」ディレクトリに、ホストのルートディレクトリがマウントされています。 パスを設定する際に、このプレフィックスを追加します。 たとえば、ホストの */home/logs/app_log/*ディレクトリのデータを収集するには、設定ページでログのパスを*/logtail_host/home/logs/app_log/*に指定します。

### CRD の設定

**CRD (CustomResourceDefinition)** 設定の詳細については、「*CRD に Kubernetes ログ収集を設定*」をご参照ください。

# 4 Log4j/Logback/Producer ライブラリ

近年、ステートレスプログラミング、コンテナ、サーバレスプログラミングの登場により、ソフトウェアの配信とデプロイの効率が大幅に向上しました。 アーキテクチャの進化に伴い、次の変化がみられます。

- アプリケーションのアーキテクチャは、単一のシステムから、マイクロサービス間の呼び出しとリクエストに移行しています。
- 従来型の物理サーバーから、仮想リソースに移行しています。

**図** 4-1 : **図** 1.**アーキテクチャの進化**

上述の **2** つの変化より、柔軟な標準化されたアーキテクチャの、運用管理 **(O**&**M)** と診断の要件がますます複雑化していることを示しています。 **10** 年前は、サーバーにログインすればすぐにログを取得することができました。 しかし、アタッチプロセスモードは、もはや存在しません。 現在直面しているのは、標準化されたブラックボックスです。

**図** 4-2 : **図** 2. **トレンドの変化**

この変化に対応するために、**DevOps** 向けに、一連の診断および分析ツールが登場しました。 集中モニタリング、集中ログシステム、さまざまな **SaaS** 導入、モニタリング、およびその他のサービスがあります。

ログを一元化すると、これらの問題が解決します。 これを行うために、アプリケーションがログを生成したら、ログはリアルタイム (または準リアルタイム) に中央ノードサーバーに送信されます。 しばしば、**Syslog**、**Kafka**、**ELK**、および **HBase** を使用して集中ストレージを実行します。

**一元管理の利点**

- 使いやすさ：**Grep** を使用して、ステートレスアプリケーションログを照会するのは面倒です。 集中型ストレージでは、以前の長いプロセスが、検索コマンドを実行することによって置き換えられます。
- 独立したストレージとコンピューティング：マシンのハードウェアをカスタマイズするときは、ログ用のストレージスペースを考慮する必要はありません。

- コストの削減：集中型ログストレージでは、より多くのリソースを予約するためにロードシフティングを実行できます。
- セキュリティ：ハッカーの侵入や災害の場合、重要なデータは証拠として保持されます。

**図** 4-3 : **図** 3. **集中化の利点**

### Collector (Java **系列**)

**Log Service** は、サーバー、モバイル端末、組み込み機器、およびさまざまな開発言語向けに **30** 以上のデータ収集方法と包括的なアクセスソリューションを提供します。 **Java** 開発者は、使い慣れたログフレームワークである **Log4j**、**Log4j2**、および **Logback Appender** が必要です。

**Java** アプリケーションには現在、主に **2** つのログ収集ソリューションがあります。

- **Java** プログラムは、ログをディスクにフラッシュし、**Logtail** をリアルタイム収集に使用します。
- **Java** プログラムは、 **Log Service** によって提供される **Appender** を直接設定します。 プログラムが実行されると、ログはリアルタイムで **Log Service** に送信されます。

**2** つの違い

| | ログをディスクに書き出す + **Logtail** を使ってログを収集する | 直接送信に **Appender** を使用する |
|---|---|---|
| 適時性 | ログはファイルに書き込まれ、**Logtail** を使用して収集される。 | ログは **Log Service** に直接送信される。 |
| スループット | 大きい | 大きい |
| 再開可能なアップロード | サポート。 **Logtail** の設定によって異なる。 | サポート。 メモリサイズによって異なる。 |
| アプリケーションの場所にセンシティブ | 収集マシングループを設定するときに必要。 | 不要。 ログは自発的に送信される。 |
| ローカルログ | サポート | サポート |
| 収集を無効にする | **Logtail** の設定を削除。 | **Appender** の設定を変更してアプリケーションを再起動。 |

**Appender** を使用すると、**Config** を使用して、コードを変更せずにリアルタイムなログ収集を簡単に完了できます。**Log Service** によって提供される **Java** 系列 **Appender** には、次の利点があります。

- プログラムを変更することなく、設定変更が反映される
- 非同期 +ブレークポイントの送信 ー **I/O** はメインスレッドに影響を与えることなく、ネットワーク障害およびサービス障害への耐障害
- 高度に並行実行 ー 大量のログ書き込みに対応
- コンテキスト照会が可能 – **Log Service** の元のプロセスで、ログのコンテキスト (ログの前後の **N** 個のログ) を正確に復元できます。

### Appender の概要と使い方

以下の **Appender** が用意されています。データの書き込みには、すべて **aliyun-log-producer -java** が使用されています。

- *aliyun-log-log4j-appender*
- *aliyun-log-log4j2-appender*
- *aliyun-log-logback-appender*

相違点

| **Appender** 名 | 説明 |
| --- | --- |
| **aliyun-log-log4j-appender** | **Log4j 1.x** 用 **Appender** (アプリケーションがログのフレームワークに **Log4j 1.x** を採用している場合の推奨 **Appender**) |
| **aliyun-log-log4j2-appender** | **Log4j 2.x** 用 **Appender** (アプリケーションがログのフレームパークに **Log4j 2.x** を採用している場合の推奨 **Appender**) |
| **aliyun-log-logback-appender** | **Logback** 用 **Appender** (アプリケーションがログフレームワークに **Logback** を採用している場合の推奨 **Appender**) |
| **aliyun-log-producer-java** | **Java** アプリケーション向けの **LogHub** クラスライブラリで、並行書き込みに使用されます。上記の **Appender** はすべて、本 **Appender** を使用してデータを書き込みます。なお、**LogHub** に書き込まれるデータのフィールドとフォーマットを指定することもできます。上記 **Appender** が要件を満たさない場合は、本 **Appender** を使用してログ収集プログラムを開発します。 |

**手順** 1. Appender **と接続**

「*aliyun-log-log4j-appender*」に記載の手順に従って、**Appender** と接続します。

設定ファイル`log4j.properties`の内容は次のとおりです。

```
log4j.rootLogger=WARN,loghub
log4j.appender.loghub=com.aliyun.openservices.log.log4j.LoghubAppender
# Log Service project name (required parameter)
log4j.appender.loghub.projectName=[your project]
# Log Service LogStore name (required parameter)
log4j.appender.loghub.logstore=[your logstore]
#Log Service HTTP address (required parameter)
log4j.appender.loghub.endpoint=[your project endpoint]
#(Mandatory) User identity
log4j.appender.loghub.accessKeyId=[your accesskey id]
log4j.appender.loghub.accessKey=[your accesskey]
```

**手順** 2. **照会/分析**

上記の手順どおりに **Appender** を設定すると、**Java** アプリケーションの生成するログは自動的に **Log Service** に送信されます。*LogSearch/Analytics*で、リアルタイムにログを照会/分析できます。例で使用するログフォーマットは次のとおりとなります。

- ログイン操作が記録されたログ

```
level: INFO
location: com.aliyun.log4jappendertest.Log4jAppenderBizDemo.login(
Log4jAppenderBizDemo.java:38)
message: User login successfully. requestID=id4 userID=user8
thread: main
time: 2018-01-26T15:31+0000
```

- 購入操作が記録されたログ

```
level: INFO
location: com.aliyun.log4jappendertest.Log4jAppenderBizDemo.order(
Log4jAppenderBizDemo.java:46)
message: Place an order successfully. requestID=id44 userID=user8
itemID=item3 amount=9
thread: main
time: 2018-01-26T15:31+0000
```

**手順** 3. **照会/分析を有効化**

データを照会/分析する前に、照会/分析機能を使用可能にします。次の手順に従って機能を有効にします。

1. **Log Service** コンソールにログインします。
2. プロジェクト一覧ページで、プロジェクト名をクリックします。
3. **Logstore** の右側の検索をクリックします。
4. 右上の有効化 > 変更を順にクリックします。

5. 既にインデックスを有効にしている場合は、インデックス属性 > 変更を順にクリックします。検索/分析ページが表示されます。

**図 4-4 : 図 4. クエリフィールドの指定**

**手順 4. ログ分析**

1. 1 時間以内にエラーが最も発生した上位 3 箇所を出力

```
level: ERROR | select location ,count(*) as count GROUP BY location
ORDER BY count DESC LIMIT 3
```

2. 直近 15 分以内のログレベルごとに、生成されたログの数を算出

```
| select level ,count(*) as count GROUP BY level ORDER BY count DESC
```

3. ログ内容を照会

   どのログに対しても、元のログファイルのコンテキスト情報を正確に再構築できます。 詳細については、「コンテキスト照会」をご参照ください。

4. 1 時間以内にログイン回数の最も多かったユーザー 3 名を出力

```
Login | select maid (message, 'userid = (? <userID>[a-zA-Z\d]+)', 1
) AS userID, count(*) as count GROUP BY userID ORDER BY count DESC
LIMIT 3
```

5. 各ユーザーの直近 15 分以内の支払い総額を算出

```
order | SELECT regexp_extract(message, 'userID=(? <userID>[a-zA-Z\
d]+)', 1) AS userID, sum(cast(regexp_extract(message, 'amount=(? <
amount>[a-zA-Z\d]+)', 1) AS double)) AS amount GROUP BY userID
```

# 5 分析 - Nginx アクセスログ

**Nginx** サーバーで **Web** サイトを構築する **Web** 管理者は数多くいます。**Nginx** アクセスログを統計分析し、**Web** サイトのページビューやアクセス時間といったデータを取得して **Web** サイトのトラフィックデータを解析します。**CNZZ (**中国のアクセス解析ツール**)** といった従来の方法では、**Web** サイトに **js** を挿入しておき、ユーザーがサイトにアクセスすると **js** を起動させています。しかし、この方法ではアクセスリクエストしか記録されません。 ストリーム処理やオフライン統計分析で **Nginx** アクセスログを分析することもできますが、環境構築に手間がかかり、適時に柔軟な分析を行うことが難しくなります。

**Log Service** ではログのリアルタイム照会、分析することができます。また、分析結果がダッシュボードに表示されるため、**Nginx** のアクセスログの複雑な解析を大幅に軽減することができ、簡単に **Web** サイトアクセスデータの統計を出すことができます。 本ドキュメントでは、**Nginx** のアクセスログを分析して、ログ分析の詳細な手順をご紹介します。

### 利用イメージ

**Nginx** をサーバーで、個人の **Web** サイト構築し、サイトのアクセス状況を把握するために **Nginx** のアクセスログを分析し、**PV** 数、**UV** 数、アクセスの多い **Web** ページ、使用率の高いリクエストメソッド、不正リクエスト、クライアント情報、および **Web** サイトのリファラー一覧を取得するものとします。

### ログフォーマット

分析シナリオに適当な、次の`log_format`を使用されることを推奨します。

```
    log_format main '$remote_addr - $remote_user [$time_local] "$
request" $http_host '
                    '$status $request_length $body_bytes_sent "$
http_referer" '
                    '"$http_user_agent" $request_time $upstream_r
esponse_time';
```

各フィールドの説明は次のとおりです。

| フィールド | 意味 |
| --- | --- |
| **remote_addr** | クライアントアドレス |
| **remote_user** | クライアントユーザー名 |
| **time_local** | サーバー時間 |
| **request** | メソッド名、アドレス、および **HTTP** プロトコルを含むリクエストコンテンツ |

| フィールド | 意味 |
| --- | --- |
| **http_host** | ユーザーリクエストで使用される **HTTP** アドレス |
| **Status** | 応答 **HTTP** ステータスコード |
| **request_length** | リクエストサイズ |
| **body_bytes_sent** | 応答の **Byte** 数 |
| **http_referer** | リファラー (参照元) |
| **http_user_agent** | クライアント名 |
| **Request_time** | リクエスト処理待ちの総時間 |
| **upstream_response_time** | アップストリームサービス処理待ち時間 |

**手順**

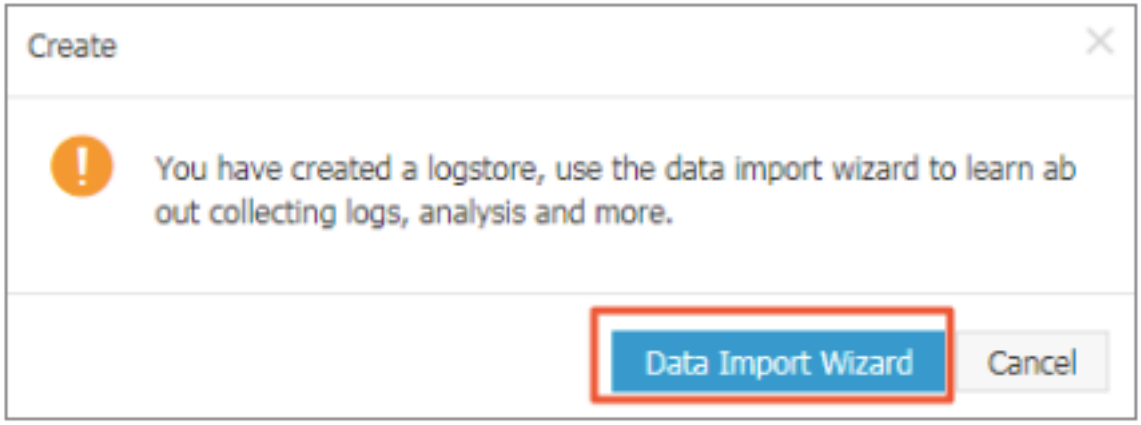
1.**データインポートウィザードを開く**

**Log Service** のデータインポートウィザードにより、データソースに迅速にアクセスできます。次のいずれかの方法でデータインポートウィザードを起動し、**Log Service** で **Nginx** のアクセスログを収集します。

・ プロジェクトの作成

**Logstore** を作成したら、既存のプロジェクトまたは新たに作成したプロジェクトのデータインポートウィザードをクリックします。

**図** 5-1 : **データインポートウィザード**



・ 既存の **Logstore** であれば、**Logstore** リストページより該当 **Logstore** のデータインポートウィザードアイコンをクリックします。

**図** 5-2 : Logstore **リスト**

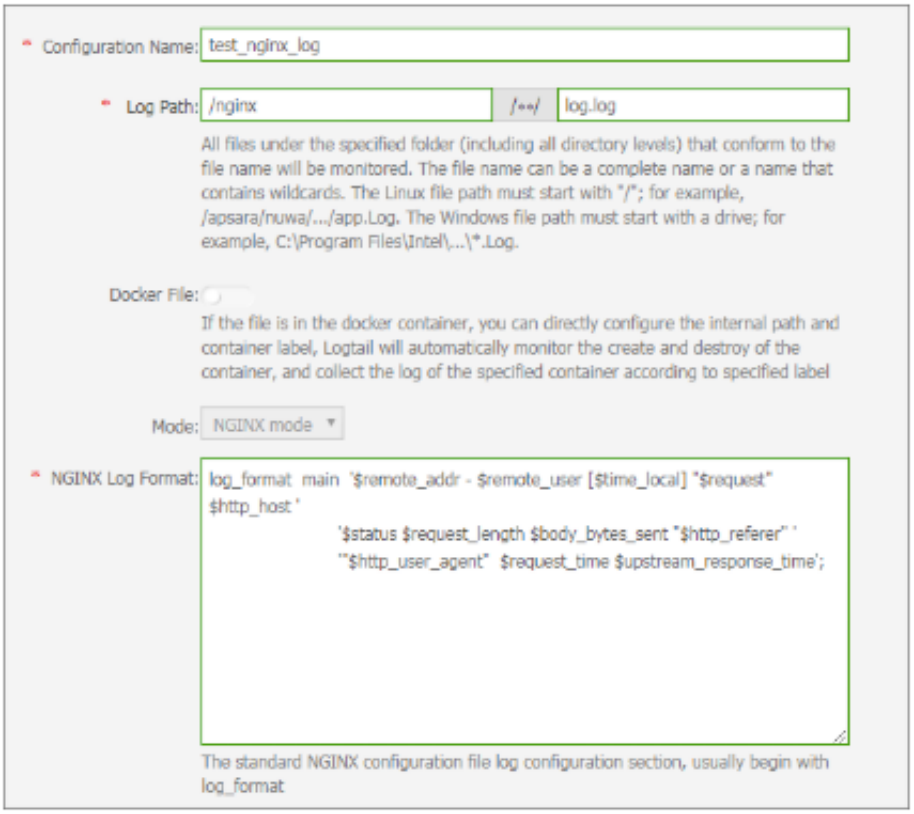
2.**データソースを選択**

**Log Service** には、クラウドサービス、サードパーティーのソフトウェア、**API**、**SDK** といった、さまざまな種類のデータソースを扱うことができます。**Nginx** アクセスログを分析するには、サードパーティ製ソフトウェア の**NGINX ACCESSLOG** > サードパーティーソフトウェアを選択します。

### 3.データソースを設定

1. ご利用環境に合わせて設定名とログパスを入力します。また、**NGINX** ログフォーマット欄に、推奨の上記`log_format`を入力します。

   **図** 5-3 : **データソースの設定**




**Log Service** は該当するキーを自動的に展開します。

注 :

`$request`は、`request_method`と`request_uri`の 2 つのキーに展開されます。

**図** 5-4 : Nginx **キー**


2. マシングループに適用します。

   マシングループをまだ作成していなければ、まずマシングループを作成します。 マシングループの作成方法については、「マシングループの作成と識別子に *IP* アドレスを指定」をご参照ください。

   注 :
   **Logtail** の設定が有効になるまで、最大で 3 分ほどかかります。

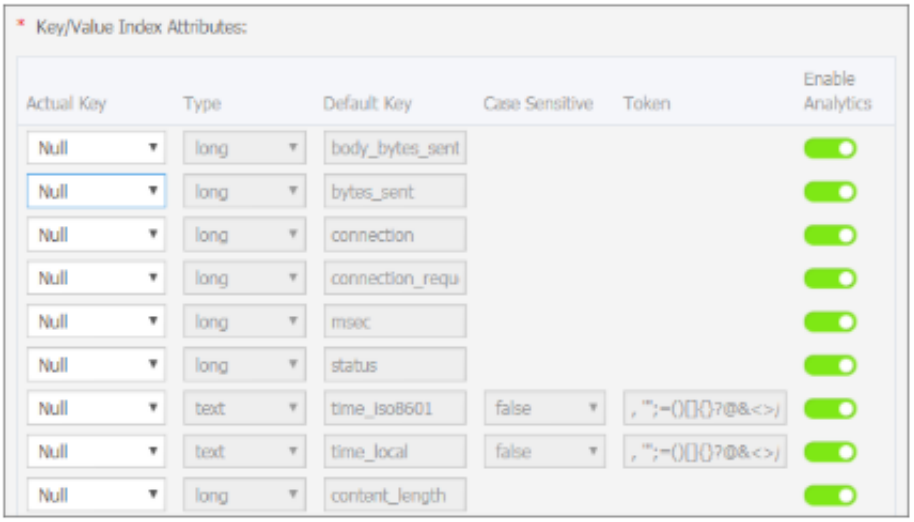
### 4.検索、分析、視覚化

**Logtail** 設定を適用するマシングループのハートビートが正常ステータスであることを確認し、右側のプレビューをクリックして収集データを取得します。

**図 5-5 : プレビュー**


**Log Service** は、分析と使用のために事前定義されたキーを提供します。 実際のキー(プレビューされたデータに従って生成) を選択して、デフォルトのキーとマッピングすることができます。

**図 5-6 : キー/値のインデックス属性**



次へをクリックします。**Log Service** はインデックス属性は自動的に設定し、分析に利用できるよう`nginx-dashboard`ダッシュボードが生成されます。

### 5.アクセスログを分析する

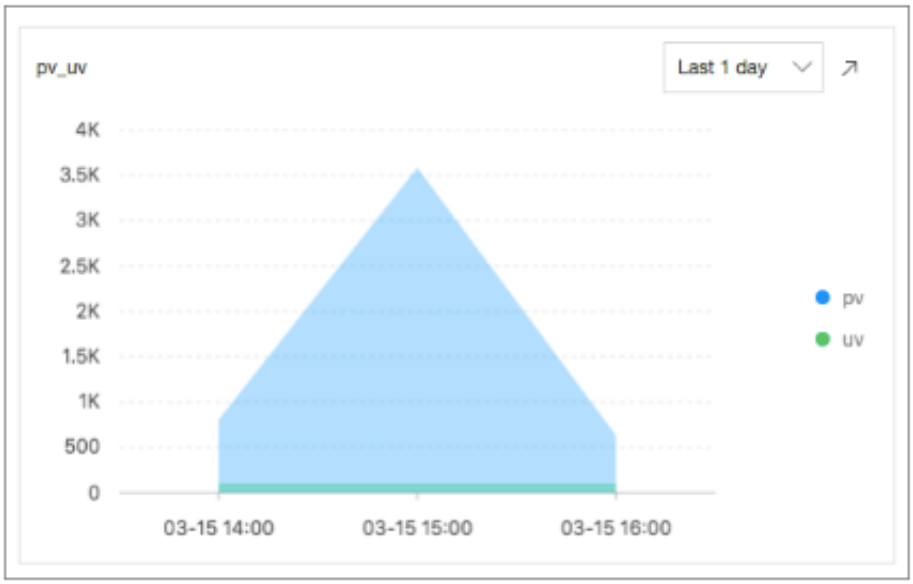
インデックス機能を有効にすると、デフォルトでダッシュボードの生成されるページに、各指標の分析が表示されます。 ダッシュボードの使い方については、「ダッシュボード」をご参照ください。

**図 5-7 : ダッシュボード**

- **PV/UV 統計 (pv_uv)**

  前日の **PV** 数および **UV** 数を集計します。

  **図** 5-8 : PV/UV**統計**




  統計ステートメント

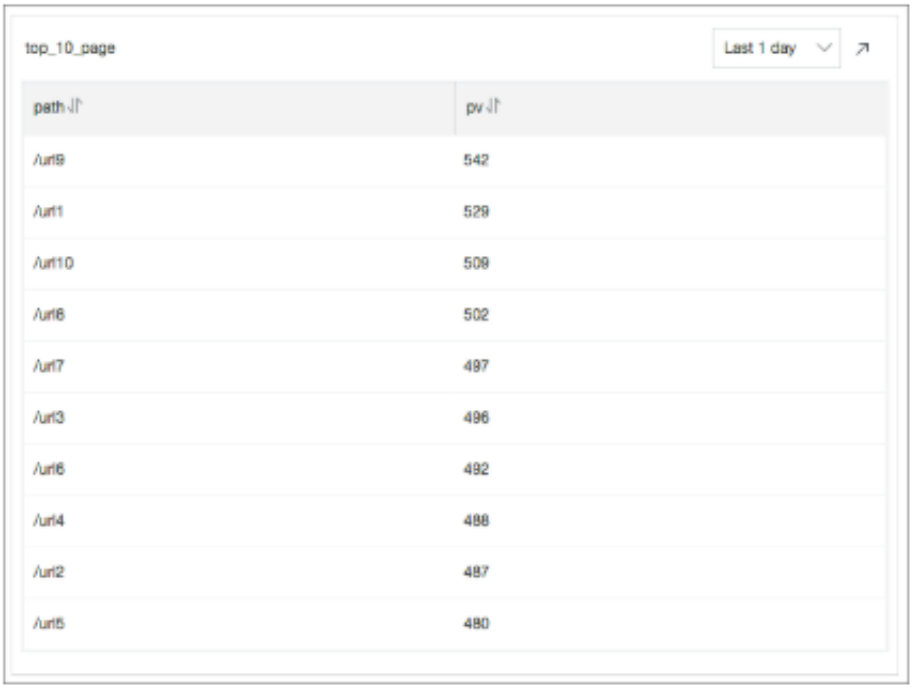
```
  * | select approx_distinct(remote_addr) as uv ,
          count(1) as pv ,
          date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as
 time
          group by date_format(date_trunc('hour', __time__), '%m-%d %
H:%i')
          order by time
```

```
        limit 1000
```

- アクセス数の多いページトップ 10 (top_page)

  前日の、PV 数の最も多かった上位 10 ページを集計します。

  **図 5-9 : アクセス数統計**




  統計ステートメント

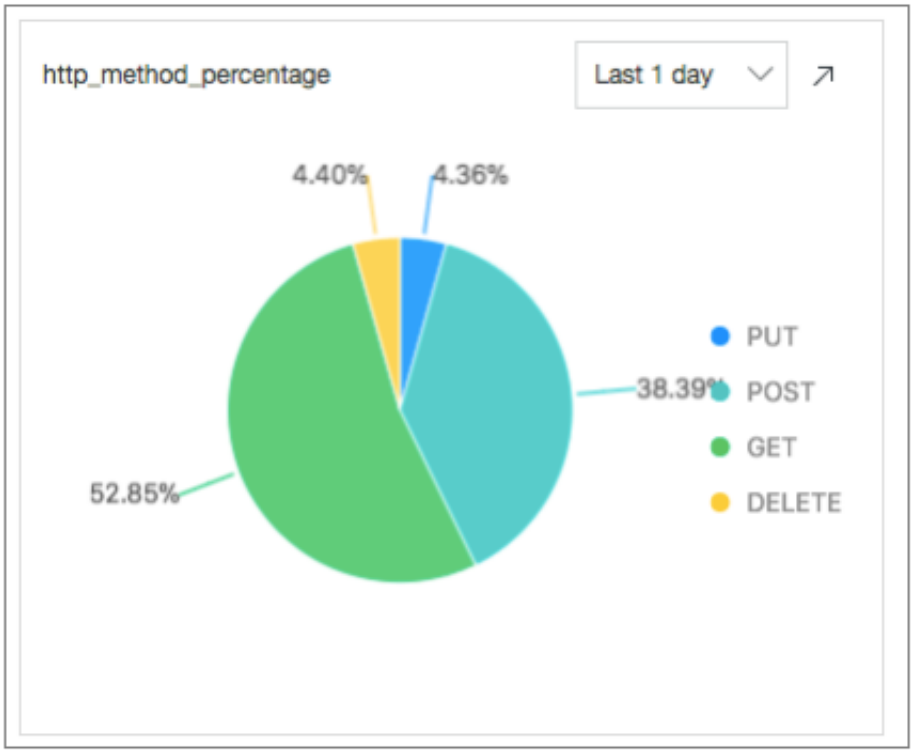
```
* | select split_part(request_uri,'?',1) as path,
    count(1) as pv
    group by split_part(request_uri,'?',1)
```

```
    order by pv desc limit 10
```

- リクエストメソッドの割合を集計 (**http_method_percentage**)

  前日に使用されたリクエストメソッドの割合を集計します。

  **図** 5-10 : **リクエストメソッドの割合**




統計ステートメント

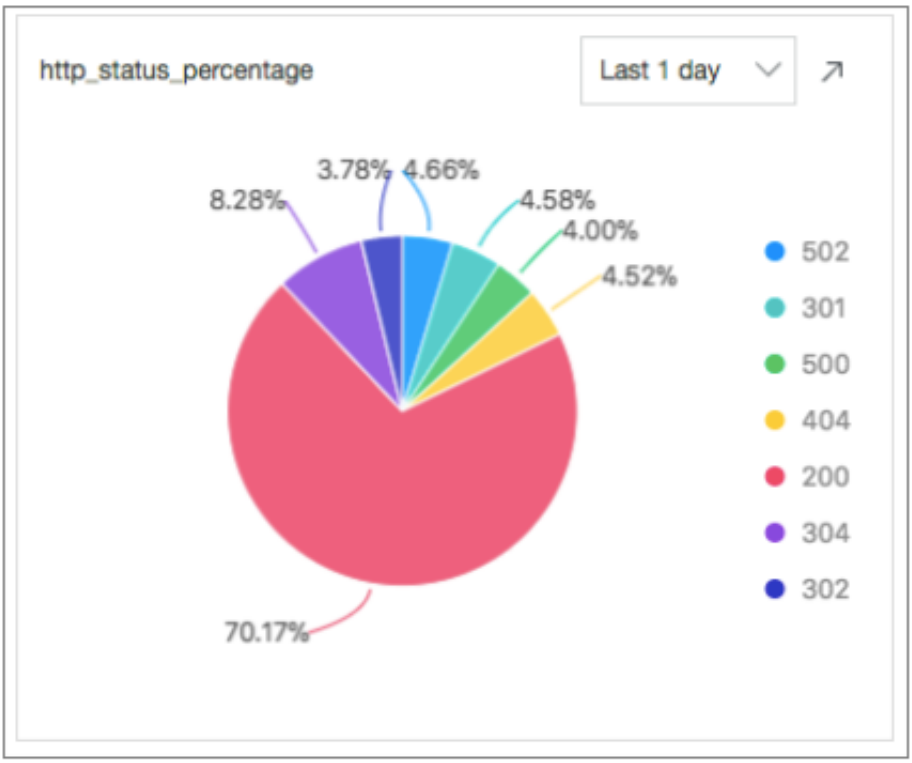
```
* | select count(1) as pv,
        request_method
```

```
        group by request_method
```

- リクエストステータスの割合を集計 (**http_status_percentage**)

  前日の各リクエストステータス(**HTTP**ステータスコード) の割合を集計します。

  **図** 5-11 : **リクエストステータスの割合**




  統計ステートメント

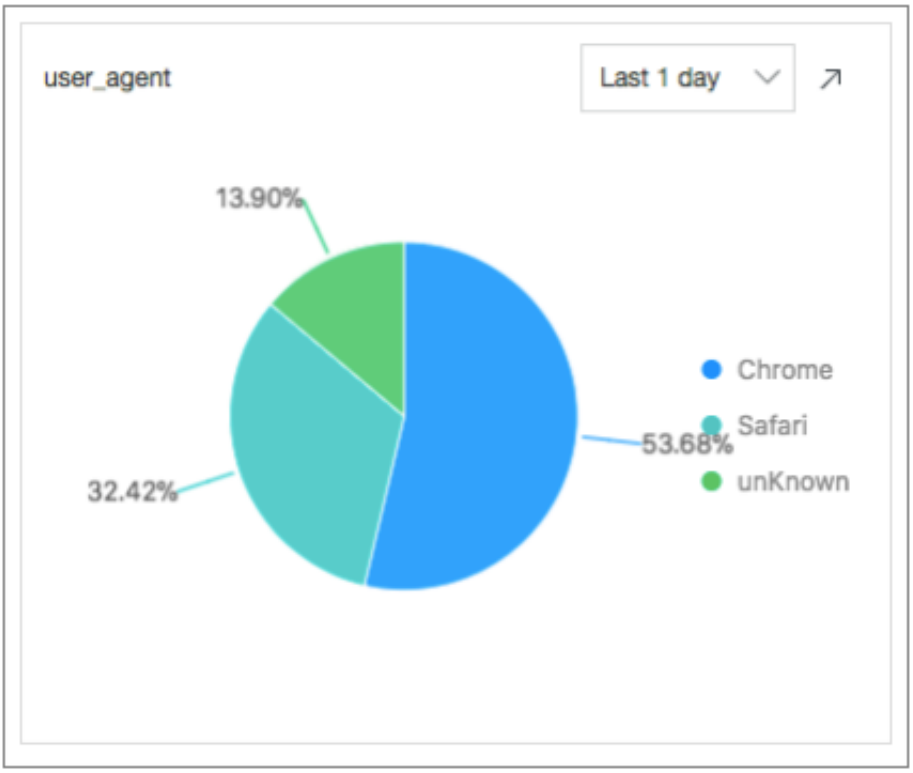
```
* | select count(1) as pv,
        status
```

```
        group by status
```

- リクエスト **UA** の割合を集計 **(user_agent)**

  前日にアクセスに使用されたブラウザの割合を集計します。

  **図** 5-12 : **リクエスト** UA **の割合**




統計ステートメント

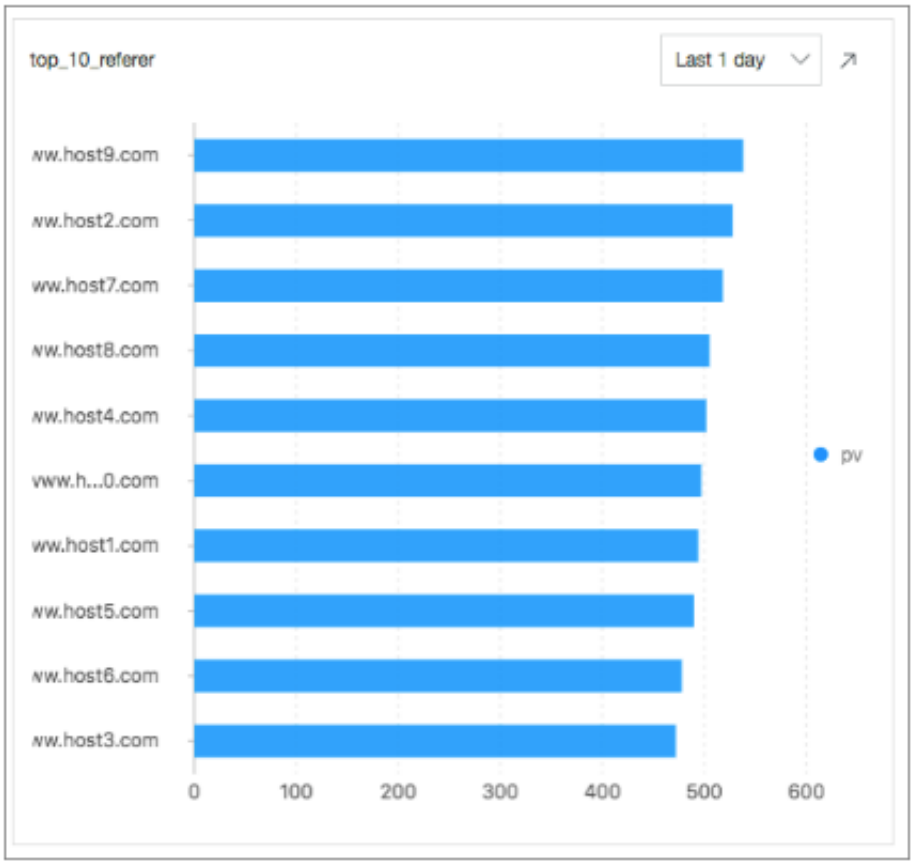
```
* | select count(1) as pv,
    case when http_user_agent like '%Chrome%' then 'Chrome'
    when http_user_agent like '%Firefox%' then 'Firefox'
    when http_user_agent like '%Safari%' then 'Safari'
    else 'unKnown' end as http_user_agent
    group by http_user_agent
    order by pv desc
```

```
    limit 10
```

- リファラートップ 10 を集計 (top_10_referer)

  前日の上位 10 リファラー (参照元) を集計します。

  **図** 5-13 : **リファラートップ** 10 **集計**




  統計ステートメント

```
* | select count(1) as pv,
        http_referer
        group by http_referer
        order by pv desc limit 10
```

### 6.アクセス解析と最適化

**Web** 管理者は、予め用意されているアクセス指標の他、リクエスト処理の待ち時間や、待ち時間の多いページを把握するために、アクセリクエストを分析することもあります。 そういった場合に、クエリページに入力すると迅速にアクセス解析できます。

- 平均レイテンシと最大レイテンシを集計

  平均レイテンシと最大レイテンシを 5 分ごとに設定して、レイテンシの課題を把握します。

  統計ステートメント

```
* | select from_unixtime(__time__ -__time__% 300) as time,
        avg(request_time) as avg_latency ,
        max(request_time) as max_latency
        group by __time__ -__time__% 300
```

- 最大レイテンシでリクエストページを集計

  最大レイテンシを把握できたら、最大レイテンシが発生しているリクエストページを特定してページレスポンスを最適化します。

  統計ステートメント

```
* | select from_unixtime(__time__ - __time__% 60) ,
        max_by(request_uri,request_time)
        group by __time__ - __time__%60
```

- リクエストレイテンシの分布を集計

  Web サイト全体のリクエストレイテンシ分布を集計します。 レイテンシを 10 個のバケットに配置し、各レイテンシの間隔でリクエストの数を確認します。

  統計ステートメント

```
* |select numeric_histogram(10,request_time)
```

- レイテンシトップ 10 を集計

  最大レイテンシだけでなく、2 番目以降 10 番目まで、またその値を集計します。

  統計ステートメント

```
* | select max(request_time,10)
```

- 最大レイテンシの発生しているページを最適化

  `/url2`ページで最大レイテンシが発生しているとします。 `/url2`ページを最適化するには、`/url2`ページのPV、UV、メソッド、ステータス、ブラウザの数や平均レイテンシ、最大レイテンシを集計します。

  統計ステートメント

```
request_uri:"/url2" | select count(1) as pv,
        approx_distinct(remote_addr) as uv,
        histogram(method) as method_pv,
        histogram(status) as status_pv,
        histogram(user_agent) as user_agent_pv,
        avg(request_time) as avg_latency,
```

```
            max(request_time) as max_latency
```

上記のデータを取得することにより、有用な **Web** サイトのアクセス状況を具体的に把握することができます。

# 6 Apache アクセスログの分析

**Log Service** では、ワンストップで **Apache** ログ収集、データインポートウィザードでインデックス作成を行うことができます。デフォルトのダッシュボード、またはクエリ分析ステートメントで、**Web** サイトアクセス情報をリアルタイム分析できます。

- **Log Service** の有効化
- プロジェクトおよび **Logstore** の作成

**Apache** は **Web** サイトの構築に **Web** 管理者の多くが利用するサーバーです。**Web** 管理者は、**Web** アクセス状況を把握するために、**Apache** アクセスログを解析して **PV**、**UV**、**IP** アドレスのリージョン分布、クライアント情報、参照元サイトといった情報を取得します。

**Log Service** では、ワンストップで **Apache** ログ収集、データインポートウィザードでインデックス作成を行うことができます。また、デフォルトで **Apache** ログ用のアクセス解析ダッシュボードが自動生成されます。

分析シナリオに合わせて、**Apache** ログの設定を次のようにすることを推奨します。

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i
\" %D %f %k %p %q %R %T %I %O" customized
```

注：

ログコンテンツに、`%t`、`%{User-Agent}i`、`%{Referer}i`といったスペースを含むフィールドがないかどうかを確認します。 スペースを含むフィールドがある場合は、`\”`で囲むと問題なくログ分析されます。

各フィールドの意味は次のとおりです。

| フィールド | フィールド名 | 説明 |
| --- | --- | --- |
| %**h** | **remote_addr** | クライアント **IP** アドレス |
| %**l** | **remote_ident** | **identd** のクライアント側のログ名。 |
| %**u** | **remote_user** | クライアントユーザー名 |
| %**t** | **time_local** | サーバーの時間 |
| %**r** | **request** | メソッド名、**IP** アドレス、および **http** プロトコルを含むリクエストコンテンツ |
| %>**s** | **status** | 応答 **http** ステータスコード |

| フィールド | フィールド名 | 説明 |
| --- | --- | --- |
| %b | response_size_bytes | 応答のサイズ |
| %{Rererer}i | http\u0008_referer | リファラー (参照元) |
| %{User-Agent}i | http_user_agent | クライアント情報 |
| %D | request_time_msec | リクエスト時間 (ミリ秒単位) |
| %f | filename | リクエストファイル名 (パスを含む) |
| %k | keep_alive | keep-alive リクエスト数 |
| %p | remote_port | サーバーのポート番号 |
| %q | request_query | クエリ文字列 (クエリ文字列がない場合は空文字列) |
| %R | response_handler | サーバー応答のハンドラー |
| %T | request_time_sec | リクエスト時間 (秒単位) |
| %I | bytes_received | サーバーの受信 Byte 数 (要 mod_logio モジュールの有効化) |
| %O | bytes_sent | サーバの送信した Byte 数 (要 mod_logio モジュールの有効化) |

1. *Log Service コンソール*にログインし、プロジェクト名をクリックします。
2. Logstore リストページで該当 Logstore のデータインポートウィザードアイコンをクリックします。
3. データソースにAPACHE アクセスログを選択します。

4. データソースを設定します。

   a) 設定の名前を入力します。

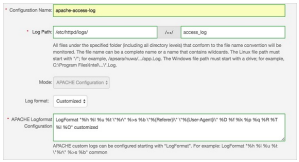
   b) ログパスを入力します。

   c) ログフォーマットを選択します。

   ログフォーマットには、**Apache** ログ設定ファイルに指定されているフォーマットを選択します。容易にログデータを照会/分析するには、**Apache** ログフォーマットのカスタム化を推奨します。

   d) **Apache** のログフォーマット設定を入力します。

   ログフォーマット に 共通 または 混合 を指定した場合は、設定が自動入力されます。カスタムログフォーマットを指定した場合は、設定を入力します。上記 **Log Service** の設定を入力されることを推奨します。



   e) **APACHE** キー名を確認します。

   **Log Service** は **APACHE** キー名を自動的に解決します。キー名は **Web** ページで確認できます。

> 注：
> **%r** より、`request_method`、`request_uri`および`request_protocol`の **3** つのキーが抽出されます。

   f) オプション: オプション: 詳細オプションを設定し、[次へ] をクリックします。

| 設定項目 | 詳細説明 |
| --- | --- |
| ローカルキャッシュ | ローカルキャッシュの有効化/無効化。有効化すると、**Log Service** が利用不可となった場合、ログはマシンのローカルディレクトリにキャッシュされ、**Log Service** 回復したら送信されます。 デフォルトでは、最大 **1GB** がキャッシュされます。 |

| 設定項目 | 詳細説明 |
| --- | --- |
| オリジナルログのアップロード | オリジナルログのアップロードの有効化/無効化。有効化すると、デフォルトでオリジナルログに新規フィールドが追加されます。 |
| トピック生成モード | ・ **NULL - Topic** を生成しない**:** 初期値。トピックには空文字列が設定されています。ログを照会する際にトピックの入力は必要ありません。<br>・ マシングループトピック属性**:** フロントエンドサーバーごとにログデータを区別する場合に適用します。<br>・ ファイルパスの正規表現**:** 本オプションを選択する場合は、カスタム正規表現の設定が必須です。正規表現パスからトピックが抽出されます。 ユーザーやインスタンスごとにログを分ける場合に使用します。 |
| カスタム正規表現 | トピック生成モードにファイルパスの正規表現を選択した場合、正規表現を入力します。 |
| ログファイルのエンコード形式 | ・ **utf8**：**UTF-8** エンコード<br>・ **gbk**：**GBK** エンコード |
| モニタリングディレクトリの最大階層数 | ソースログからログを収集するときにモニタリング対象のディレクトリの最大階層数を指定します。指定可能な値は、**0** ～ **1000** であり、**0** の場合は作業中のディレクトリパスのみがモニタリング対象となります。 |
| タイムアウト | 指定した時間内にログファイルに更新がない場合にタイムアウトしますます。 タイムアウトに、設定可能な値は次のとおりです。<br>・ タイムアウトしない**:** すべてのログファイルを常にモニタリングし、タイムアウトされません。<br>・ **30** 分でタイムアウト**: 30** 分を超えてログファイルが更新されない場合、ログファイルはタイムアウトし、モニタリング対象外となります。 |

| 設定項目 | 詳細説明 |
| --- | --- |
| フィルター設定 | フィルター条件に完全に一致するログのみ収集されます。<br>例:<br>・ 条件を満たすログを収集: `Key:level Regex:WARNING\|ERROR` の場合、**WARNING** ログまたは **ERROR** ログのみが収集されます。<br>・ 条件を満たさないログを収集：<br>- `Key:level Regex:^(?!. *(INFO\|DEBUG))`の場合、**INFO** ログおよび **DEBUG** ログは収集されません。<br>- `Key:url Regex:. *^(?!.*(healthcheck)). *`の場合、**URL** に「**healthcheck**」を含むログは収集されません。たとえば、**key** が「**url**」で、値が`/inner/healthcheck/jiankong.html`のログは収集されません。<br>その他の例については、「*regex-exclude-word*」、「*regex-exclude-pattern*」をご参照ください。 |

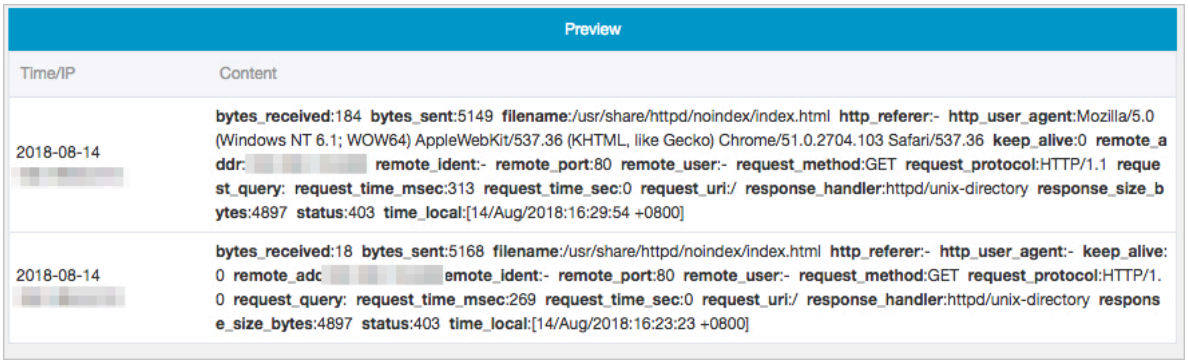
5. マシングループに設定を適用します。

   設定を適用するマシングループを選択します。右下隅のマシングループへ適用をクリックします。

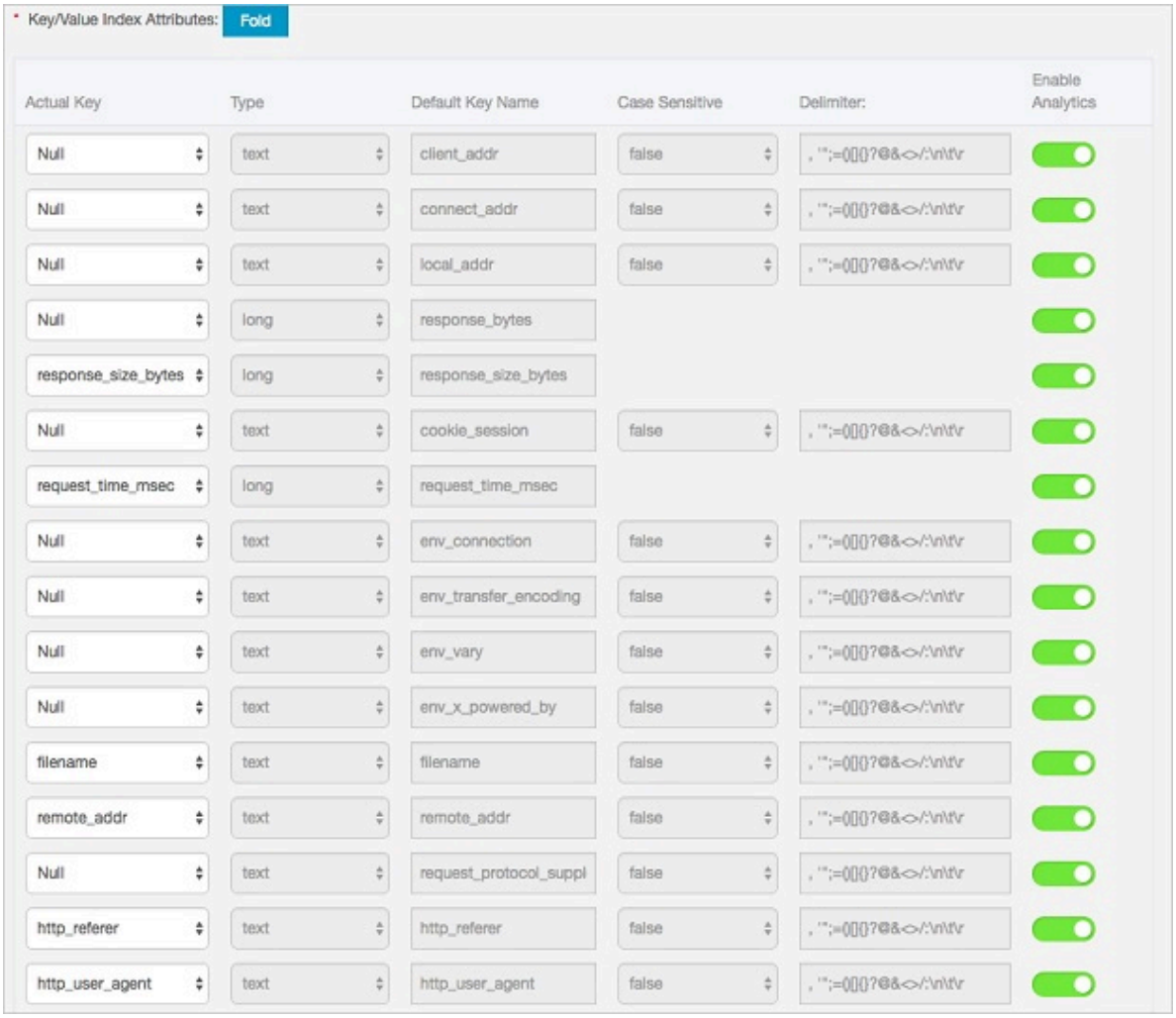
   マシングループをまだ作成していない場合は、マシングループの作成をクリックしてマシングループを作成します。

6. (オプション) クエリ、分析及び可視化を設定します。

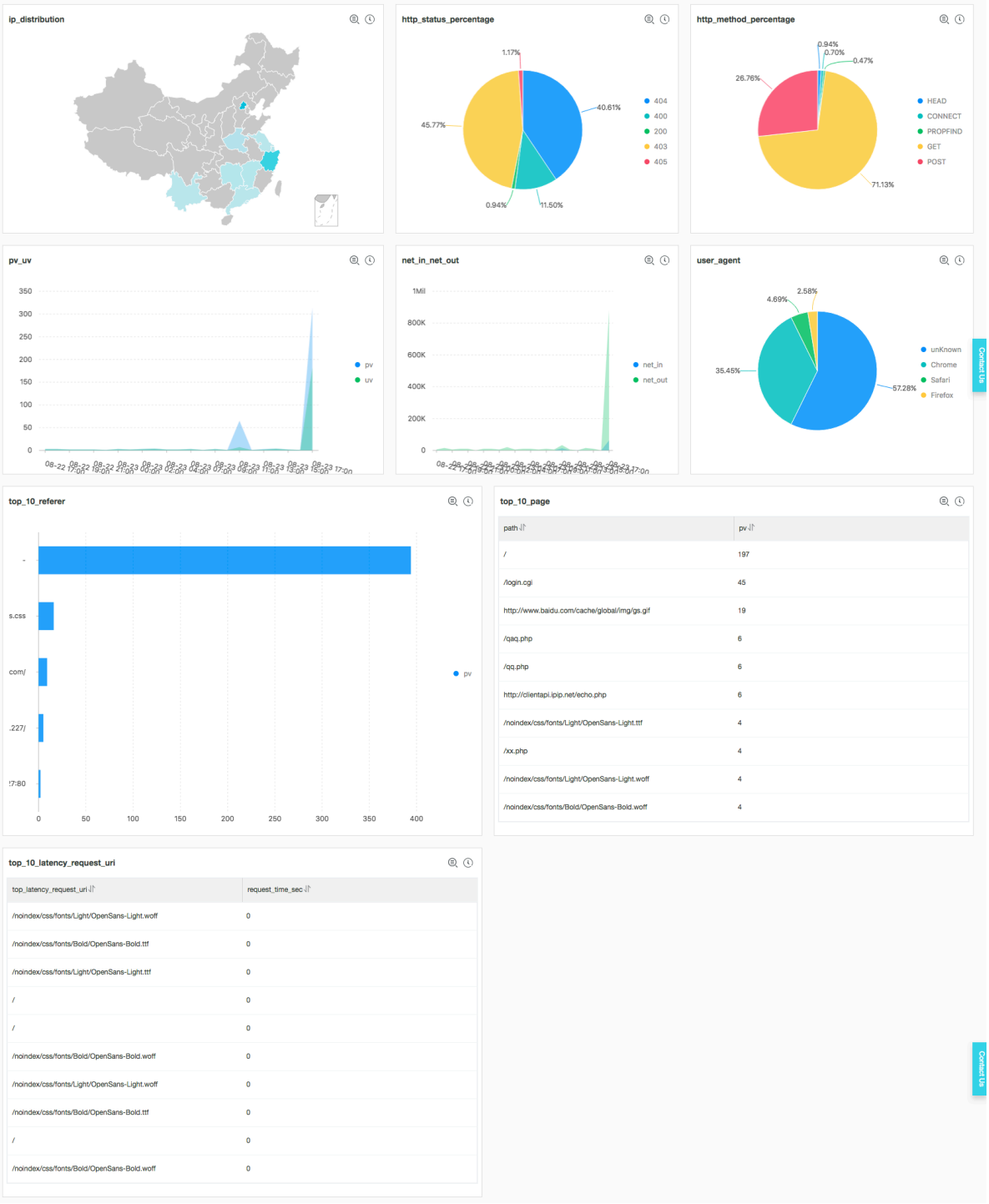
ログのマシングループのハートビートが正常ステータスであれば、プリビューボタンをクリックして収集データを表示します。




**Log Service** の収集するログデータをリアルタイムに照会/分析する必要がある場合は、インデックス属性の設定を確認します。展開をクリックして、キー値のインデックス属性を表示します。



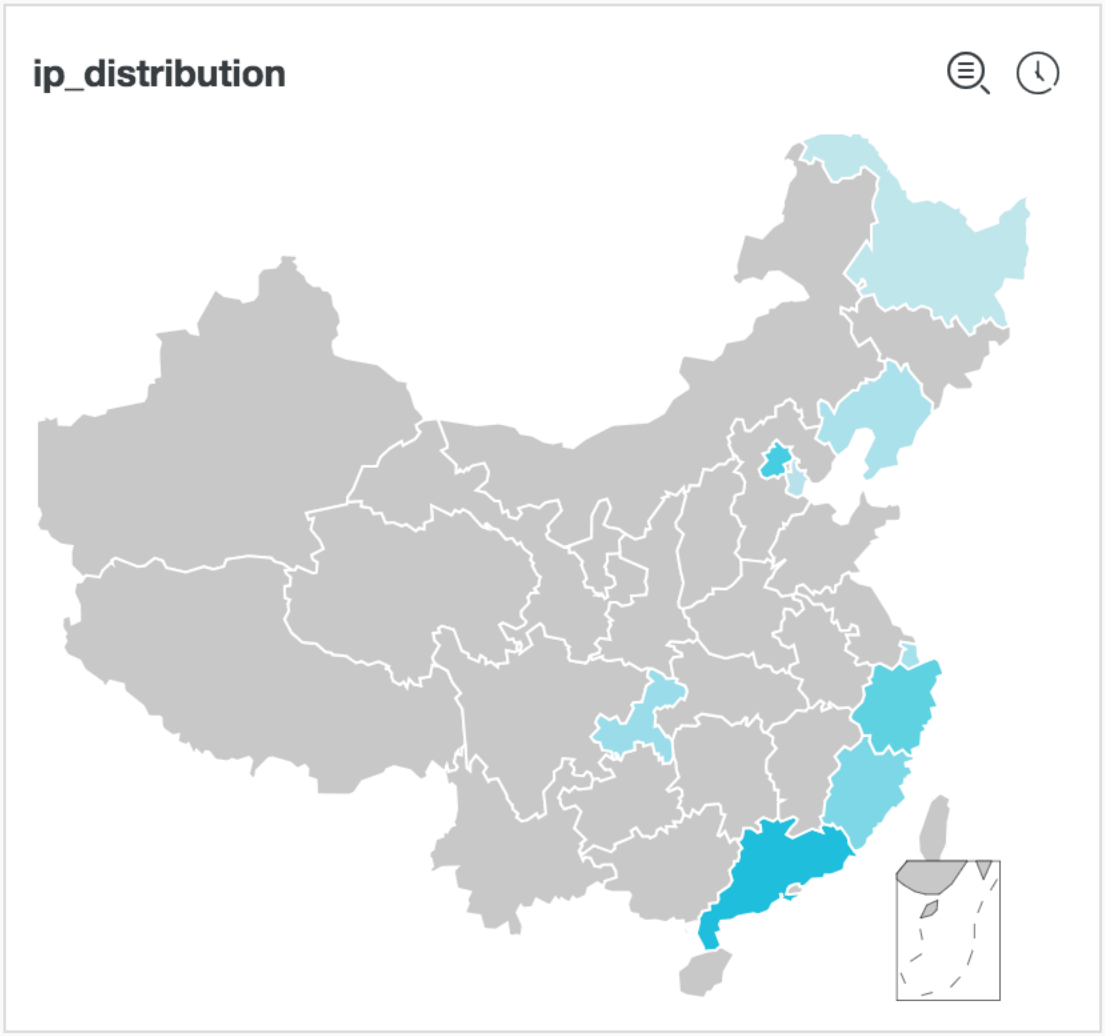
デフォルトで*LogstoreName-apache-dashboard* ダッシュボードが設定されています。設定が完了すると、ダッシュボードページで送信元 **IP** アドレスのリージョン分布、リクエストステータス比率が動的にリアルタイム表示されます。




- 送信元 **IP** アドレス分布 **(ip_distribution):** 送信元 **IP** アドレスのリージョン分布が表示されます。ステートメントは次のとおりです。

```
* | select ip_to_province(remote_addr) as address,
            count(1) as c
```

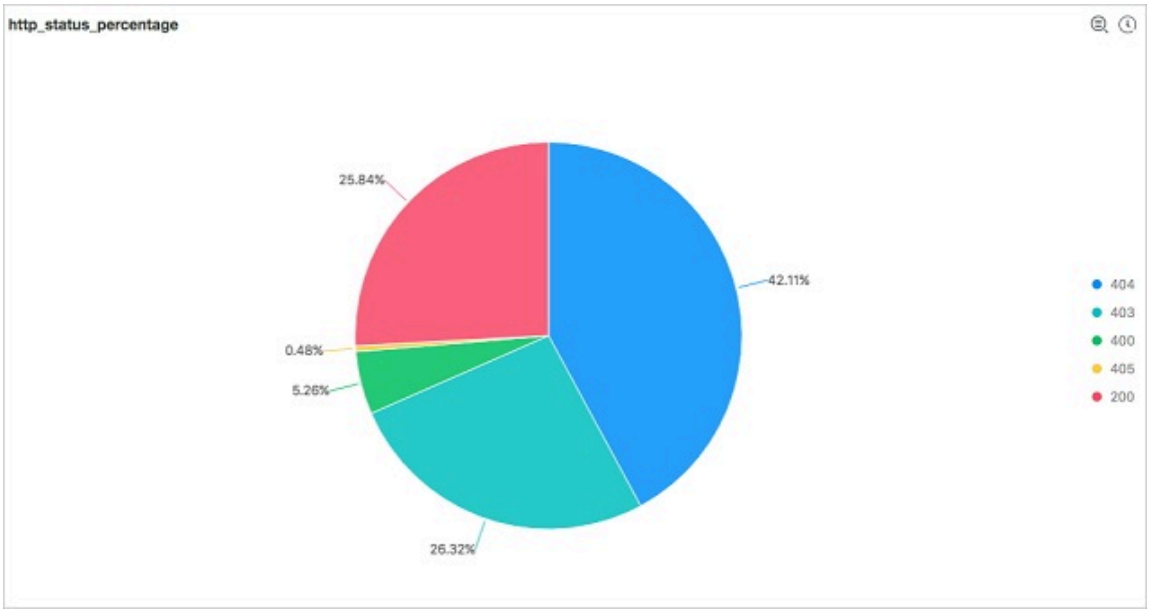
```
            group by address limit 100
```




- リクエストステータス比率 **(http_status_percentage):** 前日の各 **HTTP** ステータスコードの比率が算出されます。ステートメントは次のとおりです。

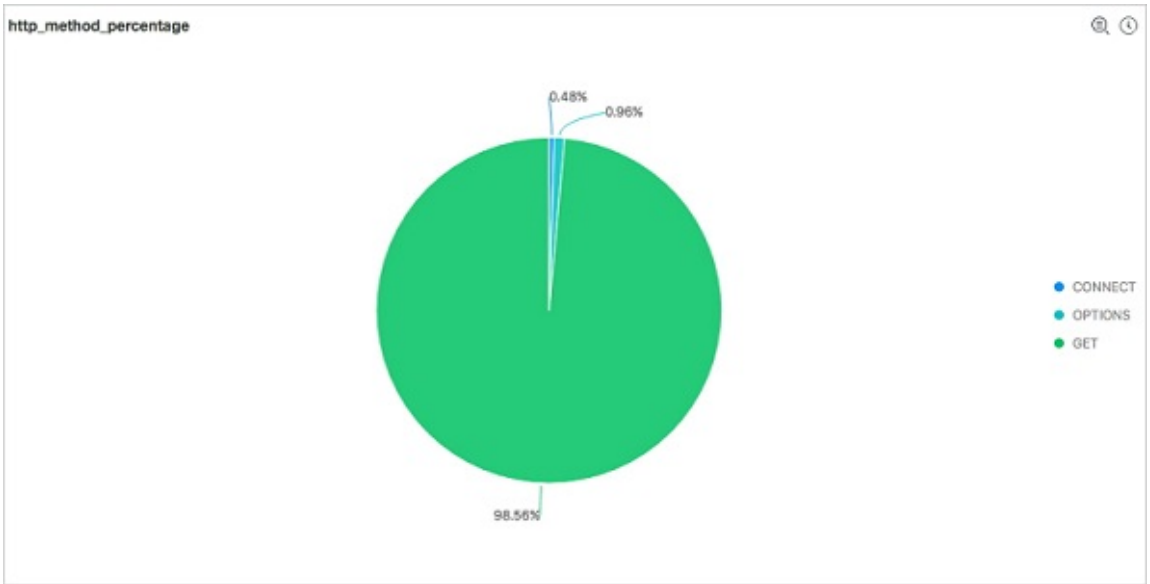
```
status>0 | select status,
                count(1) as pv
```

```
                group by status
```




- リクエストメソッド比率 **(http_method_percentage):** 前日のリクエストメソッドの比率が算出されます。ステートメントは次のとおりです。

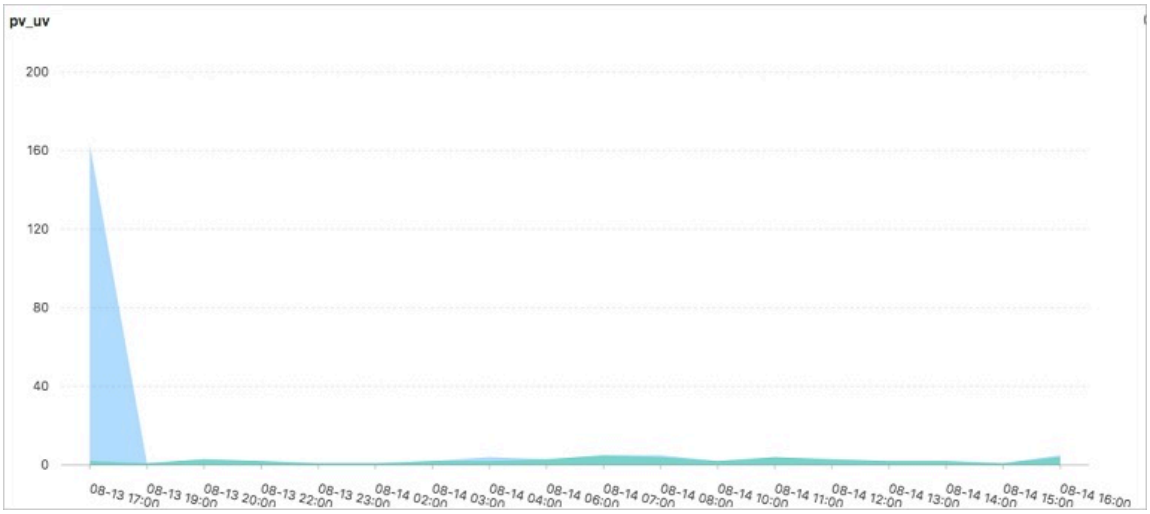
```
* | select request_method,
            count(1) as pv
            group by request_method
```




- **PV / UV** 統計 **(pv_uv):** 前日の **PV** 数と **UV** 数が算出されます。ステートメントは次のとおりです。

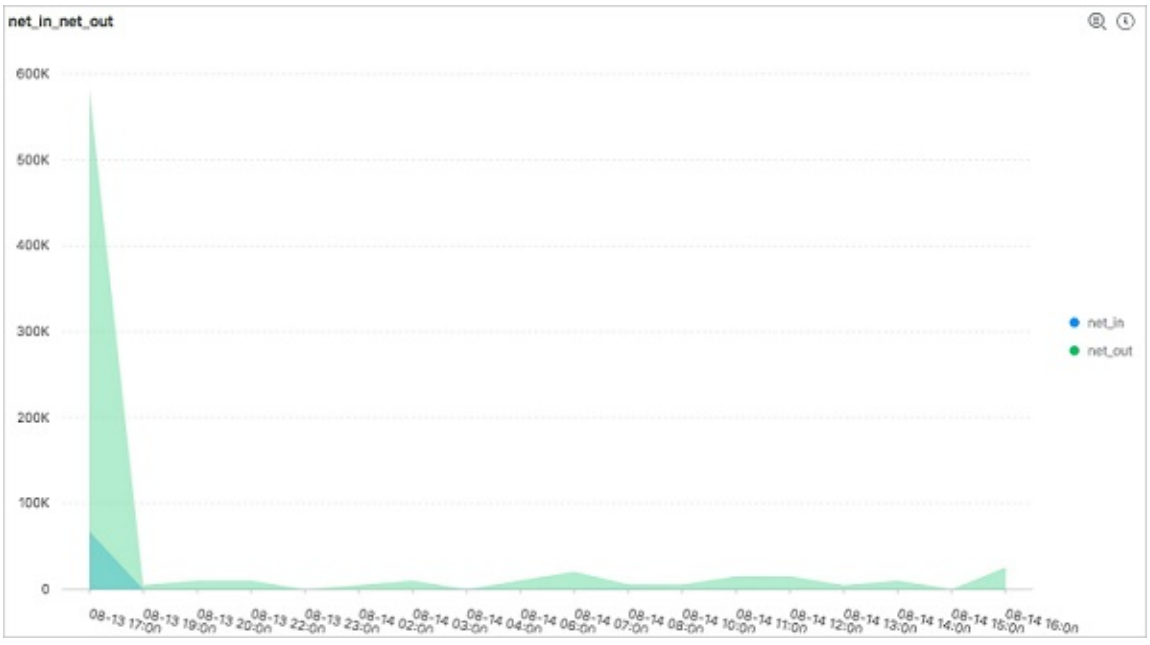
```
* | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i
')  as time,
            count(1) as pv,
            approx_distinct(remote_addr) as uv
            group by time
            order by time
```

```
            limit 1000
```




- 受信トラフィック/送信トラフィック **(net_in_net_out):** 送受信トラフィックが算出されます。ステートメントは次のとおりです。

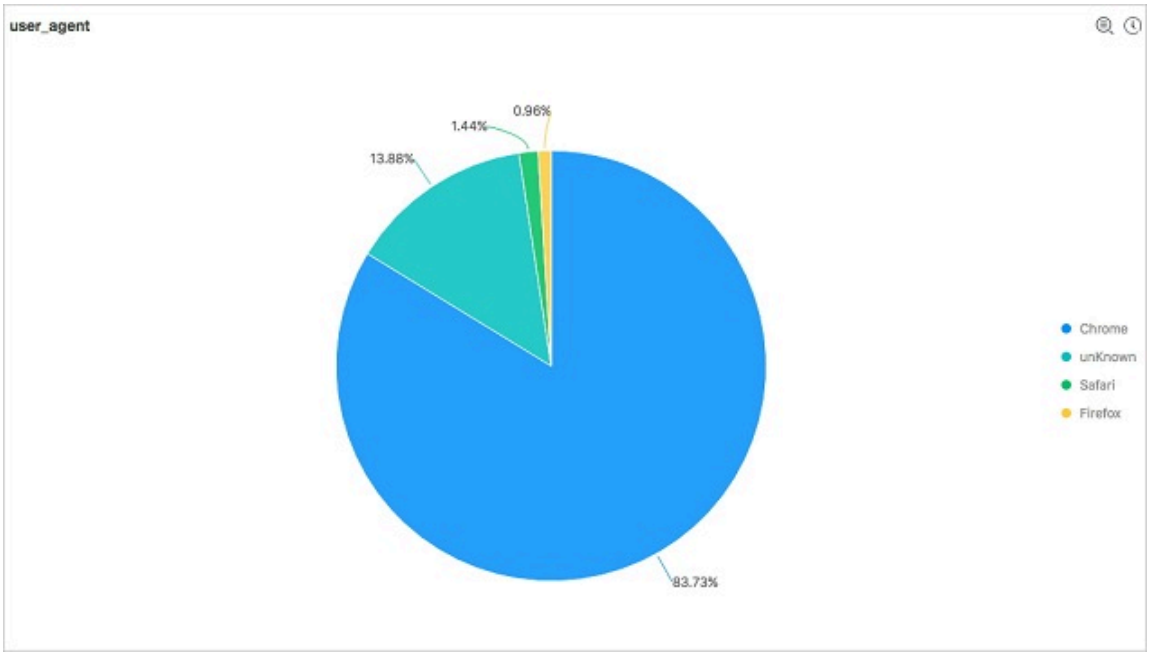
```
* | select date_format(date_trunc('hour', __time__), '%m-%d %H:%i
') as time,
            sum(bytes_sent) as net_out,
            sum(bytes_received) as net_in
            group by time
            order by time
            limit 10000mit 10
```




- リクエスト **UA** の割合 **(http_user_agent_percentage):** 前日のクライアント使用ブラウザーの割合が算出されます。 ステートメントは次のとおりです。

```
* | select  case when http_user_agent like '%Chrome%' then 'Chrome
'
            when http_user_agent like '%Firefox%' then 'Firefox'
            when http_user_agent like '%Safari%' then 'Safari'
```

```
            else 'unKnown' end as http_user_agent,count(1) as pv
            group by  http_user_agent
            order by pv desc
            limit 10
```




- リファラートップ **10 (top_10_referer):** 前日の 上位 **10** リファラーが算出されます。 ステートメントは次のとおりです。

```
* | select  http_referer,
            count(1) as pv
            group by http_referer
```

```
            order by pv desc limit 10
```


- アクセス数トップ **10** のページ（**top_page**）**:** 前日の **PV** の多かった上位 **10** ページが算出されます。 ステートメントは次のとおりです。

```
* | select split_part(request_uri,'?',1) as path,
            count(1) as pv
            group by path
```

```
            order by pv desc limit 10
```

top_10_page

| path | pv |
|---|---|
| / | 55 |
| /noindex/css/fonts/Bold/OpenSans-Bold.woff | 18 |
| /noindex/css/fonts/Light/OpenSans-Light.woff | 18 |
| /noindex/css/fonts/Light/OpenSans-Light.ttf | 18 |
| /noindex/css/fonts/Bold/OpenSans-Bold.ttf | 18 |
| /noindex/css/open-sans.css | 13 |
| /images/apache_pb.gif | 13 |
| /noindex/css/bootstrap.min.css | 13 |
| /images/poweredby.png | 13 |
| /favicon.ico | 12 |

- リクエストの応答待ち時間の多い **URI** トップ **10 (top_10_latency_request_uri):** 前日のリクエストの応答待ち時間が長かった上位 **10 URI** が算出されます。 ステートメントは次のとおりです。

```
* | select request_uri as top_latency_request_uri,
            request_time_sec
```

```
order by request_time_sec desc limit 10 10
```

**top_10_latency_request_uri**

| top_latency_request_uri | request_time_sec |
|---|---|
| /noindex/css/fonts/Light/OpenSans-Light.woff | 0 |
| /noindex/css/fonts/Bold/OpenSans-Bold.ttf | 0 |
| /noindex/css/fonts/Light/OpenSans-Light.ttf | 0 |
| / | 0 |
| / | 0 |
| /noindex/css/fonts/Bold/OpenSans-Bold.woff | 0 |
| /noindex/css/fonts/Light/OpenSans-Light.woff | 0 |
| /noindex/css/fonts/Bold/OpenSans-Bold.ttf | 0 |
| / | 0 |
| /noindex/css/fonts/Bold/OpenSans-Bold.woff | 0 |

# 7 IIS アクセスログの分析

**IIS** は、**Web** サイトを構築、運用するための拡張可能な **Web** サーバーです。 **IIS** の収集したアクセスログから、ページビュー、ユニークビジター、クライアント **IP** アドレス、不正リクエスト、送受信トラフィックといった情報を取得し、**Web** サイトへのアクセスをモニタリングおよび分析できます。

- **Log Service** を有効化します。
- プロジェクトと **Logstore** を作成します。プロジェクトと **Logstore** の作成方法については、「準備」をご参照ください。

ログフォーマット

ご要件に応じた構成にできるよう、ログフォーマットには **W3C** 拡張フォーマットを推奨します。 **IIS Manager** で、フィールドを選択トグルをクリックします。 標準フィールドリストから**sc-bytes**と**cs-bytes**を選択します。

**図** 7-1 : **フィールドの選択**

次のような設定になります。

```
logExtFileFlags="Date, Time, ClientIP, UserName, SiteName, ComputerNa
me, ServerIP, Method, UriStem, UriQuery, HttpStatus, Win32Status
, BytesSent, BytesRecv, TimeTaken, ServerPort, UserAgent, Cookie,
Referer, ProtocolVersion, Host, HttpSubStatus"
```

- フィールドの接頭辞

| 接頭辞 | 説明 |
| --- | --- |
| **s-** | サーバー操作 |
| **c-** | クライアント操作 |
| **cs-** | クライアントのサーバー操作 |
| **sc-** | サーバーのクライアント操作 |

- フィールド説明

| フィールド | 説明 |
| --- | --- |
| **date** | 操作日付 |
| **time** | 操作時刻 |
| **s-sitename** | クライアントの訪問したサイトのインターネットサービス名とインスタンス番号 |

| フィールド | 説明 |
| --- | --- |
| **s-computername** | ログを生成するサーバーの名前 |
| **s-ip** | ログを生成するサーバーの **IP** アドレス |
| **cs-method** | **GET** と **POST** といった **HTTP** リクエスト方式 |
| **cs-uri-stem** | 操作対象 |
| **cs-uri-query** | **URI (HTTP** リクエストステートメントの疑問符 **(?)** 以降**)** |
| **s-port** | クライアントの接続したサーバーのポート番号 |
| **cs-username** | サーバーにアクセスした認証済みユーザーの名前 **(**認証済みユーザーは`ドメイン名\ユーザー名`、匿名ユーザーはハイフン **(-)** を表記**)** |
| **c-ip** | リクエストの送信クライアントの **IP** アドレス |
| **cs-version** | **HTTP 1.0** や **HTTP 1.1** といったプロトコルのバージョン |
| **user-agent** | クライアントの使用ブラウザー |
| **Cookie** | 送信/受信された **Cookie** の内容 **(Cookie** がない場合は、ハイフン **(-)** を記述**)** |
| **referer** | 参照元サイト **(**ユーザーが最後に訪問したサイト**)** |
| **cs-host** | ホストのヘッダー名 |
| **sc-status** | **HTTP** や **FTP**のステータスコード |
| **sc-substatus** | **HTTP** サブプロトコルのステータスコード |
| **sc-win32-status** | **Windows** のステータスコード |
| **sc-bytes** | サーバーの送信 **Byte** |
| **cs-bytes** | サーバーの受信 **Byte** |
| **time-taken** | 操作の所要時間 **(**ミリ秒単位**)** |

1. データインポートウィザードを開始します。

   a) **Log Service** コンソールのホームページでプロジェクト名をクリックし、**Logsore** リストのページに移動します。

   b) プロジェクトのデータインポートウィザード列のアイコンをクリックします。

2. 手順 **1** のデータソース選択には、サードパーティー製のソフトウェアの**IIS ACCESS LOG**を選択します。

3. データソースを設定します。

   a) 構成名とログパスを入力します。

   ログパスは **IIS Manager** で確認できます。

4. ログフォーマットを選択します。

   **IIS** アクセスログフォーマットを選択します。

   - `IIS`: **Microsoft IIS** ログファイルフォーマット
   - `NCSA`: **NCSA** 共通ログファイルフォーマット
   - `W3C`: **W3C** 拡張ログファイルフォーマット

5. **IIS** ログフォーマット設定フィールドに入力します。

   - **Microsoft IIS** および **NCSA** 共通フォーマットの場合は、固定構成があります。
   - **IIS**アクセスログを**W3C** フォーマットに設定する手順は、次のとおりです。

   **a)** **IIS** 設定ファイルを開きます。

   - **IIS5** 設定ファイルのデフォルトパス: *C:\WINNT\system32\inetsrv\MetaBase.bin*
   - **IIS6** 設定ファイルのデフォルトパス: *C:\WINDOWS\system32\inetsrv\MetaBase.xml*
   - **IIS7** 設定ファイルのデフォルトパス: *C:\Windows\System32\inetsrv\config\applicationHost.config*

   **図** 7-2 : **設定ファイルの表示**

   **b)** 図**3**のように、`logFile logExtFileFlags`フィールドの引用符で囲まれたテキストをコピーします。

   **c)** コピーしたテキストをコンソールの**IIS** ログフォーマット設定フィールドに貼り付けます。

   **図** 7-3 : **データソースの設定**

6. キー名を確認します。

   **IIS** ログサービスにより、キー名が自動的に抽出されます。

   **図** 7-4 : IIS **キー名**

7. 詳細オプション (オプション)

| パラメーター | 説明 |
| --- | --- |
| 生ログのアップロード | 生ログをアップロードするかどうかを指定します。 このスイッチをオンにすると、未処理のログコンテンツが **_raw__** フィールドとしてアップロードされ、解析されたログコンテンツが表示されます。 |
| トピック生成モード | ・ **Null** - トピックを生成しない：トピックが **null** 文字列に設定されるように指定するデフォルト値。 トピックを入力せずにログを照会できます。<br>・ マシン グループ トピック属性：異なるフロントエンド サーバーで生成されたログデータと区別するために、マシン グループに基づいてトピックを設定します。<br>・ ファイル パス正規表現：カスタム正規表現を使用して、ログパスの一部をトピックとして抽出します。 ユーザーとインスタンスによって生成されたログデータを区別するために使用されるモードです。 |
| カスタム正規表現 | トピック生成モードを **[ファイルパスの正規表現]** に設定する場合、カスタム正規表現を入力する必要があります。 |
| ログファイルエンコーディング | ・ **utf8**：**UTF-8** エンコーディングを指定します。<br>・ **gbk**：**GBK** エンコーディングを指定します。 |
| 監視ディレクトリの最大深度 | ログソースからログを収集するときの監視ディレクトリの最大深度つまり、管理対象のディレクトリのレベルの最大数 有効値： **[0, 1000] 0** は、現在のディレクトリのみが監視されていることを示します。 |
| **Timeout** | 指定した期間内にファイルの更新がないときに、ログファイルがタイムアウトしたとシステムが考慮するかどうかを指定します。 **Timeout** は以下のように設定できます。<br>・ タイムアウトにならない：すべてのログファイルはタイムアウトなしで監視が継続されるように指定します。<br>・ **30** 分タイムアウト：ログファイルが **30** 分以内に更新されない場合、ログファイルがタイムアウトしたと見なし、ファイルの監視を停止するように指定します。 |

| パラメーター | 説明 |
| --- | --- |
| フィルター設定 | 収集前にログが 完全に満たす 必要のあるフィルター条件<br>例：<br>・ 条件を満たすログの収集： `Key:level Regex:WARNING\|ERROR` は、レベルが「**WARNING**」または「**ERROR**」のログのみを収集することを表します。<br>・ 条件に適合しないログをフィルターします：<br>- `Key:level Regex:^(?!.*(INFO\|DEBUG)).*` は、レベルが「**INFO**」または「**DEBUG**」のログが収集されていないことを表します。<br>- **Set a condition** `Key:url Regex:.*^(?!.*(healthcheck)).*` は、**url** に **heartcheck** のあるログは収集されないことを表します。 たとえば、**key** が **url**、**value** が `/inner/healthcheck/jiankong.html` のログは収集されません。<br>その他の例については、「*regex-exclude-word*」 と 「*regex-exclude-pattern*」をご参照ください。 |

設定情報を確認して、次へをクリックします。

8. マシングループに設定を適用します。

   設定を適用するマシングループを選択します。ページの右下のマシングループに適用をクリックします。

   マシングループをまだ作成していなければ、マシングループの作成をクリックして作成します。

9. 検索、分析、可視化を設定します (オプション)。

   マシングループの「ハートビート」が正常ステータスであれば、プレビューをクリックしてログデータを表示できます。

   **図** 7-5 : **ログのプレビュー**

   作業中のページのインデックス属性を確認して、収集したログデータを表示および分析します。 開くをクリックしてキー/値のインデックス属性を表示します。

   キー名のマッピングを構成します。 キー名は、プレビューされたデータに基づいて生成され、デフォルトのキー名に対応しています。

   **図** 7-6 : **キー/値インデックス属性**

   デフォルトで*LogstoreName-iis-dashboard*ダッシュボードが用意されています。 上記を設定したら、ダッシュボードにリアルタイムデータを表示できます (クライアント **IP** 分布、**HTTP** ステータスの割合など)。

   **図** 7-7 : **ダッシュボード**

   - クライアント **IP** 分布を取得するステートメントは、次のとおりです。

     ```
     | select ip_to_geo("c-ip") as country, count(1) as c group by
     ip_to_geo("c-ip") limit 100
     ```

     **図** 7-8 : **クライアント** IP **分布**

   - ページビューおよびユニークビジターを確認するステートメントは、次のとおりです。

     ```
     *| select approx_distinct("c-ip") as uv ,count(1) as pv ,
     date_format(date_trunc('hour', __time__), '%m-%d %H:%i') as time
     ```

```
group by date_format(date_trunc('hour', __time__), '%m-%d %H:%i')
order by time limit 1000
```

**図** 7-9 : **ページビューおよびユニークビジター**

- **HTTP**ステータスの割合を取得するステートメントは、次のとおりです。

```
*| select count(1) as pv ,"sc-status" group by "sc-status"
```

**図** 7-10 : HTTP **ステータスの割合**

- 送受信トラフィックを表示するステートメントは、次のとおりです。

```
*| select sum("sc-bytes") as net_out, sum("cs-bytes") as net_in ,
date_format(date_trunc('hour', time), '%m-%d %H:%i') as time group
 by date_format(date_trunc('hour', time), '%m-%d %H:%i') order by
time limit 10000
```

**図** 7-11 : **送受信トラフィック**

- **HTTP** リクエストメソッドの割合を取得するステートメントは、次のとおりです。

```
*| select count(1) as pv ,"cs-method" group by "cs-method"
```

**図** 7-12 : HTTP **リクエストメソッドの割合**

- 使用ブラウザー比率を取得するステートメントは、次のとおりです。

```
*| select count(1) as pv, case when "user-agent" like '%Chrome%'
then 'Chrome' when "user-agent" like '%Firefox%' then 'Firefox'
when "user-agent" like '%Safari%' then 'Safari' else 'unKnown' end
 as "user-agent" group by case when "user-agent" like '%Chrome%'
then 'Chrome' when "user-agent" like '%Firefox%' then 'Firefox'
```

```
when "user-agent" like '%Safari%' then 'Safari' else 'unKnown' end
 order by pv desc limit 10
```

**図** 7-13 : **使用ブラウザー比率**

- 訪問数の多い **URI** トップ **10**を表示するステートメントは、次のとおりです。

```
*| select count(1) as pv, split_part("cs-uri-stem",'?',1) as path
group by split_part("cs-uri-stem",'?',1) order by pv desc limit 10
```

**図** 7-14 : **訪問数の多い** URI **トップ** 10