Alibaba Cloud Log Service

データ収集

Document Version20200706

目次

1 ログ収集方法

LogHub は、クライアント、Web サイト、プロトコル、SDK、API を介するなど、複数の方法をサポートしています。 すべての収集方法は、Restful API に基づいて実装されます。 API と SDK を使用して、新しい収集方法を実装することもできます。

データ ソース

Log Service は次のソースからログを収集できます。

カテゴリ	ソース	収集方法	参照
アプリケーション	プログラム出力	Logtail	
	アクセスログ	Logtail	#unique_3
	リンクトラッキング	Jaeger Collector および Logtail	-
言語	Java	SDK および Java Producer Library	-
	Log4j appender	1.x および 2.x	-
	Logback appender	Logback	-
	C	ネイティブ	-
	Python	Python	-
	Python ロギング	Python ロギングハンドラ	-
	PHP	PHP	-
	C#	C#	-
	C++	#unique_6	-
	Go	Go および Go Producer Library	-
	Node.js	Node.js	-
	JavaScript	JavaScript/Web tracking	-
オペレーティング システム (OS)	Linux	Logtail	-
	Windows	Logtail	-
	Mac OS または Unix	Native C	-

カテゴリ	ソース	収集方法	参照
	ドッカーファイル	Logtail を使用してのドッカーファイルの収集	-
	ドッカー出力	Logtail を使用してのコンテナーログの収集	-
モバイル クライアント	iOS/アンドロイド	#unique_11, #unique_12	-
	Web ページ	JavaScript/Web トラッキング	-
	インテリジェント IoT	C プロデューサー ライブラリ	
Alibaba Cloud サービス	ECS、OSS、およびその他の Alibaba Cloud サービス。 詳細については「Flume の使用」をご参照ください。	Alibaba Cloud コンソール上で Log Service を有効化	クラウドサービスログ
	MaxCompute import	DataWorks を使用しての MaxCompute データのエクスポート	DataWorks を使用して MaxCompute データを Log Service に収集
サードパーティ製ソフトウェア	Logstash	Logstash	-
	Flume	#unique_16	-

次の表に、Log Service がログを収集できる Alibaba Cloud サービスを示します。

タイプ	クラウドサービス	有効化する方法	備考
エラスティックコンピューティング	ECS	Logtail をインストールします。	Logtail の紹介
	Container Service / Container Service for Kubernetes	Container Service または Container Service for Kubernetes コンソールにて有効化	テキストログ と 出力
ストレージ	OSS	OSS コンソールにて有効化	#unique_17
ネットワーク	SLB	SLB コンソールにて有効化	レイヤ 7 SLB のアクセスログ

タイプ	クラウドサービス	有効化する方法	備考
	VPC	VPC コンソールにて有効化	#unique_19
	API Gateway	API Gateway コンソールにて有効化	API Gateway のアクセスログ
セキュリティ	ActionTrail	ActionTrail コンソールにて有効化	概要
	Anti-DDoS Pro /BGP-line Anti-DDoS Pro	Anti-DDoS Pro コンソールにて有効化	Anti-DDoS Pro 概要 と BGP-line Anti-DDoS Pro 概要
アプリケーション	Log service	Log Service コンソールにて有効化	#unique_24

ネットワークの選択

Log Service は、さまざまな Alibaba Cloud リージョンのサービスエンドポイントを提供します。 詳細については「Flume の使用」をご参照ください。 各リージョンでは、次のネットワークからのアクセスが許可されます。

- 内部ネットワーク (クラシックネットワーク)またはプライベートネットワーク(VPC)：同リージョン内の別の Alibaba Cloud サービスへのアクセスを可能にし、最適なリンク帯域幅が提供されます。 このオプションの選択を推奨します。
- パブリック ネットワーク (クラシック ネットワーク)：無制限にアクセスできます。 伝送速度はリンクの品質によって異なります。 HTTPS を使用して、安全なデータ転送の確保を推奨します。

よくある質問

- Q：プライベート回線アクセスにはどのネットワークを選択すればよいですか？

  A：内部ネットワークまたはプライベート ネットワークを選択します。
- Q：パブリック ネットワーク データを収集するときにパブリック IP アドレスを収集できますか?

  A：Log Service がパブリック IP アドレスを記録できるようにする必要があります。 詳細については「Flume の使用」をご参照ください。

- Q：リージョン A から ECS ログを収集し、これらのログをリージョン B の Log Service プロジェクトに書き込む場合は、どのネットワークを選択しますか。

  A：パブリック ネットワークを選択します。 インターネット伝送用のリージョン A の ECS インスタンスに Logtail をインストールし、リージョン B に関連付けられているサービスエンドポイントを指定できます。ネットワークの選択方法の詳細については、「ネットワークタイプの選択」をご参照ください。
- Q：サービスエンドポイントにアクセスできるかどうかを判断するにはどうすればよいですか。

  A：次のコマンドを実行できます。 情報が返された場合、サービスエンドポイントにアクセスできます。

  ```
  curl $myproject.cn-hangzhou.log.aliyuncs.com
  ```

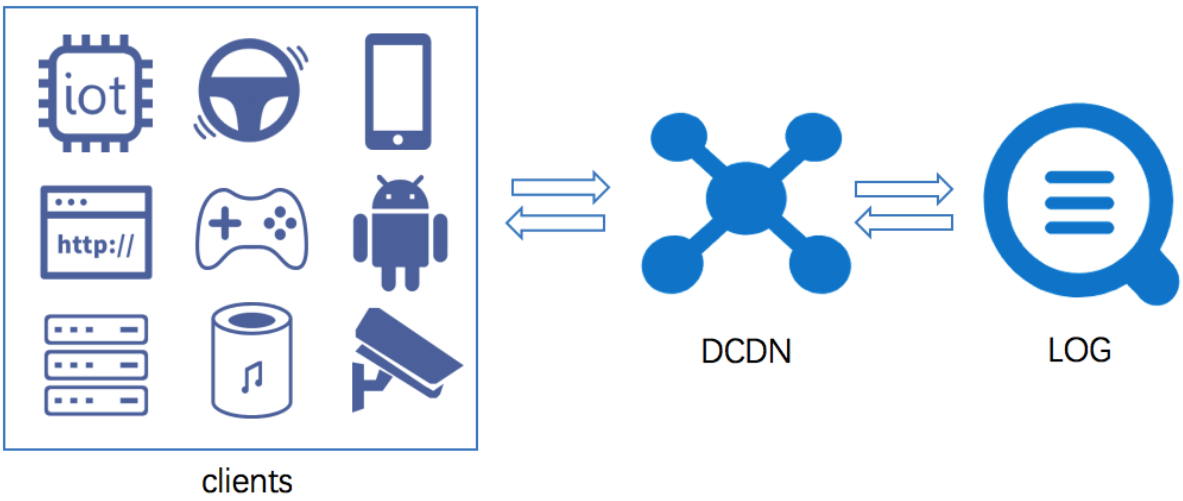
  $myproject はプロジェクト名を指定し、cn-hangzhou.log.aliuncs.com はサービスエンドポイントを指定します。

2 収集の加速

2.1 概要

Log Serviceは、Virtual Private Cloud（VPC）およびパブリックネットワークに基づいて、**Global Acceleration** パブリックネットワークのネットワークタイプを追加します。 Global Acceleration パブリックネットワークは、通常のパブリックネットワークアクセスと比較して、遅延と安定性の面で大きな利点があり、データ収集、消費遅延、信頼性の高いシナリオに適しています。 Log Service 向けのGlobal Acceleration は、Alibaba Cloud Dynamic Routeが提供するCDNプロダクトのアクセラレーション環境に依存します。 この機能により、キャリア間のアクセス、不安定なネットワーク、トラフィックの急増、ネットワークの輻輳といった要因により発生する、遅い応答、パケットの消失、不安定なサービスなどの問題を解決して、サイト全体のパフォーマンスとユーザーエクスペリエンスを向上します。

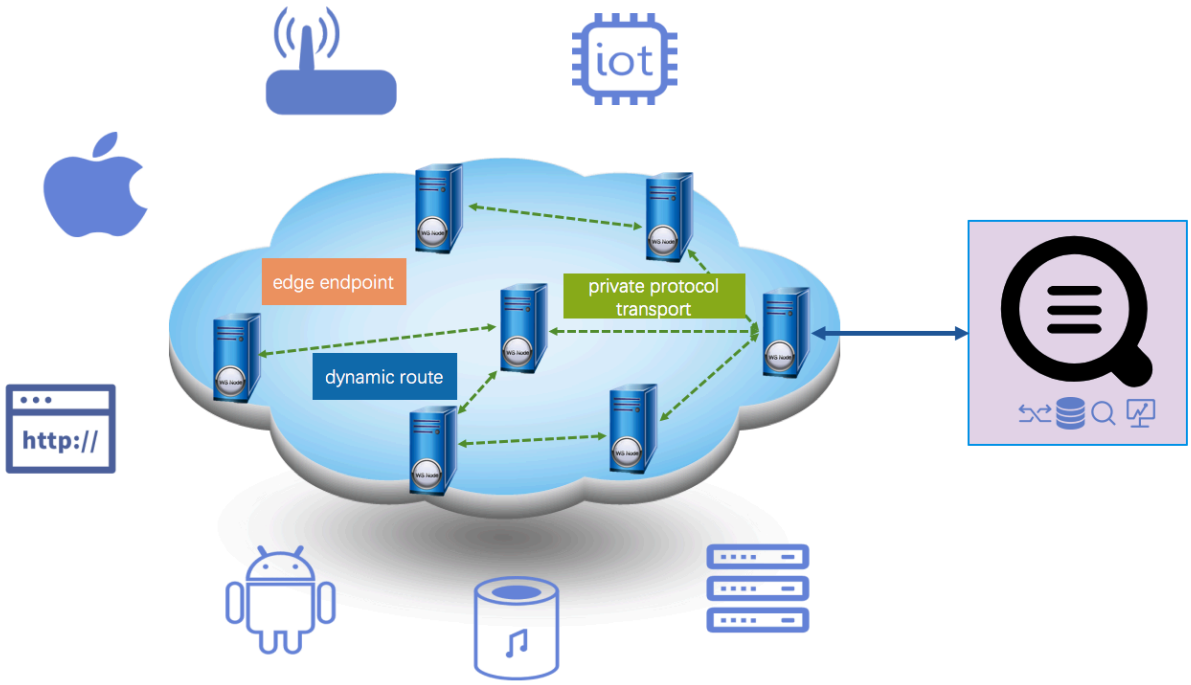
Log Service 向けのGlobal Acceleration は、Alibaba Cloud Content Delivery Network（CDN）ハードウェアリソースに基づいており、携帯電話、IoT（Internet of Things）デバイス、スマートデバイスなどのさまざまなデータソースからのログ収集およびデータ転送の安定性を最適化します 、セルフビルドインターネットデータセンター（IDC）、およびその他のクラウドサーバーが含まれます。



技術原理

Log Service 向けのGlobal Acceleration は、Alibaba Cloud CDNハードウェアリソースに基づいています。 お使いのグローバルアクセス端末（携帯電話、IOTデバイス、スマートデバイス、セルフビルドIDC、その他のクラウドサーバーなど）は、世界中のAlibaba Cloud CDNの最寄ノー

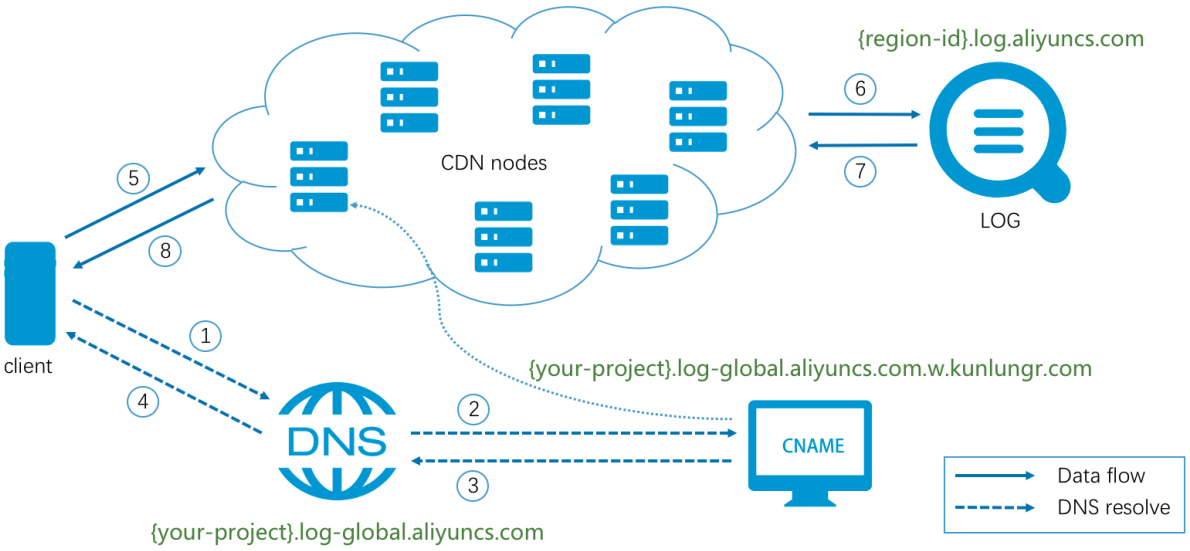
ドにアクセスし、 CDN内部高速チャネルを通じてLog Serviceにルーティングします。 この方法は、一般的なパブリックネットワーク転送と比較して、ネットワークの遅延とジッターを大幅に削減できます。



Log Service 向けのGlobal Acceleration リクエストの処理フローは上図の通りです。 全体的な流れは次のようになります：

1. クライアントは、ログアップロードまたはログダウンロードリクエストをログサービスアクセラレーションドメイン名your-project.log-global.aliyuncs.comに送信する前に、パブリックDNSにドメイン名解決リクエストを送信する必要があります。
2. パブリックDNSのドメイン名your-project.log-global.aliyuncs.comは、CNAMEアドレスyour-project.log-global.aliyuncs.com.w.kunlungr.comを指します。 ドメイン名解決は、Alibaba Cloud CDNのCNAMEノードに転送されます。
3. Alibaba Cloud CDNスマートスケジューリングシステムに基づき、CNAMEノードは最適なCDNエッジノードのIPアドレスをパブリックDNSに返します。
4. パブリックDNSは、最終的にクライアントに解決されたIPアドレスを返します。
5. クライアントは、取得したIPアドレスに基づいてサーバにリクエストを送信します。
6. リクエストを受信すると、CDNエッジノードは、ダイナミックルート検索およびプライベートトランスポートプロトコルに基づいて、Log Serviceサーバに最も近いノードにリクエストをルーティングします。 リクエストは最終的にLog Serviceに転送されます。
7. Log Serviceのサーバは、CDNノードからリクエストを受信すると、そのリクエストの結果をCDNノードに返します。

8. CDNは、Log Serviceによって返された結果またはデータを透過的にクライアントに送信します。



課金方法

Log Service 向けGlobal Acceleration の課金項目は次のとおりです：

- Log Serviceへのアクセスによって発生する費用

  Log Serviceへのアクセスによって発生する費用は、一般的なパブリックネットワークの費用と同様です。 Log Serviceは、従量課金をサポートし、無料クォータも提供しています。 詳細は、#unique_30を参照してください。
- CDN用ダイナミックルートサービスの費用

  CDN用ダイナミックルートのクラウドプロダクト費用については、CDNのダイナミックルートの請求方法を参照してください。

シナリオ

- 広告

  広告閲覧回数とクリック回数に関するログデータは、広告の料金請求にとって非常に重要です。 広告キャリアは、携帯端末、H5ページ、PC端末などがあります。 一部の辺鄙な地方では、パブリックネットワークのデータ転送の安定性が低く、ログ損失のリスクが存在します。 Global Acceleration によって、より安定した信頼性の高いログアップロードチャネルを取得することができます。

- オンラインゲーム

  オンラインゲーム業界では、公式サイト、ログインサービス、セールスサービス、ゲームサービス、およびその他のサービスにおけるデータ収集のパフォーマンスと安定性に対する高い要件があります。 モバイルゲームデータ収集やグローバル化されたゲームのデータバック送信の場合、データ収集の適時性や安定性を保証するのは難しくなります。 この問題を解決するには、Log Service 向けの Global Acceleration を使用することを推奨します。

- 金融

  金融関係のアプリケーションでは、ネットワークの高可用性と高いセキュリティが求められます。 各トランザクションおよび各ユーザーアクションの監査ログは、安全かつ確実にサーバーに収集する必要があります。 現在、モバイルトランザクションが主流になっています。 たとえば、オンラインバンキング、クレジットカードモール、モバイル証券、およびその他のタイプのトランザクションは、Log Service 向け HTTPS Global Acceleration を使用して、安全で高速で安定したログ収集を実現します。

- IoT

  IoTデバイスやスマートデバイス（スマートスピーカーやスマートウォッチなど）は、データ分析のためにセンサーデータ、操作ログ、重要なシステムログ、およびその他のデータをサーバーに収集します。 これらのデバイスは、通常、世界中に分散されており、周囲のネットワークは必ずしも信頼性の高いものではありません。 安定した信頼性の高いログ収集を実現するには、Log Service 向け Global Acceleration を使用することを推奨します。

アクセラレーションの効果

リージョン	遅延**ms**（一般的パブリックネットワーク）	遅延**ms**（アクセラレーション）	タイムアウト割合%（一般的パブリックネットワーク）	タイムアウト割合%（アクセラレーション）
杭州	152.881	128.501	0.0	0.0
ヨーロッパ	1750.738	614.227	0.5908	0.0
アメリカ	736.614	458.340	0.0010	0.0
シンガポール	567.287	277.897	0.0024	0.0
中東	2849.070	444.523	1.0168	0.0
オーストラリア	1491.864	538.403	0.014	0.0

テスト環境は次の通り：

- Log Serviceリージョン：中国（フフホト）
- アップロードするパケットの平均サイズ：10KB
- テスト時間範囲：1日（平均値）
- リクエストタイプ：HTTPS
- リクエストサーバー：Alibaba Cloud ECS（仕様：1C1GB）

> 注：
> アクセラレーション効果は参考用です。

2.2 Global Acceleration の有効化

Global Accleration を有効化するには、次の手順に従ってください。

前提条件

- Log Serviceを有効にして、プロジェクトと Logstore が作成されていること。
- Dynamic Route for CDN が有効になっていること。
- HTTPS acceleration を有効にするには、 まず HTTP acceleration を有効にする必要があります。

設定

プロジェクトのHTTP Global Acceleration が有効になった後、必要に応じてLogtail、SDKなどの Global Acceleration を構成することもできます。

1. HTTP アクセラレーションの有効化。

2. Logtail、SDKなどの Global Acceleration を有効にする。

   - HTTPS

     HTTPSを使用してLog Serviceにアクセスする場合は、HTTPSアクセラレーションを確実に有効にしてください。 HTPSアクセラレーションを設定するには、HTTPS アクセラレーションの有効化を参照してください。

   - Logtailログ収集

     Logtailをインストールする際に、ページプロンプトでグローバルアクセラレーションネットワークタイプを選択します。 これにより、Logtailを使用してログを収集すると、Global Acceleration を取得できます。

   - SDK/Producer/Consumer

     エンドポイントをlog- global.aliyuncs.comに置き換えることで、SDK、Producer、ConsumerなどのLog Serviceにアクセスする他の方法をアクセラレーションすることができます。

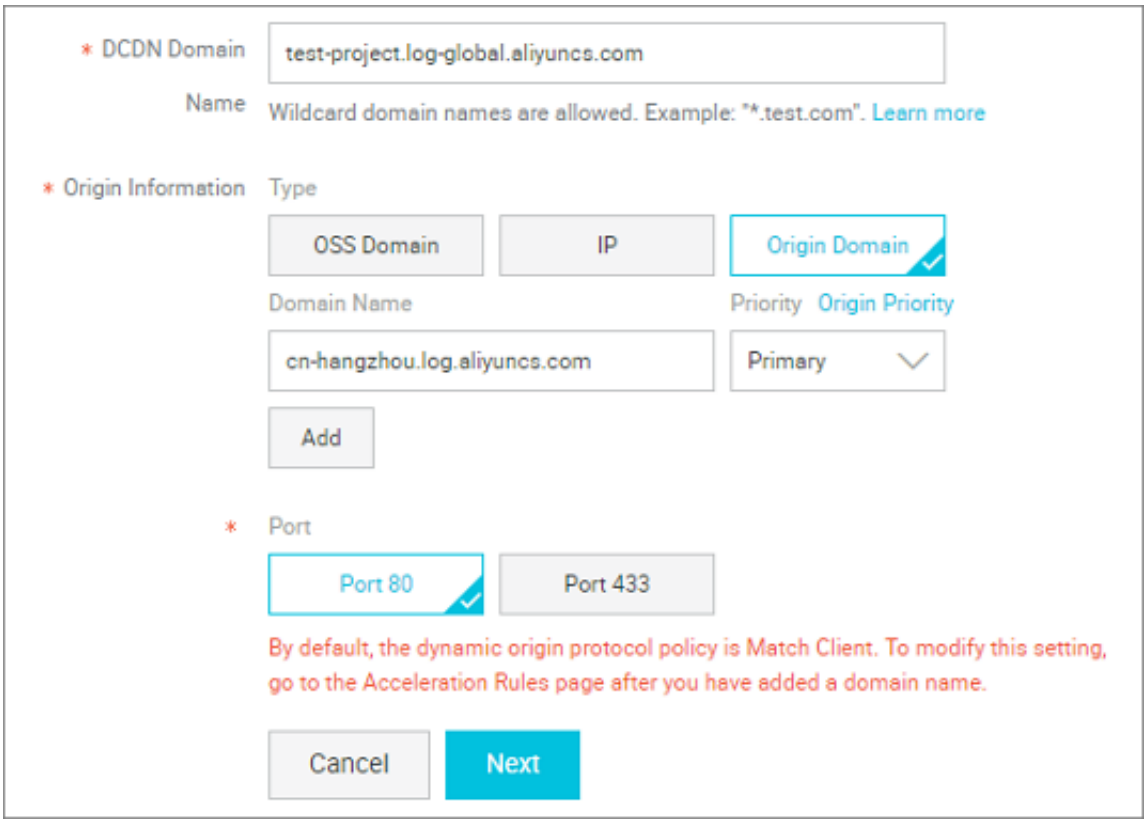
HTTPアクセラレーションの有効化

1. Dynamic Route for CDNコンソールにログインします。 左側のメニューでドメイン名をクイックしてドメイン名ページへ移動します。
2. 左上のドメイン名の追加ボタンをクイックしてドメイン名の追加ページへ移動します。
3. **DCDN**ドメイン名などの情報を入力して、次をクリックします。

構成項目	説明
ドメイン名のアクセラレーション	project_name.log-global.aliyuncs.com の project_name をご使用するプロジェクト名に置き換えます。
オリジン情報のタイプ	オリジンドメインを選択します。
ドメイン名	プロジェクトが属するリージョンのパブリックネットワークエンドポイントを入力します。 エンドポイントに関する情報は、サービスエンドポイントを参照してください。
ポート	ポート80を選択します。 HTTPSアクセラレーションで要件がある場合、**HTTPS**アクセラレーションの有効化を参照してください。

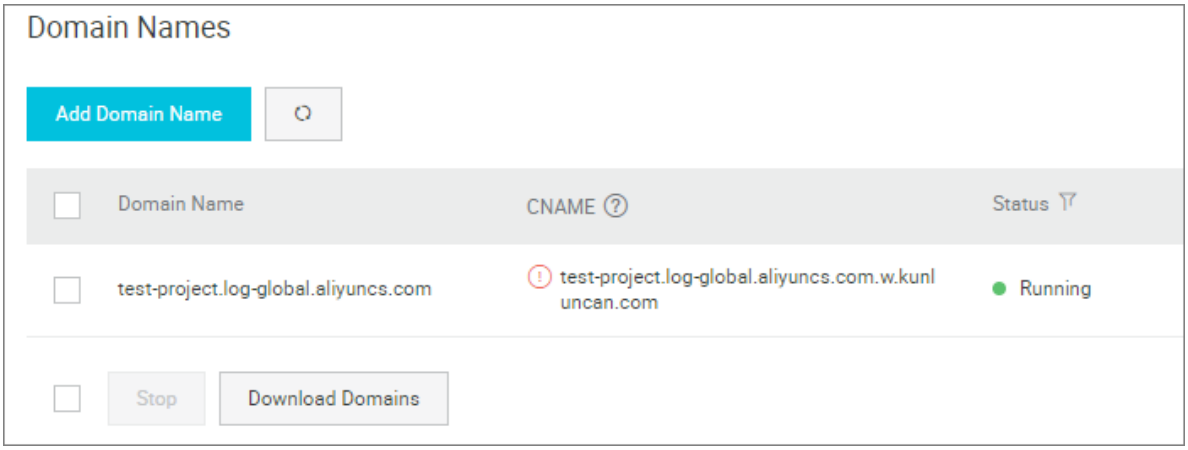
構成項目	説明
アクセラレーションリージョン	この構成項目はデフォルトで表示されず、アクセラレーションリージョンは中国本土となります。 Global Acceleration が必要な場合は、ホワイトリストに申請するため、Dynamic Route for CDNへチケットを起票してサポートセンターへお問い合わせください。

ドメイン名の追加に関する詳細情報は、「ドメイン名の追加」に参照してください。

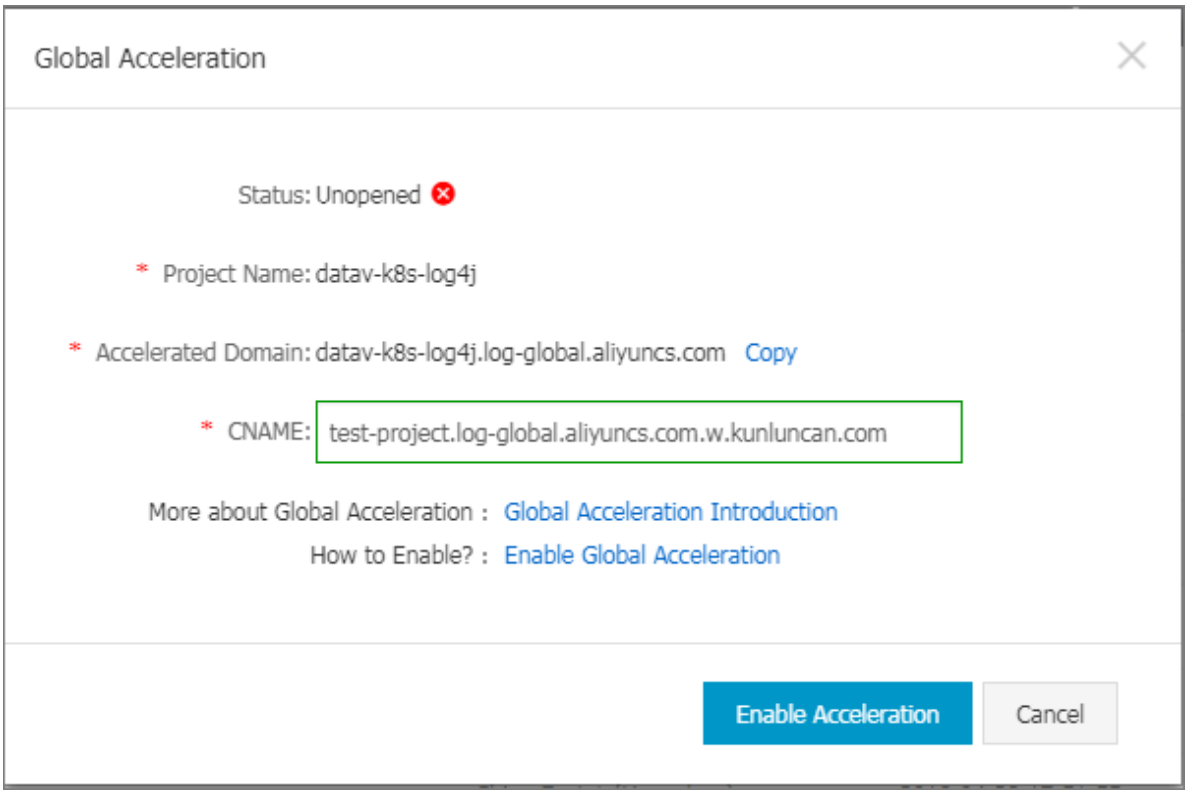


4. ページの指示に従ってドメイン名ページへ移動します。

ドメイン名ページで、対応する各ドメイン名の**CNAME**を表示できます。



5. Log Serviceコンソールにログインして、プロジェクトリスト内の特定プロジェクトの右側にある**Global Acceleration**をクリックします。
6. ダイアログボックスで加速されたドメイン名に対応する**CNAME**を入力します。 アクセラレーションを有効化をクリックします。



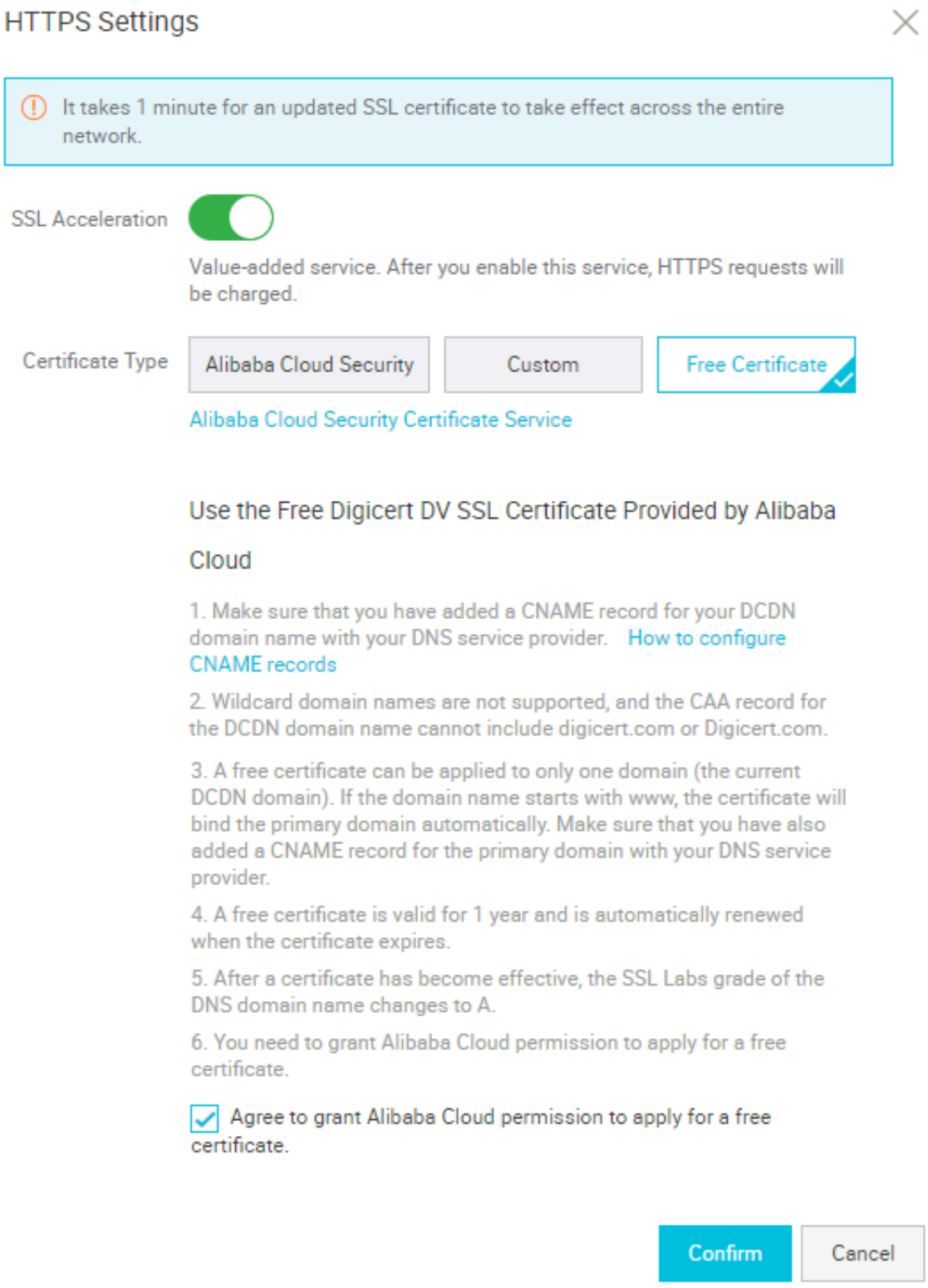
上記の手順を実行すると、Log Service 向けGlobal Acceleration が有効になります。

HTTPSアクセラレーションの有効化

HTTPアクセラレーションを有効にした後、HTTPSアクセスの要件がある場合、次の手順に従ってHTTPSアクセラレーションを有効にすることができます。

1. Dynamic Route for CDNコンソールにログインします。 左側のメニューで、ドメイン名をクリックしてドメイン名ページへ移動します。
2. 特定のドメイン名の右側にある設定をクリックします。
3. 左側のメニューで、**HTTPS**設定をクリックし、**SSL**証明書セッション内の変更をクイックして**HTTPS**設定ページを開きます。

4. **SSL**アクセラレーションと証明書タイプを設定します。
   - **SSL**アクセラレーションを有効にします。
   - 証明書タイプで無料証明書を選択します。



設定が終わったら、**Alibaba Cloud**が無料証明書を申請することに同意する。にチェックを入れ、**OK**をクリックします。

アクセラレーション設定が有効になっているか確認する

よくある質問

- アクセラレーション設定が有効かどうかを確認する方法は？

  設定が終わったら、アクセラレートされたドメイン名にアクセスすることでアクセラレーションが有効になっているかどうかを確認できます。

  たとえば、Global Acceleration がtest-projectプロジェクトに対して有効になっている場合、curlを使用してアクセラレートされたドメイン名にリクエストを送信します。 次のタイプのアウトプットが返されると、アクセラレーションが有効だと判断できます。

  ```
  $curl test-project.log-global.aliyuncs.com
  {"Error":{"Code":"OLSInvalidMethod","Message":"The script name is invalid : /","
  RequestId":"5B55386A2CE41D1F4FBCF7E7"}}
  ```

  チェック方法に関する詳細情報は、アクセラレーション設定が有効かどうかを確認する方法を参照してください。
- アクセラレートされたドメイン名にアクセスする際に返されるproject not existというエラーをどのように処理しますか？

  この問題は、通常、無効な送信元サイトアドレスによって発生します。 Dynamic Route for CDNコンソールにログインし、ソースサイトのアドレスを、プロジェクトが属するリージョンのパブリックネットワークアドレスに変更します。 アドレスリストに関する詳細情報は、サービスエンドポイントを参照してください。

  > 注：
  > ソースサイトアドレスを変更すると、数分間の同期化遅延が発生します。

2.3 Logtail 収集アクセラレーションの設定

Global Acceleration が有効になった後、Global Acceleration モードでインストールされた Logtail は、Gloabl Accelerationモードでログを自動的に収集します。 Global Acceleration が有効になる前にインストールされた Logtail の場合は、このドキュメントを参照し、アクセラレーションモードをに手動で切り替える必要があります。

前提条件

1. HTTPアクセラレーションの有効化.
2. (オプション) HTTPアクセラレーションの有効化.

   HTTPS を使用して Log Service にアクセスする場合は、HTTPS アクセラレーションが有効になっていることや、HTTPアクセラレーションの有効化の指示に従って HTTPS アクセラレーションが構成されていることを確認します。
3. アクセラレーションが正常に機能することを確認します

   グローバルアクセラレーションの有効化に記載されている手順に従います。

始める前に

Logtail 収集アクセラレーションを構成する前に、次の点に注意します：

- グローバルアクセラレーションを有効にしてから Logtail をインストールする場合は、Logtail の Linux へのインストールの指示に従い、インストールモードを グローバルアクセラレーション に設定する必要があります。 その後、Logtail はグローバルアクセラレーションモードの方法を使用してログを収集します。
- グローバルアクセラレーションが有効になる前に Logtail がインストールされている場合は、本ドキュメントを参照し、Logtail 収集モードをグローバルアクセラレーションに切り替える必要があります。

Logtail 収集モードをグローバルアクセラレーションへの切り替え

1. Logtail を停止します。
   - Linux の場合、管理者アカウントで `/etc/init.d/ilogtaild stop` を実行します。
   - Windows の場合：
     a. コントロールパネルで、システムとセキュリティ > 管理ツールを選択します。
     b. サービスプログラムを開き、LogtailWorker ファイルを検索します。
     c. ファイルを右クリックし、ショートカットメニューで停止をクリックします。
2. Logtail 起動構成ファイル ilogtail_config.jsonを変更します。

   起動構成ファイル (ilogtail_config.json)の指示に従ってエンドポイントの **data_server_list** を `log-global.aliyuncs.com` に変更します。

3. Logtail を起動します。
   - Linuxの場合、管理者アカウントで /etc/init.d/ilogtaild start を実行します。
   - Windows の場合：
     a. コントロールパネルで、システムとセキュリティ > 管理ツールを選択します。
     b. サービスプログラムを開き、LogtailWorker ファイルを検索します。
     c. ファイルを右クリックし、ショートカットメニューで開始クリックします。

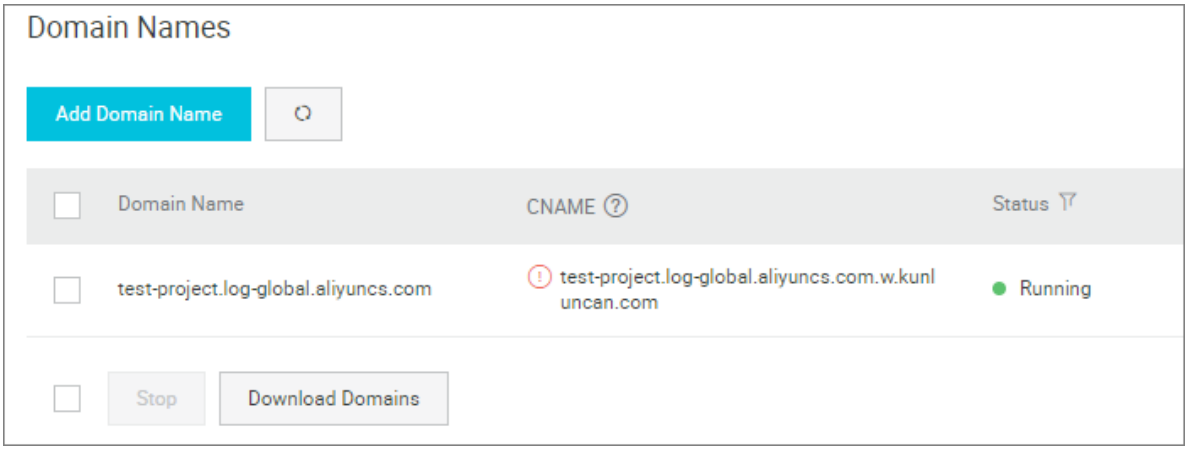
2.4 Global Accelerationを無効化する

Log ServiceのGlobal Acceleration を無効化するには、次の操作を実行します。

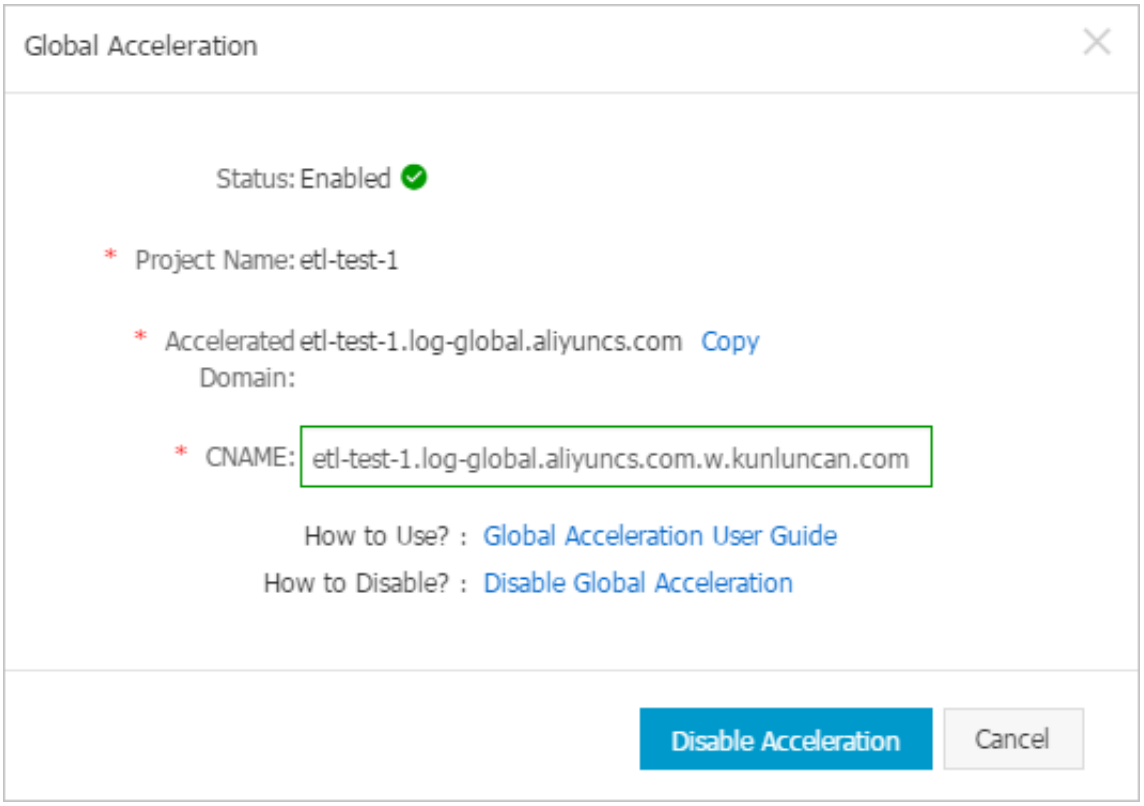
> 注：
> Global Acceleration を無効にすると、プロビジョニング中に設定したドメイン名が利用できなくなります。 Global Acceleration を無効にする前に、すべてのクライアントがドメイン名を使用してデータのアップロードやリクエストを行っていないことを確認してください。

Global Acceleration を無効化

1. DCDNコンソールにログインします。 左側のドメイン名をクリックしてドメイン名ページを開きます。
2. 無効化しようとするドメイン名に対応する**CNAME**を確認します。



3. Log Serviceコンソールにログインします。 プロジェクトリストページで、対象プロジェクトの右側にある**Global Acceleration**をクリックします。
4. **CNAME**を入力して加速を無効化をクリックします。


Global Acceleration
Status: Enabled
* Project Name: etl-test-1
* Accelerated Domain: etl-test-1.log-global.aliyuncs.com Copy
* CNAME: etl-test-1.log-global.aliyuncs.com.w.kunluncan.com
How to Use? : Global Acceleration User Guide
How to Disable? : Disable Global Acceleration
Disable Acceleration
Cancel

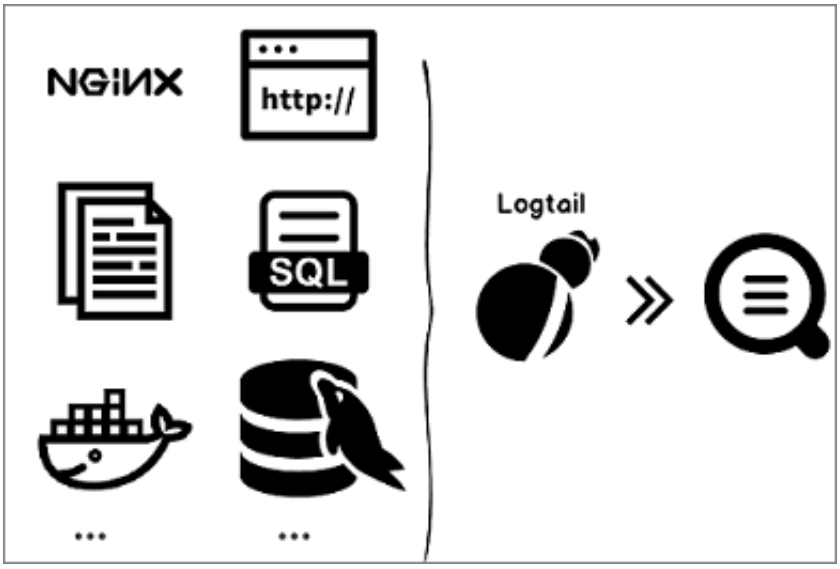
3 Logtailでの収集

3.1 概要

3.1.1 概要

Logtail アクセスサービスは、 Log Service によって提供されるログ収集エージェントです。 Logtail を使用して、Alibaba Cloud Elastic Compute Service （ECS）インスタンスなどのサーバーから、 Log Service コンソールでリアルタイムでログを収集することができます。

図 3-1：機能の利点



利点

- ログファイルに基づく非接続のログ収集。 アプリケーションのコードを変更する必要はなく、ログ収集はアプリケーションの動作ロジックに影響を与えません。
- テキストログの収集に加えて、binlog、http、およびcontainer stdoutなど、より多くの収集方法がサポートされています。
- コンテナーは十分にサポートされています。 このサービスは、標準コンテナー、集団クラスター、およびKubernetesクラスターでのデータ収集をサポートします。
- Logtail は、ログ収集プロセスで発生した例外を処理します。 ネットワークや Log Service などで異常が発生し、ユーザデータが一時的に予約帯域の書き込み制限を超えるといった問題が発生すると、Logtail はローカルで積極的にデータを再試行してキャッシュし、データセキュリティを保証します。

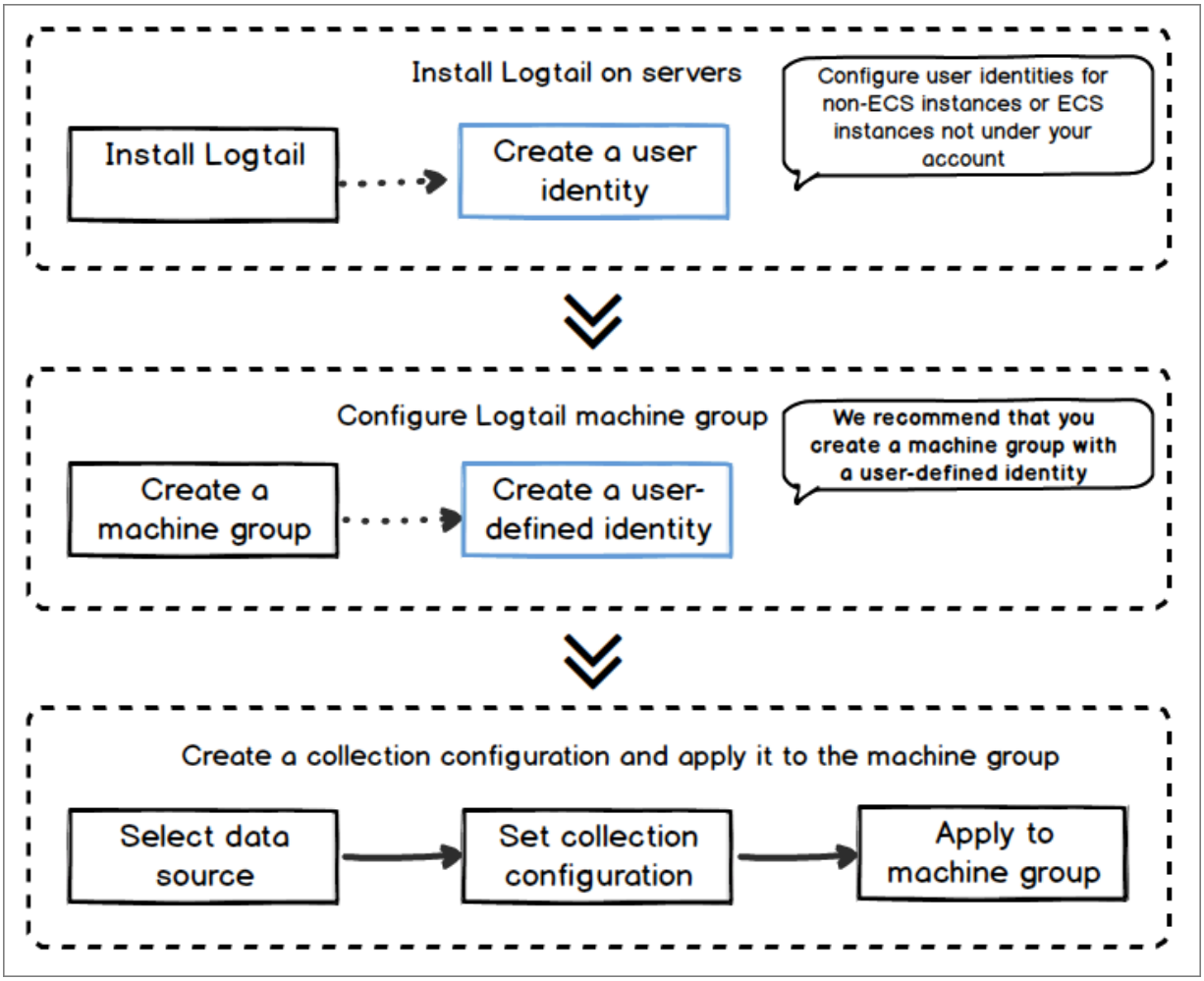
- ログサービスに基づく集中管理機能。 Logtail をインストールした後は、ログを収集するマシンや Log Service での一元的な収集方法などの設定を、サーバーにログインすることなく、また個別に設定することなく行うことができます。 Logtail のインストール方法については、Logtail の Windowsへのインストールと Logtail の Linux へのインストールを参照してください。
- 包括的な自己保護の仕組み。 マシン上で実行されている収集エージェントがサービスのパフォーマンスに大きな影響を与えないように、Logtail クライアントは CPU、メモリ、ネットワークリソースの使用を厳しく保護し、制限します。

処理能力と制限

制限を参照してください。

手順

図 3-2 : 設定プロセス



Logtail を使用してサーバからログを収集するには、次の手順に従います。

1. Logtail をインストールします。 ログを収集するサーバーに Logtail をインストールします。 詳細：Logtail の Windowsへのインストール と Logtail の Linux へのインストール
2. マシングループにユーザー定義 ID を設定する。 Alibaba Cloud ECS インスタンスからログを収集しようとしている場合は、この手順をスキップしてください
3. Logtail マシングループの作成。 Log Service は、マシングループの形式で Logtail クライアントを使用してログを収集するすべてのサーバーを管理します。 Log Service は、IP またはユーザー定義の ID を使用してマシングループを定義することをサポートします。 マシングループに適用ページの指示どおりにマシングループを作成することもできます。
4. Logtail コレクション設定を作成し、それをマシングループに適用します。 データインポートウィザードで Logtail 設定を作成することにより、テキストファイルの収集や#unique_45などのデータを収集することができます。 次に、Logtail 設定をマシングループに適用することができます。

上記の手順を完了すると、ログを収集するサーバーの増分ログがアクティブに収集され、対応するログストアに送信されます。 履歴ログは収集されません。 コンソールまたは API/SDK を使用してこれらのログを照会することができます。 収集が正常かどうか、エラーが発生したかどうかなど、コンソールの Logtail ログ収集状況を照会することもできます。

Log Service コンソールの Logtail アクセスサービスの完全な手順については、Logtail を使用したテキストファイルの収集を参照してください。

コンテナー

- Alibaba クラウド Container Service ：#unique_46を参照してください。
- Alibaba クラウドService Kubernetesクラスター：Kubernetes のログ収集
- 自己構築型Kubernetes：自分で構築した Kubernetes をインストールする
- その他自己構築Dockerクラスタ：標準の Docker ログの収集

主なコンセプト

- グループ：マシングループには、ログタイプを収集する 1 つ以上のマシンが含まれています。 Logtail 構成をマシングループに適用すると、 Log Service は同じ Logtail 構成に従ってマシングループ内のすべてのマシンからログを収集します。 Log Service コンソールでは、マシングループの作成/削除、マシングループへのマシンの追加 / 削除など、マシングループを管理することもできます。 マシングループは、Windows マシンと Linux マシンの両方を含むことはできませんが、異なるバージョンの Windows マシンまたは異なるバージョンの Linux マシンを含むことができます。

- **Logtail**クライアント： Logtail は、ログを収集し、ログが収集されるサーバー上で実行されるエージェントです。 Logtail のインストール方法については、Logtail の Windowsへのインストールと Logtail の Linux へのインストールを参照してください。 サーバに Logtail をインストールした後、Logtail 設定を作成し、それをマシングループに適用します。
  - **Linux**では、Logtail は/usr/local/ilogtailディレクトリに インストールされ、ilogtail で始まる 2 つの独立したプロセス（収集プロセスとデーモンプロセス）を開始します。 プログラム実行ログは /usr/local/ilogtail/ilogtail.LOGです。
  - **Windows**では、Logtail は C:\Program Files\Alibaba\Logtailディレクトリ（32 ビットシステム用） またはC:\Program Files(x86)\Alibaba\Logtailディレクトリ（64 ビット用 システム）にインストールされます。 Windows 管理ツール> サービスに移動すると、 LogtailWorker と LogtailDaemon の 2 つの Windows サービスを表示できます。 LogtailDaemon はデーモンとして動作します。 プログラム実行ログは、 インストールディレクトリのlogtail_*.logです。
- **Logtail** 構成： Logtail 構成は、Logtail を使用してログを収集するポリシーの集合です。 データソースや収集モードなどの Logtail パラメーターを設定することで、マシングループ内のすべてのマシンのログ収集ポリシーをカスタマイズできます。 Logtail 設定は、マシンからログの種類を収集し、収集したログを解析し、指定された Log Service の Logstore にそれらを送信するために使用されます。 Logstore がこの Logtail 構成を使用して収集されたログを受信できるようにするには、コンソール内の各 Logstore に Logtail 構成を追加できます。

基本機能

Logtail アクセスサービスは、以下の機能を提供します。

- リアルタイムログ収集： Logtail は動的にログファイルを監視し、増分ログをリアルタイムで読み込み、解析します。 一般に、ログが生成されてから Log Service ログが送信されるまでの間に、3 秒未満の遅延があります。

  注：

  Logtail は履歴データの収集をサポートしていません。 ログが読み込まれてからログが生成されるまでの間隔が 5 分を超えるログは破棄されます。
- 自動ログローテーション処理：多くのアプリケーションは、ログファイルをファイルサイズまたは日付に従ってローテーションします。 ローテーション処理中に、元のログファイルの名前が変更され、ログの書き込み用に新しい空のログファイルが作成されます。 例えば、監視されたapp.LOGはapp.LOG. 1とapp.LOG. 2を生成するためにローテーションします。 app.LOG のように、収集されたログが書き込まれるファイルを指定することができます。 Logtail は自

動的にログローテーションプロセスを検出し、このプロセス中にログデータが失われないことを保証します。

- 複数の収集入力ソース：テキストログ以外にも、Loglog は syslog、HTTP、MySQL、binlog などの入力ソースをサポートしています。 詳細については、「ログサービスユーザーガイド」の「データソース」を参照してください。
- オープンソース収集エージェントとの互換性：Logstashの入力ソースは、LogstashやBeatsなどのオープンソースソフトウェアによって収集されたデータにすることができます。 詳細については、「ログサービスユーザーガイド」の「データソース」を参照してください。
- 収集例外の自動処理： Log Service エラー、ネットワーク対策、制限を超えるクォータなどの例外によりデータ送信が失敗した場合、Logtail は特定のシナリオに基づいて積極的に再試行します。 再試行が失敗した場合、Logtail はローカルキャッシュにデータを書き込み、その後、自動的にデータを再送信します。
- 柔軟な収集ポリシー設定： Logtail 設定を使用して、サーバからログを収集する方法を柔軟に指定することができます。 具体的には、実際のシナリオに基づいて、ワイルドカードと完全一致またはファジー一致をサポートするログディレクトリとファイルを選択できます。 ログ収集の抽出方法と抽出されたフィールドの名前をカスタマイズすることができます。 Log Service、正規表現を使用してログを抽出できます。 Log Service のログデータモデルでは、各ログに正確なタイムスタンプが必要です。 したがって、Logtail はカスタムログの時刻形式を提供し、さまざまな形式のログデータから必要なタイムスタンプ情報を抽出することができます。
- 収集構成の自動同期：一般的に、 Log Service コンソールで構成を作成または更新すると、Logtail は自動的にその構成を受け入れ、3 分以内に構成を有効にします。 構成の更新時に収集されたデータは失われません。
- クライアントの自動アップグレード： Logtail をサーバーに手動でインストールすると、 Log Service は自動的に Operation＆Maintenance（O＆M） およびLogtail のアップグレードを実行します。 Logtail をアップグレードしてもログデータは失われません。
- ステータス監視： Logtail クライアントがリソースを消費し過ぎてサービスに影響を与えないようにするため、Logtail クライアントは CPU とメモリの消費量をリアルタイムで監視します。Logtail クライアントは、リソースの使用量が制限を超えた場合に自動的に再起動され、マシン上の他の操作に影響を与えないようにします。 Logtail クライアントは、ネットワークトラフィックを積極的に制限して、過剰な帯域幅消費を防ぎます。

- シグネチャによるデータ送信：送信プロセス中にデータが改ざんされるのを防ぐため、Logtail クライアントは Alibaba Cloud AccessKey を取得し、送信されるすべてのログデータパケットに署名を提供します。

   注：

  Alibaba Cloud AccessKey のセキュリティを維持するため、Logtail は HTTPS トンネルを使用してAccessKeyを取得します。

3.1.2 Logtail の収集プロセス

Logtail クライアントがサーバーからログを収集する処理には、ファイルのモニタリング、ファイルの読み取り、ログの処理、ファイルのフィルタリング、ログの集約、およびログの転送の 6 つのステップがあります。

サーバーに Logtail クライアントをインストールして Logtail 構成を設定すると、Logtail は Log Service へのログの収集を開始します。 ログ収集は次の処理を行います。

1. ファイルのモニタリング
2. ファイルの読取り
3. ログの処理
4. ログのフィルタリング
5. ログの集約
6. ログの送信

 注：

- 詳細は、 Alibaba Cloud Community をご参照ください。
- マシングループに Logtail 構成を設定すると、そのマシングループ内のサーバーのログに変更のなかったものは、過去のファイルとみなされます。過去のファイルは収集されません。過去のログを収集するには、過去ログのインポートをご参照ください。

ファイルのモニタリング

サーバーに Logtail クライアントをインストールし、データソースに合わせて Logtail 構成を設定すると、その Logtail 構成は Logtail にすぐに送信されます。Logtail 構成に基づいてファイルのモニタリングが開始します。

1. 具体的には、設定されたログパスと最大モニタリングディレクトリの階層に合わせて、ファイル命名規則に準拠した指定ログディレクトリとファイルが階層ごとにスキャンされます。

   ログ収集の効率と安定性を確保するために、Logtail は収集ディレクトリ (つまり、Linux の inotify ディレクトリまたは Windows の ReadDirectoryChangesW ディレクトリ) のイベントモニタリングを登録し、定期的にポーリングします。
2. モニタリング結果に、ファイル命名規則に準拠した指定されたディレクトリに変更のなかったログファイルがあった場合、そのファイルは収集されません。変更のあったログファイルがあれば、収集プロセスが起動し、そのファイルは読み取られます。

ファイルの読み取り

変更のあったファイルを読み取ります。

1. 初めて読み取るファイルの場合、ファイルサイズがチェックされます。
   - ファイルサイズが 1 MB より小さい場合、ファイルは先頭から読み取られます。
   - ファイルサイズが 1 MB より大きい場合、ファイルの最後の 1 MB が読み取られます。
2. 以前に読み込まれたことのあるファイルの場合、最後のチェックポイントからファイルが読み込まれます。
3. 一度に 512 KB のみが読み取り可能です。したがって、ログは 512 KB 以下にします。

 注：
サーバー時間を変更した場合は、手動で Logtail を再起動する必要があります。再起動しないと、ログ生成時間が正確なものではなくなり、誤ってログが削除される可能性があります。

ログの処理

ログは行に分割また解析され、時間フィールド設定が適正が確認されます。

1. 行に分割

   Logtail 構成で改行正規表現を指定すると、改行設定に従ってログは複数の行に分割されます。各行は 1 つのログとして処理されます。改行の正規表現を指定しない場合、データブロックが 1 つのログとして処理されます。
2. 解析

   Logtail 構成に設定された正規表現、区切り文字、JSON 配列に基づいてログコンテンツが解析されます。

    注：

正規表現を複雑にし過ぎると、CPU 使用率が異常に高くなる可能性があります。したがって、正規表現を効率のよいものにされることを推奨します。

3. 解析失敗処理

   Logtail 構成で解析失敗ログの破棄機能の有効/無効によって、解析失敗ログは次のように処理されます。

   - 有効化の場合、ログは破棄され、エラーが報告されます。
   - 無効化の場合、元のログと併せて **raw_log** のキーとログ内容の値をアップロードする必要があります。

4. 時間フィールド設定

   - 本フィールドを設定しない場合、現在の解析時間がログ生成時間に適用されます。
   - 本フィールドを設定した場合、ログ生成時間は、次のとおりとなります。
     - 現在時刻から 12 時間経過していない場合、解析済み時刻フィールドから時間が抽出されます。
     - 現在時刻から 12 時間以上経過した場合、ログは破棄されエラーが報告されます。

ログのフィルタリング

Logtail 構成のフィルター設定に従ってログがフィルタリングされます。

- フィルターを設定した場合、ログはフィルタリングされず、直接ログを統計します。
- フィルターを設定しない場合、各ログのすべてのフィールドがスキャンおよび検証されます。
  - フィルター設定に適合するログが収集されます。フィルター設定のすべてのフィールドがログに含まれ、すべてのフィールドが設定条件を満たしている。
  - フィルター設定に適合しないログは収集されません。

ログの集約

ログデータは Log Service に送信されます。その前にログは一旦キャッシュされます。ネットワークリクエストの数を減らすために、必要なログは集約、パッケージ化されてから Log Service に送信されます。

キャッシュされたログが次のいずれかに適合する場合、そのログはパッケージ化されてから送信されます。

- ログの集約に 3 秒を超える
- 集約するログが 4,096 を超える
- ログが 512 KB を超える

ログの送信

Log Service に集約ログが送信されます。起動パラメータの構成の手順に従って、起動パラメータの max_bytes_per_sec (ログ送信速度) およびsend_request_concurrency (同時送信できるログの最大数) を調整します。調整した値を超えて送信されることがなくなります。

ログの送信に失敗した場合、エラーメッセージのとおり、タスクは自動的に再試行または終了します。

エラーメッセージ	説明	処理方法
エラーコード：401	Logtail クライアントにはデータを収集する権限がありません。	Logtail はログパッケージを削除
エラーコード：404	Logtail Config で指定されたプロジェクトまたは Logstore が存在しません。	Logtail はログパッケージを削除
エラーコード：403	シャードクォータが上限を超えています。	3 秒後に再試行
エラーコード：500	サーバーでエラーが発生しました。	3 秒後に再試行
ネットワークタイムアウト	ネットワーク接続エラーが発生しました。	3 秒後に再試行

3.1.3 Logtail 構成とログファイル

Logtail は、いくつかの設定ファイルに基づいて実行され、特定の情報を記録したログファイルを生成します。本ドキュメントでは、一般的な生成ファイルの概要とパスについて説明します。

構成ファイル：

- 起動構成ファイル (ilogtail_config.json)
- AliUid 構成ファイル
- カスタム ID ファイル (user_defined_id)
- Logtail 構成ファイル (user_log_config.json)

記録ファイル：

- AppInfo 記録ファイル (app_info.json)
- Logtail 操作ログファイル (ilogtail.LOG)
- Logtail プラグインログファイル (logtail_plugin.LOG)
- コンテナパスマッピングファイル (docker_path_config.json)

起動構成ファイル (ilogtail_config.json)

Logtail の実行パラメータが記載されており、表示および設定できます。JSON 形式のファイルです。

Logtail をインストールした場合、本ファイルに対して次の操作ができます。

- Logtail 実行パラメータの変更

  CPU 使用率や常駐メモリー使用率のしきい値といった一般的な設定を変更します。
- インストールコマンドが正しいことの確認

  config_server_address と data_server_list は、インストール時に指定したものになります。指定されたパラメータのリージョンが Log Service のリージョンと異なる場合、またはアドレスにアクセスできない場合は、インストール中に誤ったパラメータまたはコマンドが使用されたことを意味します。Logtail によるログ収集はできないため、再インストールする必要があります。

 注：

- ファイルは有効な JSON 配列である必要があります。有効な JSON 配列でない場合、Logtail は起動されません。
- ファイルに加えた変更は、Logtail を再起動しない限り反映されません。

下表は、デフォルトの構成項目一覧です。その他の構成項目については、「起動パラメータを設定」をご参照ください。

表 3-1：起動構成ファイルのデフォルト構成項目

構成項目	説明
config_server_address	サーバーから取得する Logtail 構成に設定されているアドレス (インストール時に指定) アクセス可能なアドレスであり、Log Service と同一リージョンのアドレスを指定する必要があります。
data_server_list	データサーバーのアドレス (インストール時に指定) アクセス可能なアドレスであり、パラメータで指定したリージョンは Log Service と同一リージョンに属している必要があります。
cluster	リージョン名

構成項目	説明
endpoint	サービスのエンドポイント
cpu_usage_limit	CPU 使用率のしきい値 (コア数による)
mem_usage_limit	常駐メモリの使用率しきい値
max_bytes_per_sec	送信可能な Raw データの最大サイズ。データ送信速度が 20 Mbit/s を超える場合は適用されない。
process_thread_count	Logtail がログファイルにデータを書き込むために使用するスレッド数
send_request_concurrency	Logtail が同時非同期に送信できるデータパケットの数。 デフォルトでは、Logtail はデータパケットを非同期で送信します。 書き込み TPS が非常に高い場合は、値を大きくします。

ファイルアドレス：

- Linux: /usr/local/ilogtail/ilogtail_config.json
- コンテナ: ファイルは Logtail コンテナに格納され、ファイルアドレスは環境変数 ALIYUN_LOGTAIL_CONFIG に設定されます。アドレスの確認には、Docker inspect $ {logtail_container_name} | grep ALIYUN_LOGTAIL_CONFIG を実行します (例: /Etc/ilogtail/CONF/CN-Hangzhou/FIG)。
- Windows
  - x64: C:\Program Files (x86)\Alibaba\Logtail\ilogtail_config.json
  - x32: C:\Program Files\Alibaba\Logtail\ilogtail_config.json

ファイル例

```
$cat /usr/local/ilogtail/ilogtail_config.json
{
   "config_server_address" : "http://logtail.cn-hangzhou-intranet.log.aliyuncs.com",
   "data_server_list" :
   [
      {
         "cluster" : "ap-southeast-2",
         "endpoint" : "cn-hangzhou-intranet.log.aliyuncs.com"
      }
   ],
   "cpu_usage_limit" : 0.4,
   "mem_usage_limit" : 100,
   "max_bytes_per_sec" : 2097152,
   "process_thread_count" : 1,
   "send_request_concurrency" : 4,
   "streamlog_open" : false
```

```
}
```

AliUid 構成ファイル

Alibaba Cloud アカウントの AliUid が含まれます。AliUid は、サーバーにアクセスしてログを収集する権限のある Alibaba Cloud アカウントであることを証明するものです。別の Alibaba Cloud アカウントの ECS インスタンス、または、オンプレミス IDC のログを収集する場合は、AliUid 構成ファイルを手動で作成します。詳細は、Alibaba Cloud ECS インスタンス以外または他のアカウントの ECS インスタンスからログを収集するをご参照ください。

 注：

- オプションファイルであり、別の Alibaba Cloud アカウントの ECS インスタンス、または、オンプレミス IDC からログを収集する場合にのみ使用されます。
- ファイルには現在の Alibaba Cloud アカウントの AliUid のみを含めることができます。現在の Alibaba Cloud アカウントであっても、RAM ユーザーアカウントの AliUid を含めることはできません。
- ファイル名に拡張子を含めることはできません。
- Logtail に複数の AliUid 構成ファイルを設定できますが、Logtail コンテナに設定できる AliUid 構成ファイルは 1 つのみです。

ファイルアドレス

- Linux: `/etc/ilogtail/users/`
- コンテナ: Logtail コンテナの環境変数 `ALIYUN_LOGTAIL_USER_ID` に設定します。ファイルの確認には、`docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_USER_ID`を実行します。
- Windows: `C:\LogtailData\users\`

ファイル例

```
$ls /etc/ilogtail/users/
15591********** 13292**********
```

カスタム ID ファイル(user_defined_id)

マシングループにカスタム ID を設定した場合に使用されます。 詳細は、「マシングループの作成とカスタム ID 割り当て」をご参照ください。

 注：

- カスタム ID でマシングループの構成にのみ使用されるファイルです。

- マシングループに複数のカスタム ID を設定する場合は、区切り文字で区切ります。

ファイルアドレス

- Linux: /etc/ilogtail/user_defined_id
- コンテナ: Logtail コンテナの環境変数 ALIYUN_LOGTAIL_USER_DEFINED_ID にファイルが設定されています。ファイルの確認には、docker inspect ${logtail_container_name} | grep ALIYUN_LOGTAIL_USER_DEFINED_IDを実行します。
- Windows: C:\LogtailData\user_defined_id

ファイル例

```
$cat /etc/ilogtail/user_defined_id
aliyun-ecs-rs1e16355
```

Logtail 構成ファイル (user_log_config.json)

Logtail のサーバーより取得された Logtail 構成情報が記載されます。JSON 形式のファイルであり、Logtail 構成に変更があると、更新されます。サーバーに Logtail 構成が送信されているかどうかを確認する際に使用します。本ファイルがあり、内容が最新であれば、Logtail 構成情報は正常にサーバーに送信されています。

 注：

- キーまたはデータベースのパスワードを変更する必要がない限り、手動でファイルを変更されないことをお勧めします。
- チケットを起票して、サポートセンターにお問い合わせの際は、本ファイルを添付してください。

ファイルアドレス

- Linux: /usr/local/ilogtail/user_log_config.json
- コンテナ: /usr/local/ilogtail/user_log_config.json
- Windows
  - x64: C:\Program Files (x86)\Alibaba\Logtail\user_log_config.json
  - x32: C:\Program Files\Alibaba\Logtail\user_log_config.json

ファイル例

```
$cat /usr/local/ilogtail/user_log_config.json
{
  "metrics" : {
    "##1.0##k8s-log-c12ba2028*****939f0b$app-java" : {
      "aliuid" : "16542189*****50",
```

```
        "category" : "app-java",
        "create_time" : 1534739165,
        "defaultEndpoint" : "cn-hangzhou-intranet.log.aliyuncs.com",
        "delay_alarm_bytes" : 0,
        "enable" : true,
        "enable_tag" : true,
        "filter_keys" : [],
        "filter_regs" : [],
        "group_topic" : "",
        "local_storage" : true,
        "log_type" : "plugin",
        "log_tz" : "",
        "max_send_rate" : -1,
        "merge_type" : "topic",
        "plugin" : {
          "inputs" : [
            {
              "detail" : {
                "IncludeEnv" : {
                  "aliyun_logs_app-java" : "stdout"
                },
                "IncludeLable" : {
                  "io.kubernetes.container.name" : "java-log-demo-2",
                  "io.kubernetes.pod.namespace" : "default"
                },
                "Stderr" : true,
                "Stdout": true,
              },
              "type" : "service_docker_stdout"
            }
          ]
        },
        "priority" : 0,
        "project_name" : "k8s-log-c12ba2028c*****ac1286939f0b",
        "raw_log" : false,
        "region" :  "cn-hangzhou"、
        "send_rate_expire" : 0,
        "sensitive_keys" : [],
        "tz_adjust" : false,
        "version" : 1
      }
    }
}
```

AppInfo ログファイル (app_info.json)

Logtail の起動時間や、Logtail の IP アドレスおよびホスト名を取得した時間といったあらゆる時間情報が含まれます。マシングループを作成して IP アドレス割り当てるを実施する際に、本ファイルに記載されている IP アドレスが必要になります。

通常、Logtail は次の規則に従ってサーバの IP アドレスを取得します。

- サーバーファイル/etc/hosts で IP アドレスにホスト名を割り当てている場合、Logtail は自動的に IP アドレスを取得します。
- ホストに IP アドレスが割り当てられていない場合、Logtail はホストのプライマリ NIC の IP アドレスを自動的に取得します。

注：

- Logtail の内部情報のみが含まれています。手動でファイルに変更を加えても、Logtail の基本設定は変更されません。
- ホスト名の変更といった、サーバーのネットワーク設定を変更した場合は、Logtail を再起動して新しい IP アドレスを取得します。

表 3-2：フィールド説明

フィールド	フィールドの説明
UUID	サーバーのシリアル番号
hostname	ホスト名
instance_id	ランダムに生成された Logtail 固有の識別子
ip	Logtail の取得した IP アドレス。フィールドが空の場合、Logtail は IP アドレスを取得できず、正常に動作していないことになります。この場合は、サーバーに IP アドレスを設定して Logtail を再起動します。 注： マシングループの識別に IP アドレスを使用している場合、本フィールドはマシングループに設定されている IP アドレスになっているはずです。サーバーに誤った IP アドレスを設定していた場合には、マシングループの IP アドレスを修正し、1 分後に再度ご確認ください。
logtail_version	Logtail クライアントのバージョン
os	OS のバージョン
update_time	Logtail の最終起動時間

ファイルアドレス

- Linux: /usr/local/ilogtail/app_info.json
- コンテナ: /usr/local/ilogtail/app_info.json
- Windows
  - x64: C:\Program Files (x86)\Alibaba\Logtail\app_info.json
  - x32: C:\Program Files\Alibaba\Logtail\app_info.json

ファイル例

```
$cat /usr/local/ilogtail/app_info.json
{
  "UUID" : "",
  "hostname" : "logtail-ds-slpn8",
  "instance_id" : "E5F93BC6-B024-11E8-8831-0A58AC14039E_172.20.3.158_1536053315",
  "ip" : "172.20.3.158",
  "logtail_version" : "0.16.13",
  "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64
",
  "update_time" : "2018-09-04 09:28:36"
}
```

Logtail 操作ログファイル(ilogtail.LOG)

Logtail クライアントに関する実行中の情報が含まれています。ログは INFO、WARN、ERROR の順に重要度が高くなります。 INFO タイプのログは無視して構いません。

 注：

- 収集のエラー診断を実施し、エラーおよび Logtail 操作ログをもとにトラブルシューティングします。
- Logtail 収集の例外に関して、チケットを起票して、サポートセンターにお問い合わせの際は、本ファイルを添付してください。

ファイルアドレス

- Linux: /usr/local/ilogtail/ilogtail.LOG
- コンテナ: /usr/local/ilogtail/ilogtail.LOG
- Windows
  - x64: C:\Program Files (x86)\Alibaba\Logtail\logtail_*.log
  - x32: C:\Program Files\Alibaba\Logtail\logtail_*.log

ファイル例

```
$tail /usr/local/ilogtail/ilogtail.LOG
[2018-09-13 01:13:59.024679]  [INFO]  [3155]  [build/release64/sls/ilogtail/elogtail.
cpp:123]  change working dir:/usr/local/ilogtail/
[2018-09-13 01:13:59.025443]  [INFO]  [3155]  [build/release64/sls/ilogtail/AppConfig.
cpp:175]  load logtail config file, path:/etc/ilogtail/conf/ap-southeast-2/ilogtail_config
.json
[2018-09-13 01:13:59.025460]  [INFO]  [3155]  [build/release64/sls/ilogtail/AppConfig.
cpp:176]  load logtail config file, detail:{
  "config_server_address" : "http://logtail.ap-southeast-2-intranet.log.aliyuncs.com",
  "data_server_list" : [
    {
      "cluster" : "ap-southeast-2",
      "endpoint" : "ap-southeast-2-intranet.log.aliyuncs.com"
    }
```

```
]
```

Logtail プラグインログファイル (logtail_plugin.LOG)

コンテナの標準出力、binlog、http プラグイン、その他プラグインに関する実行中の情報が含まれています。 ログは、INFO 、WARN、ERRORの順に重要度が高くなります。INFO タイプのログは無視していただいて構いません。

CANAL_RUNTIME_ALARM といったプラグインエラーがあり、収集例外を診断を実施する際は、Logtail プラグインログのエラーをもとにトラブルシューティングします。

注：
プラグイン例外に関して、チケットを起票して、サポートセンターにお問い合わせの際は、本ファイルを添付してください。

ファイルアドレス

- Linux: /usr/local/ilogtail/logtail_plugin.LOG
- コンテナ: /usr/local/ilogtail/logtail_plugin.LOG
- Windows: プラグインのログには対応していません。

ファイル例

```
$tail /usr/local/ilogtail/logtail_plugin.LOG
2018-09-13 02:55:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 02:55:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
2018-09-13 03:00:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
2018-09-13 03:00:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##sls-zc-test-hz
-pub$docker-stdout-config,k8s-stdout]    open file for read, file:/logtail_host/var/lib/
docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2
d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.
log    offset:40379573    status:794354-64769-40379963
2018-09-13 03:03:26 [INF] [log_file_reader.go:221] [ReadOpen] [##1.0##k8s-log-
c12ba2028cfb444238cd9ac1286939f0b$docker-stdout-config,k8s-stdout]    open file for
 read, file:/logtail_host/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a
70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1
148b2e2f31bd3410f5b2d624-json.log    offset:40379573    status:794354-64769-
40379963
2018-09-13 03:04:26 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##sls-zc-test-hz-pub
$docker-stdout-config,k8s-stdout]    close file, reason:no read timeout    file:/logtail_ho
st/var/lib/docker/containers/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31
bd3410f5b2d624/7f46afec6a14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2
d624-json.log    offset:40379963    status:794354-64769-40379963
2018-09-13 03:04:27 [INF] [log_file_reader.go:308] [CloseFile] [##1.0##k8s-log-
c12ba2028cfb444238cd9ac1286939f0b$docker-stdout-config,k8s-stdout]    close file
, reason:no read timeout    file:/logtail_host/var/lib/docker/containers/7f46afec6a
14de39b59ee9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624/7f46afec6a14de39b59e
e9cdfbfa8a70c2fa26f1148b2e2f31bd3410f5b2d624-json.log    offset:40379963    status:
794354-64769-40379963
2018-09-13 03:05:30 [INF] [docker_center.go:525] [func1] docker fetch all:start
```

```
2018-09-13 03:05:30 [INF] [docker_center.go:529] [func1] docker fetch all:stop
```

コンテナパスマッピングファイル (docker_path_config.json)

コンテナファイルが収集されるときにのみ自動生成されるファイルです。コンテナファイルのパスと実際のファイルパスとのマッピングが記録されます。JSON 形式のファイルです。

収集のエラー診断を実施する際、**DOCKER_FILE_MAPPING_ALARM** エラー報告がある場合、Logtail は Docker ファイルマッピングに追加することができません。エラーのトラブルシューティングに本ファイルを使用します。

 注：

- 情報のみが含まれているファイルです。ファイルに変更を加えても何の効果もありません。ファイルが削除された場合には自動的に再生成されます。 サービスには影響ありません。
- コンテナログの収集に例外が発生に関し、チケットを起票して、サポートセンターにお問い合わせの際は、本ファイルを添付してください。

ファイルアドレス

/usr/local/ilogtail/ilogtail_config.json

ファイル例

```
$cat /usr/local/ilogtail/docker_path_config.json
{
  "detail" : [
    {
      "config_name" : "##1.0##k8s-log-c12ba2028cfb444238cd9ac1286939f0b$nginx",
      "container_id" : "df19c06e854a0725ea7fca7e0378b0450f7bd3122f94fe3e75
4d8483fd330d10",
      "params" : "{\n   \"ID\" : \"df19c06e854a0725ea7fca7e0378b0450f7bd312
2f94fe3e754d8483fd330d10\",\n   \"Path\" : \"/logtail_host/var/lib/docker/overlay2/
947db346695a1f65e63e582ecfd10ae1f57019a1b99260b6c83d00fcd1892874/diff/var/
log\",\n   \"Tags\" : [\n     \"nginx-type\",\n     \"access-log\",\n     \"_image_name_\",
\n     \"registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest\",\n
 \"_container_name_\",\n     \"nginx-log-demo\",\n     \"_pod_name_\",\n     \"nginx-
log-demo-h2lzc\",\n     \"_namespace_\",\n     \"default\",\n     \"_pod_uid_\",\n     \"
87e56ac3-b65b-11e8-b172-00163f008685\",\n     \"_container_ip_\",\n     \"172.20.4.
224\",\n     \"purpose\",\n     \"test\"\n   ]\n}\n"
    }
  ],
  "version" : "0.1.0"
```

```
}
```

3.2 ネットワークタイプの選択

収集されたログデータは、**Alibaba Cloud** イントラネット、インターネット、または**Global Acceleration**を介して Log Service に送信されます。

ネットワークタイプ

- インターネット: インターネットを介したログデータの送信は、ネットワーク帯域幅の制限を受ける可能性があります。 また、ジッタ、遅延、パケット損失といったネットワークの問題により、データ転送の速度と安定性が影響を受ける可能性もあります。
- **Alibaba Cloud** イントラネット: Alibaba Cloud イントラネットの共有帯域幅はギガビットレベルであるため、インターネットよりも安定しており、ログデータの送信は高速です。 イントラネットには、**Virtual Private Cloud (VPC)** 環境とクラシックネットワーク環境があります。
- **Global Acceleration**: Alibaba Cloud の Content Delivery Network (CDN) のエッジノードを使用したネットワークサービスであり、ログ収集が高速化されます。インターネットと比較して、Global Acceleration はより低遅延に高い安定性を確保できます。

ネットワークタイプの選択

- イントラネット

  サーバータイプ、また、サーバーと Log Service プロジェクトが同じリージョンにあるかどうかで、ログデータが Alibaba Cloud イントラネットを介して送信されるかどうかが決まります。Alibaba Cloud イントラネットを介してログデータを送信できるのは次の場合のみです。

  - アカウントの **ECS** インスタンスおよび **Log Service** プロジェクトのリージョンが同じである
  - 別アカウントの **ECS** インスタンスおよび **Log Service** プロジェクトのリージョンが同じである

  したがって、Log Service プロジェクトは、ECS インスタンスと同じリージョンに作成してログ収集されることを推奨します。ECS インスタンスのログデータは、インターネットの帯域幅を使用せずに、**Alibaba Cloud** イントラネットを介して Log Service に書き込まれます。

   注：

  サーバーに Logtail クライアントをインストールする際、Log Service プロジェクトと同じリージョンを選択します。リージョンが異なる場合、ログデータは収集されません。

- **Global Acceleration**

  海外のオンプレミスサーバー、または、海外の他クラウドベンダーのサーバーからインターネット経由でデータを送信すると、遅延が大きくなり、送信が不安定になるといった問題が発生します。そういった場合に、Global Accelerationを採用します。 Global Accelerationは、Alibaba Cloud CDN のエッジノードを使用するため、ログ収集が高速化されます。インターネット経由でのデータ転送と比較して、Global Acceleration は最小限の転送遅延でより安定したネットワークとなります。

- インターネット

  次の場合には、インターネットを選択することをお勧めします。

  - サーバーは ECS インスタンスであるが、Log Service プロジェクトと同じリージョンではない
  - オンプレミスサーバーや他ベンダーのサーバーである

サーバータイプ	プロジェクトと同じリージョン	**AliUid** を設定する必要がある	ネットワークタイプ
使用中のアカウントの ECS インスタンス	はい	いいえ	Alibaba Cloud イントラネット
	いいえ	いいえ	インターネット、または Global Acceleration
その他アカウントの ECS インスタンス	はい	はい	Alibaba Cloud イントラネット
	いいえ	はい	インターネット、または Global Acceleration
他クラウドベンダーまたはオンプレミスサーバー	-	はい	インターネット、または Global Acceleration

 注：

Log Service は、別のアカウントや他のサーバーの ECS インスタンスの所有者情報を取得できません。 そのため、Logtail クライアントをインストール後に、各サーバに AliUid を設定します。AliUid の設定されていないサーバーとは正常なハートビートが取れず、ログを収集できません。詳細は、「非 Alibaba Cloud ECS インスタンスまたはAlibaba Cloud 別アカウントの ECS インスタンスのログ収集」をご参照ください。

ネットワークタイプの選択例

次の例では、いくつかの一般的なシナリオで適切なネットワークを選択する方法について説明します。

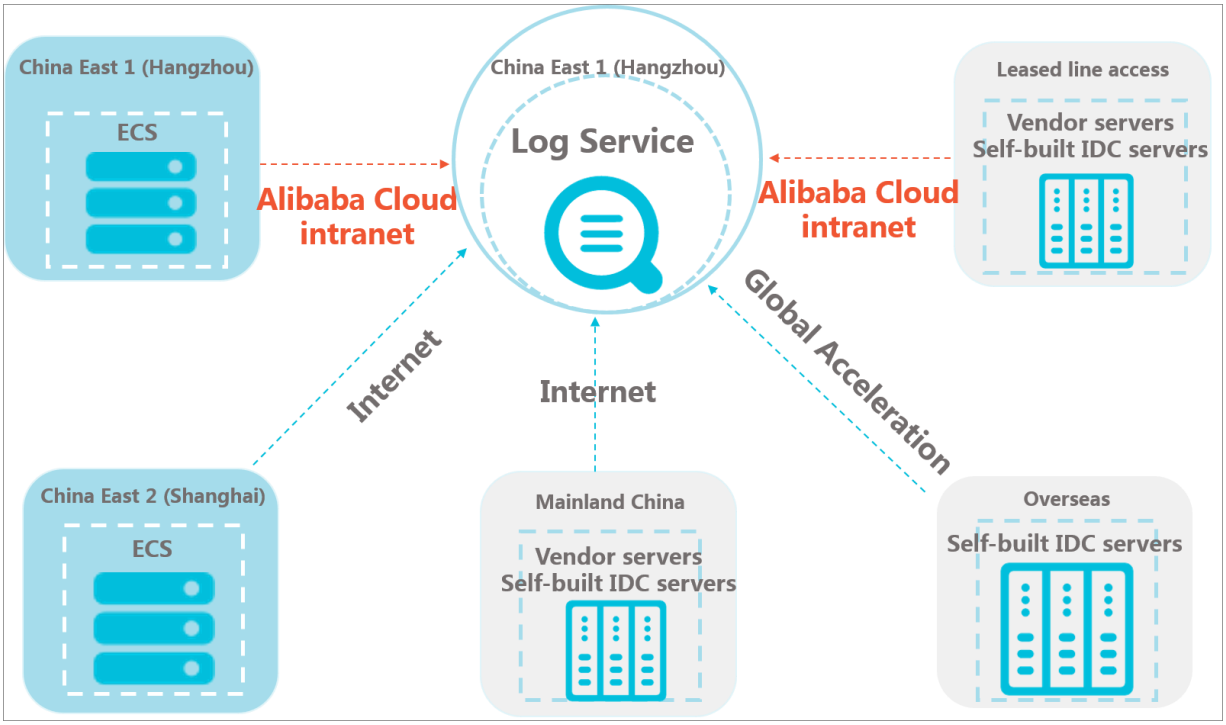
 注：

Global Acceleration シナリオでは、データ収集の速度と信頼性が極めて重要になります。Log Service プロジェクトを香港リージョンに作成しても、接続先のオンプレミスサーバーは世界中に散らばっています。そのため、このシナリオで Logtail クライアントをインストールする際には、香港リージョンの Global Accelleration ネットワークタイプを選択することをお勧めします。Global Acceleration の場合、インターネットよりも高い安定性とパフォーマンスでログデータを送信されます。

シナリオ	**Log Service** プロジェクトのリージョン	サーバータイプ	**ECS** インスタンスのリージョン	**Logtail** クライアントインストール時に選択したリージョン	ネットワークタイプ	**AliUid** を設定する必要がある
ECS とプロジェクトが同じリージョンにある	中国 (杭州)	現在のアカウントの ECS	中国 (杭州)	中国 (杭州)	イントラネット	いいえ
ECS とプロジェクトのリージョンが異なる	中国 (上海)	現在のアカウントの ECS	中国 (北京)	中国 (北京)	インターネット	いいえ
その他アカウント	中国 (上海)	その他アカウントの ECS インスタンス	中国 (北京)	中国 (北京)	インターネット	はい
オンプレミスサーバー	中国 (深セン)	オンプレミス	-	中国 (深セン)	インターネット	はい

シナリオ	**Log Service** プロジェクトのリージョン	サーバータイプ	**ECS** インスタンスのリージョン	**Logtail** クライアントインストール時に選択したリージョン	ネットワークタイプ	**AliUid** を設定する必要がある
Global Acceleration	中国 (香港)	オンプレミス	-	中国 (香港)	Global Acceleration	はい

図 3-3：ネットワークタイプの選択例



クラシックネットワークを VPC に切り替えてから構成を更新

Logtail クライアントをインストール後、ECS インスタンスがクラシックネットワークから VPC に切り替えられた場合は、ネットワーク構成を更新する必要があります。 構成を更新するには、次の手順に従います。

1. Logtail クライアントを管理者として再起動します。

   - **Linux**

     ```
     sudo /etc/init.d/ilogtaild stop
     ```

```
sudo /etc/init.d/ilogtaild start
```

- **Windows**

  コントロールパネルの管理ツールをダブルクリックします。サービスをクリックし、LogtailWorker を右クリックして再起動を選択します。

2. マシングループ構成を更新します。

- カスタム **ID**

  マシングループの識別にカスタム ID を定義している場合は、マシングループ構成を変更することなく VPC ネットワークをそのまま使用できます。

- **IP** アドレス

  マシングループの識別に ECS インスタンスの IP アドレスを使用している場合は、元の IP アドレスを再起動した Logtail クライアントより新たに取得された IP アドレスに置き換える必要があります (app_info.json ファイルの IP アドレスフィールド)。

  app_info.json のファイルパス

  - Linux: /usr/local/ilogtail/app_info.json
  - Windows x64: C:\Program Files (x86)\Alibaba\Logtail\app_info.json
  - Windows x32: C:\Program Files\Alibaba\Logtail\app_info.json

3.3 インストール

3.3.1 Logtail の Linux へのインストール

対応システム

Logtail は、次のリリースの Linux x86-64 (64 ビット) サーバーをサポートしています。

- Aliyun Linux
- Ubuntu
- Debian
- CentOS
- OpenSUSE

Logtail のインストール

Logtail を上書きモードでインストールします。 以前の Logtail がインストールされている場合、インストーラはその Logtail をアンインストールし、/usr/local/ilogtailディレクトリを削除して

から、Logtail をインストールします。 デフォルトでは、インストール後に Logtail が起動し、スタートアップに登録されます。

マシンのネットワーク環境と Log Service のリージョンに基づいて、インストーラをダウンロードします。 インストールごとに異なるパラメータを選択します。

このドキュメントのインストール方法に従って、Logtail をインストールします。 インストールが失敗した場合は、チケットを起票し、サポートセンターへお問い合わせください。

インストール方法

Logtail をインストールするには、インストールスクリプトをダウンロードして実行します。 リージョンとネットワークタイプに基づいて、インストールパラメーターを選択する必要があります。

インストールパラメーター

 注：

Docker または Kubernetes に Logtail をインストールする場合、${yourregion_name} に、次の表のパラメーターを設定します。 対応するインストールステートメントを直接コピーします。

それぞれのリージョンとネットワークタイプに対応するインストールパラメーターは、次のとおりです (対応するインストールステートメントを直接コピーすることを推奨します)。

リージョン	クラシックネットワークと VPC	インターネット (自己構築型 IDC)
中国 (青島)	cn-qingdao	cn-qingdao-internet
中国 (北京)	cn-beijing	cn-beijing-internet
中国 (杭州)	cn-hangzhou	cn-hangzhou-internet
中国 (上海)	cn-shanghai	cn-shanghai-internet
中国 (張家口)	cn-zhangjiakou	cn-zhangjiakou-internet
中国 (フフホト)	cn-huhehaote	cn-huhehaote-internet
中国 (深セン)	cn-shenzhen	cn-shenzhen-internet
中国 (成都)	cn-chengdu	cn-chengdu-internet
中国 (香港)	cn-hongkong	cn-hongkong-internet
米国 (シリコンバレー)	us-west-1	us-west-1-internet
米国(バージニア)	us-east-1	us-east-1-internet
シンガポール	Ap-southeast-1	ap-southeast-1-internet

リージョン	クラシックネットワークと **VPC**	インターネット **(**自己構築型 **IDC)**
オーストラリア (シドニー)	ap-southeast-2	ap-southeast-2-internet
マレーシア(クアラルンプール)	ap-southeast-3	ap-southeast-3-internet
インドネシア (ジャカルタ)	ap-southeast-5	ap-southeast-5-internet
インド (ムンバイ)	ap-south-1	ap-south-1-internet
日本 (東京)	ap-northeast-1	ap-northeast-1-internet
ドイツ (フランクフルト)	eu-central-1	eu-central-1-internet
UAD (ドバイ)	me-east-1	me-east-1-internet
イギリス (ロンドン)	eu-west-1	eu-west-1-internet
中国 (杭州) (financial cloud)	cn-hangzhou-finance	None
中国 (上海) (financial cloud)	cn-shanghai-finance	None
中国 (深セン) (financial cloud)	cn-shenzhen-finance	None

ECS (クラシックネットワーク、VPC)

ECS 上のデータは、Alibaba Cloud イントラネットを介して Log Service に書き込まれます。インターネット帯域幅は使用しません。

リージョンパラメーターの自動選択：

ECS が配置されているリージョンまたは ECS の ID を特定できない場合、Logtail インストーラの auto パラメーターを使用して、Logtail をインストールすることができます。 このパラメーターを指定すると、Logtail インストーラはサーバーを介して#unique_68を取得し、リージョンを自動的に決定します。

手順は次のとおりです。

1. パブリックネットワーク経由で Logtail インストーラを入手します。 このステップでは、パブリックネットワークにアクセスし、約 10 KB のパブリックネットワークトラフィックを使用します。

```
$ wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh;chmod 755 logtail.sh
```

2. インストールパラメーター auto を使用します。 このステップでは、対応するリージョンのインストールプログラムを自動的にダウンロードします。パブリックネットワークトラフィックは使用しません。

```
$ ./logtail.sh install auto
```

region パラメータを手動で選択する

auto パラメータを指定したインストールが失敗した場合は、手動でインストールすることができます。 次のコマンドを実行して、直接インストールしてください。

注 :

- 次のコマンドの ${your_region_name} を、ECS が配置されているリージョン (cn-beijing や cn-hangzhouなど) に置き換えます。
- 内部ネットワークを介してインストーラを取得します。パブリックネットワークトラフィックを使用しません。

```
wget http://logtail-release-${your_region_name}.oss-${your_region_name}-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ${your_region_name}
```

ECS が配置されているリージョンに応じて、次のいずれかのコマンドを実行し、インストールを実行することもできます。

- 中国 (北京)

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

- 中国 (青島)

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

- 中国 (杭州)

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

- 中国 (上海)

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

- 中国 (深セン)

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

- 中国 (張家口)

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

- 中国 (フフホト)

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

- 中国 (成都)

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

- 中国 (香港)

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

- 米国 (シリコンバレー)

```
wget http://logtail-release-us-west-1.oss-us-west-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-west-1
```

- 米国 (バージニア)

```
wget http://logtail-release-us-east-1.oss-us-east-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-east-1
```

- シンガポール

```
wget http://logtail-release-ap-southeast-1.oss-ap-southeast-1-internal.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ap-southeast-1
```

 注：

Log Service は Alibaba Cloud 以外のマシンの所有者情報を取得できません。 したがって、Logtail をインストールした後、手動でユーザー ID を設定する必要があります。 詳細：Alibaba Cloud ECS インスタンス以外または他のアカウントの ECS インスタンスからログを収集する。 ユーザー ID を設定しないと、 Logtail は異常と判断し、ログを収集することができません。

次のコマンドの ${your_region_name} を、Log Service プロジェクトが配置されているリージョン (cn-beijing や cn-hangzhou など) に置き換えます。

```
wget http://logtail-release-${your_region_name}.oss-${your_region_name}.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install ${your_region_name}-internet
```

Log Service プロジェクトが配置されているリージョンに応じて、次のいずれかのコマンドを実行し、インストールを実行することもできます。

- 中国 (北京)

```
wget http://logtail-release-cn-beijing.oss-cn-beijing.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-beijing-internet
```

- 中国 (青島)

```
wget http://logtail-release-cn-qingdao.oss-cn-qingdao.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-qingdao-internet
```

- 中国 (杭州)

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

- 中国 (上海 (中国)

```
wget http://logtail-release-cn-shanghai.oss-cn-shanghai.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shanghai-internet
```

- 中国 (深セン)

```
wget http://logtail-release-cn-shenzhen.oss-cn-shenzhen.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-shenzhen-internet
```

- 中国 (張家口)

```
wget http://logtail-release-cn-zhangjiakou.oss-cn-zhangjiakou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-zhangjiakou-internet
```

- 中国 (フフホト)

```
wget http://logtail-release-cn-huhehaote.oss-cn-huhehaote.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-huhehaote-internet
```

- 中国 (成都)

```
wget http://logtail-release-cn-chengdu.oss-cn-chengdu.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-chengdu-internet
```

- 中国 (香港)

```
wget http://logtail-release-cn-hongkong.oss-cn-hongkong.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install cn-hongkong-internet
```

- 米国 (シリコンバレー)

```
wget http://logtail-release-us-west-1.oss-us-west-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-west-1-internet
```

- 米国 (バージニア)

```
wget http://logtail-release-us-east-1.oss-us-east-1.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh install us-east-1-internet
```


- シンガポール


```
wget http://logtail-release.oss-cn-hangzhou.aliyuncs.com/linux64/logtail.sh -O logtail.sh; chmod 755 logtail.sh; sh logtail.sh install ap-southeast-1-internet
```

```
{
  "UUID" : "0DF18E97-0F2D-486F-B77F-*********",
  "hostname" : "david*******",
  "instance_id" : "F4FAFADA-F1D7-11E7-846C-00163E30349E_*********_1515129548",
  "ip" : "***********",
  "logtail_version" : "0.16.0",
  "os" : "Linux; 2.6.32-220.23.2.ali1113.el5.x86_64; #1 SMP Thu Jul 4 20:09:15 CST 2013;
x86_64",
  "update_time" : "2018-01-05 13:19:08"
}
```

Logtail の更新

Logtail を更新する手順は、Logtail をインストールする手順と同じです。 Logtail を更新すると、Logtail が自動的にアンインストールされ、次に Logtail の最新バージョンがインストールされます。

手動による Logtail の起動と停止

- Logtail の起動

  管理者として次のコマンドを実行します。

  ```
  /etc/init.d/ilogtaild start
  ```

- Logtail の停止

  管理者として次のコマンドを実行します。

  ```
  /etc/init.d/ilogtaild stop.
  ```

Logtail のアンインストール

インストーラ**logtail.sh**をダウンロードします。 詳細については、Logtail のインストールをご参照ください。 シェルモードで次のコマンドを管理者として実行します。

```
wget http://logtail-release-cn-hangzhou.oss-cn-hangzhou.aliyuncs.com/linux64/logtail
.sh -O logtail.sh; chmod 755 logtail.sh; ./logtail.sh uninstall
```

3.3.2 Logtail の Windowsへのインストール

対応システム

Logtailは、Windows Server 2003 (32/64 ビット) 及びそれ以降のバージョンをサポートしています。を参照ください：

- Windows 7 (クライアント) 32bit
- Windows 7 (クライアント) 64bit
- Windows Server 2003 32bit
- Windows Server 2003 64bit

- Windows Server 2008 32bit
- Windows Server 2008 64bit
- Windows Server 2012 64bit

Logtailのインストール

1. インストールパッケージをダウンロードします。

   インストールパッケージは、こちらからダウンロードできます。
2. 現行ディレクトリにlogtail.zipを解凍します。
3. マシンのネットワーク環境とLog Serviceのリージョンに基づいて、Logtailをインストールします。

   Windows PowerShellかcmd.exeを起動して、logtail_installerディレクトリに移動します。マシンのネットワーク環境とリージョンに基づいて、対応するコマンドを実行します。

   インストールコマンド：

リージョン	クラシックネットワークとVPC	インターネット(自己構築型IDC)	Global Acceleration
中国 (青島)	.\logtail_installer.exe install cn-qingdao	.\logtail_installer.exe install cn-qingdao-internet	.\logtail_installer.exe install cn-qingdao-acceleration
中国 (北京)	.\logtail_installer.exe install cn-beijing	.\logtail_installer.exe install cn-beijing-internet	.\logtail_installer.exe install cn-beijing-acceleration
中国 (張家口)	.\logtail_installer.exe install cn-zhangjiakou	.\logtail_installer.exe install cn-zhangjiakou-internet	.\logtail_installer.exe install cn-zhangjiakou-acceleration
中国 (フフホト)	.\logtail_installer.exe install cn-huhehaote	.\logtail_installer.exe install cn-huhehaoteinternet	.\logtail_installer.exe install cn-huhehaote-acceleration
中国 (杭州)	.\logtail_installer.exe install cn-hangzhou	.\logtail_installer.exe install cn-hangzhou-internet	.\logtail_installer.exe install cn-hangzhou-acceleration

リージョン	クラシックネットワークと**VPC**	インターネット**(**自己構築型**IDC)**	**Global Acceleration**
中国 (上海)	.\logtail_installer.exe install cn-shanghai	.\logtail_installer.exe install cn-shanghai-internet	.\logtail_installer.exe install cn-shanghai-acceleration
中国 (深圳)	.\logtail_installer.exe install cn-shenzhen	.\logtail_installer.exe install cn-shenzhen-internet	.\logtail_installer.exe install cn-shenzhen-acceleration
中国 (成都)	.\logtail_installer.exe install cn-chengdu	.\logtail_installer.exe install cn-chengdu-internet	.\logtail_installer.exe install cn-chengdu-acceleration
中国 (香港)	.\logtail_installer.exe install cn-hongkong	.\logtail_installer.exe install cn-hongkong-internet	.\logtail_installer.exe install cn-hongkong-acceleration
米国（シリコンバレー）	.\logtail_installer.exe install us-west-1	.\logtail_installer.exe install us-west-1-internet	.\logtail_installer.exe install us-west-1-acceleration
米国 (バージニア)	.\logtail_installer.exe install us-east-1	.\logtail_installer.exe install us-east-1-internet	.\logtail_installer.exe install us-east-1-acceleration
シンガポール	.\logtail_installer.exe install ap-southeast-1	.\logtail_installer.exe install ap-southeast-1-internet	.\logtail_installer.exe install ap-southeast-1-acceleration
オーストラリア (シドニー)	.\logtail_installer.exe install ap-southeast-2	.\logtail_installer.exe install ap-southeast-2-internet	.\logtail_installer.exe install ap-southeast-2-acceleration
マレーシア (クアラルンプール)	.\logtail_installer.exe install ap-southeast-3	.\logtail_installer.exe install ap-southeast-3-internet	.\logtail_installer.exe install ap-southeast-3-acceleration

リージョン	クラシックネットワークとVPC	インターネット(自己構築型IDC)	Global Acceleration
インドネシア (ジャカルタ)	.\logtail_installer.exe install ap-southeast-5	.\logtail_installer.exe install ap-southeast-5-internet	.\logtail_installer.exe install ap-southeast-5-acceleration
インド (ムンバイ)	.\logtail_installer.exe install ap-south-1	.\logtail_installer.exe install ap-south-1-internet	.\logtail_installer.exe install ap-south-1-acceleration
日本 (日本)	.\logtail_installer.exe install ap-northeast-1	.\logtail_installer.exe install ap-northeast-1-internet	.\logtail_installer.exe install ap-northeast-1-acceleration
ドイツ (フランクフルト)	.\logtail_installer.exe install eu-central-1	.\logtail_installer.exe install eu-central-1-internet	.\logtail_installer.exe install eu-central-1-acceleration
UAE (ドバイ)	.\logtail_installer.exe install me-east-1	.\logtail_installer.exe install me-east-1-internet	.\logtail_installer.exe install me-east-1-acceleration
イギリス（ロンドン）	.\logtail_installer.exe install eu-west-1	.\logtail_installer.exe install eu-west-1	.\logtail_installer.exe install eu-west-1-acceleration

注：

Log ServiceはAlibaba Cloud以外のマシンの所有者情報を取得できません。 そのため、自己構築型IDCや他のクラウドホストでLogtailを使用する場合、Logtailをインストール後に手動でユーザー ID を設定する必要があります。 ユーザーIDを設定しないと、Logtailは異常と判断し、ログを収集することができません。 詳細は、Alibaba Cloud ECS インスタンス以外または他のアカウントの ECS インスタンスからログを収集するを参照してください。

Logtailのアンインストール

Windows PowerShell か cmd.exe を起動してlogtail_installerディレクトリに移動し、次のコマンドを実行します。

```
.\logtail_installer.exe uninstall
```

3.3.3 Logtail 起動設定パラメーター

このドキュメントでは、Logtail 起動設定パラメーターについて説明します。 特別な要件がある場合は、このドキュメントに従って起動パラメーターを設定できます。

シナリオ

次のシナリオでは、Logtail 起動設定パラメーターを設定する必要があります。

- 各ファイルのメタデータ情報（ファイルの署名、収集場所、ファイル名など）は、メモリに保持する必要があります。
- そのため、多数のログファイルを収集する場合は、メモリ使用量が大きくなることがあります。
- ログデータの量が多く、 Log Service 送信されるトラフィックが多いため、CPU 使用率が高くなります。
- Syslog/TCP データストリームが収集されます。

起動の設定

- ファイルパス

  ```
  /usr/local/ilogtail/ilogtail_config.json
  ```

- ファイル形式

  JSON

- ファイルサンプル（部分的な設定項目のみを表示）

  ```
  {
     ...
     "cpu_usage_limit" : 0.4,
     "mem_usage_limit" : 100,
     "max_bytes_per_sec" : 2097152,
     "process_thread_count" : 1,
     "send_request_concurrency" : 4,
     "buffer_file_num" : 25,
     "buffer_file_size" : 20971520,
     "buffer_file_path" : "",
     ...
  ```

```
}
```

一般的な設定パラメーター

パラメーター名	パラメーター説明	値
cpu_usage_limit	CPU 使用率のしきい値。コアごとに計算されます。 たいていの場合、シングルコア処理能力はシンプルモードでは約24 MB /秒で、では約12 MB /秒です。	Double型。 最小値は0.1で、最大値は現在のマシンのCPUコア数です。 デフォルト値は2です。 たとえば、値 0.4 は、Logtail の CPU 使用率がシングルコア CPU の 40% に制限されていることを示します。 しきい値を超えた場合、Logtail は自動的に再起動します。
mem_usage_limit	常駐メモリの使用しきい値。 1000 を超えるファイルを収集するには、しきい値を適切に増やします。	Int型。 MB で測定されます。 最小値は128で、最大値は現在のマシンの有効メモリ値です。 デフォルト値は2048です。 たとえば、値 100 は、Logtail のメモリ使用量が 100 MB に制限されていることを示します。 しきい値を超えた場合、Logtail は自動的に再起動します。
max_bytes_per_sec	Logtail が送信した生データのトラフィック制限は、20MB /秒を超えるストリームには制限されません。	Int型。 バイト/秒で測定されます。 範囲は1024 - 52428800で、デフォルト値は20971520です。 たとえば、値 2097152 は、Logtail のデータ転送速度が 2 MB/秒に制限されていることを示します。

パラメーター名	パラメーター説明	値
process_thread_count	Logtail がログファイルのデータを書き込んだスレッドの数。 通常、シンプルモードでは 24 MB/秒、フルモードでは 12 MB/秒の書き込み速度をサポートします。 デフォルトでは、この値を調整する必要はありませんが、必要に応じてしきい値を増やすことができます。	Int型。 単位：個。 範囲：1~64。デフォルト値は1。
send_request_concurrency	非同期並行処理の数。デフォルトでは、Logtail はデータパケットを非同期で送信します。 書き込み TPS が大きい場合は、より大きな非同期並行性値を設定できます。 Can be supported with a single concurrency of 0.5 Mb/s ~ It is based on the network delay to calculate the throughput of 1 Mb/s network. 注： 単一の並行処理が全体を通して 0.5-1 MB/秒のネットワークをサポートしているという条件に基づいて同時実行数量を計算できます。 実際の並行処理量は、ネットワーク遅延によって異なります。	Int型。 単位：個。 範囲：1~1000、デフォルト値は20です。

パラメーター名	パラメーター説明	値
buffer_file_num	ネットワーク例外が発生するか、または書き込み制限を超えた場合、Logtailは、リアルタイムで解析されたログをキャッシュとして、インストールディレクトリにあるローカルファイルに書き込みます、復帰後にログをLog Serviceに再送します。 このパラメーターは、キャッシュファイルの最大数を制限します。	Int型。 単位：個。 範囲：1~100、デフォルト値は25です。
buffer_file_size	キャッシュファイルの最大Byteを設定できます。 **buffer_file_num * buffer_file_size**がキャッシュファイルが使用できる最大のディスク容量です。	Int型。 単位：個。 範囲：1048576 - 104857600、デフォルト値は20971520 Bytes (20 MB)です。
buffer_file_path	キャッシュファイルを保存するディレクトリ。 このパラメータ値を変更した後、旧キャッシュディレクトリの logtail_buffer_file_ *の形式でファイルを手動で新しいキャッシュディレクトリに移動し、ログを送信後 Logtail がキャッシュファイルを読み込んで削除できるようにする必要があります。	デフォルト値は null で、キャッシュファイルが Logtail インストールディレクトリ(/usr/local/ilogtail)に保存されていることを示します。
bind_interface	ローカルマシンにバインドされているネットワークカードの名前（eth1 など）（Linux バージョンのみがサポートされています）。	このパラメータはデフォルトで空となります。 利用可能なネットワークカードが自動的にバインドされます。 このパラメータが設定されている場合、Logtail はログをアップロードするためにこのネットワークカードを強制的に使用します。

パラメーター名	パラメーター説明	値
check_point_filename	チェックポイントファイルに保存されているフルパス。Logtail のチェックポイントの保存位置をカスタマイズするために使用されます。 Dockerユーザーがこのファイルストレージアドレスを変更し、チェックポイントファイルが存在するディレクトリをホストにマウントすることを推奨します。 そうしないと、チェックポイント情報がないためにコンテナが解放されたときに重複したコレクションが発生します。 例えば、Docker のcheck_point_filenameを/data/logtail/check_point.datとして設定し、-v /data/docker1/logtail:/data/logtailをDockerスタートアップコマンドに追加して /data/docker1/logtailディレクトリを Docker の/data/logtailディレクトリにコピーします。	デフォルト値は/tmp/logtail_check_pointです。

 注：

- 上記の表には、注意が必要な一般的な起動パラメーターのみが記載されています。 ilogtail_config.json のパラメーターが表にない場合、デフォルト値が適用されます。

- 必要に応じて設定パラメーターの値を追加または変更します。 未使用の設定パラメーター（syslog データストリームの収集に関連するパラメーターなど）はilogtail_config.jsonに追加する必要はありません。

設定の変更

1. 必要に応じてilogtail_config.jsonを設定してください。

   変更された設定が有効な JSON 形式であることを確認します。

2. 変更された設定を適用するには、Logtail を再起動します。

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

3.4 マシングループ

3.4.1 概要

Log Service では、マシングループで、ログ収集クライアントである Logtail を実装したサーバーをまとめて管理します。

マシングループは、複数サーバーの集まった仮想のグループです。マシングループにサーバーを追加し、そのマシングループに Logtail 構成を適用することで、複数のサーバーの各 Logtail クライアントを同じ構成にすることができます。

マシングループを次のいずれかの方法で定義します。

- IP アドレス: マシングループにすべてのサーバーの IP アドレスを追加します。グループの各サーバーは、それぞれに固有の IP アドレスで特定することができます。
- カスタム ID: マシングループに ID を設定し、その ID をグループの各マシンに割り当てます。

 注 :

- 他のクラウドベンダーのサーバーやお客様のローカル IDC、または、別のアカウントの ECS インスタンスをマシングループに追加する前に、サーバーまたはインスタンスに AliUid を設定する必要があります。 詳細は、「非 Alibaba Cloud ECS インスタンスまたは別アカウントの Alibaba Cloud ECS インスタンス」をご参照ください。
- Windows サーバーと Linux サーバーを同じマシングループに追加することはできません。

IP アドレスベースのマシングループ

マシングループに複数のサーバーを追加するには、マシングループに各サーバーの IP アドレスを追加します。マシングループ内の全サーバーに Logtail クライアントを一括して設定できるようになります。

- ECS サーバーに hostname がバインドされていなく、かつネットワークタイプが変更されていない場合、その ECS サーバーのプライベート IP アドレスをマシングループに設定します。
- それ以外の場合は、Logtail クライアントの自動取得するサーバーの IP アドレスをマシングループに指定します。 各サーバーの IP アドレスは、サーバー上の app_info.json サーバーファイルの IP アドレスフィールドに登録されています。

> 注：
> app_info.json ファイルには、Logtail クライアントに関する情報が記録されています。 その情報には、Logtail クライアントの自動取得したサーバーの IP アドレスが含まれます。 このファイルの IP アドレスフィールドを手動で変更しても、Logtail クライアントの取得する IP アドレスは変更されません。

Logtail クライアントがサーバーの IP アドレスを自動取得する方法は、次のとおりです。

- サーバーの /etc/hosts ファイルに IP アドレスと hostname がバインドされている場合は、その IP アドレスを Logtail クライアントは自動取得します。
- サーバーの IP アドレスと hostname がバインドされていない場合、Logtail はサーバーのネットワークインタフェース（NI）のプライマリ IP アドレスを自動取得します。

> 注：
> データ収集に **Alibaba Cloud** のイントラネットが使用されるかどうかは、プライベート **IP** アドレスベースのマシングループかどうかは関係がありません。 Alibaba Cloud ECS インスタンスを使用しており、その ECS インスタンスに Logtail をインストールする際に **Alibaba Cloud** イントラネット **(**クラシックネットワークと **VPC)**を選択した場合にのみ、そのサーバーのログデータは Alibaba Cloud イントラネットを介して収集されます。

詳細は、「マシングループの作成と IP アドレス割り当て」をご参照ください。

カスタム ID ベースのマシングループ

IP アドレス以外に、カスタム ID でマシングループを定義する方法もあります。

以下の場合は、マシングループにカスタム ID を定義されることをお勧めします。

- VPC といったカスタムネットワークの場合、1 つの IP アドレスを複数のサーバーが使用していることがあります。そういった場合、Log Service ではサーバー上の Logtail クライアントを管理することはできません。 この問題は、カスタム ID 定義によるマシングループで解決できます。
- 1 つのカスタム ID で、マシングループ内の各サーバーを自動スケーリングすることができます。 新しいサーバーに同じカスタム ID を設定する場合、Log Service は新しいサーバーを自動的に識別してそれをマシングループにそのサーバーを追加します。

通常、システムには複数のモジュールがあります。各モジュールは水平方向にスケーリングできます。つまり、各モジュールに複数のサーバーを追加できます。モジュールごとにマシングループを作成することで、モジュールごとにログを収集できます。そのためには、各モジュールにカスタム ID を作成し、各モジュールの各サーバーにマシングループ ID を設定する必要があります。たとえば、一般的な Web サイトには、HTTP リクエスト処理モジュール、キャッシュモジュール、ロジック処理モジュール、およびストアモジュールがあります。この場合には、HTTP リクエスト処理モジュールのカスタム ID は http_module、キャッシュモジュールは cache_module、ロジック処理モジュールは logic_module、ストアモジュールは store_module に設定することができます。

詳細は、マシングループ作成とカスタム ID 設定をご参照ください。

3.4.2 Alibaba Cloud ECS インスタンス以外または他のアカウントの ECS インスタンスからログを収集する

Logtail を使用して、Alibaba Cloud ECS 以外またはご自身で作成されたものではない ECS インスタンスからログを収集するには、サーバーにログサービスに Logtail をインストールしてユーザー ID（アカウント ID）を設定し、ご自身のアカウントからそのサーバーにアクセスできることを検証してください。 そうしなければ、ハートビートステータスが異常と設定され、Logtail は Log Service にデータを収集することができません。 以下の手順に従って、ユーザー ID（アカウント ID）を設定します。

1. Logtail のインストール

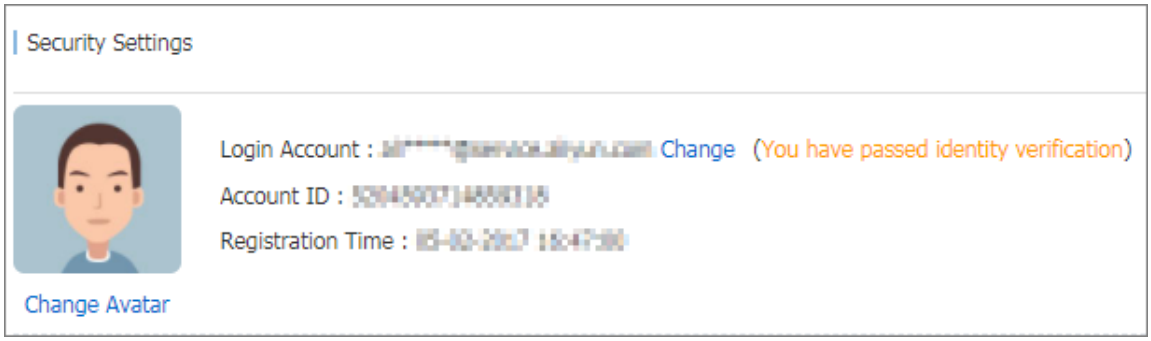
   ログを収集したいサーバーに Logtail をインストールするには、Linux の場合は、Logtail の Linux へのインストールを、Windowsの場合は、Logtail の Windowsへのインストールを参照してください。

2. ユーザ ID の設定

a) Alibaba Cloud アカウント ID の表示

Alibaba Cloud Account Management ページにログインし、Log Service プロジェクトのアカウント ID を表示します。

図 3-4：ユーザ ID の表示



b) サーバーにアカウント ID ファイルを設定

- **Linux**

  /etc/ilogtail/users ディレクトリにアカウント ID の名前の付いたファイルを作成します。ディレクトリが存在しない場合は、手動で作成できます。 以下のように、複数のアカウント ID を 1 台のマシン上に作成することもできます。

  ```
  touch /etc/ilogtail/users/15591**********
  touch /etc/ilogtail/users/13292**********
  ```

  Log Service プロジェクトにデータを収集するために Logtail を必要としない場合は、ユーザー ID を削除してください。

  ```
  rm /etc/ilogtail/users/15591**********
  ```

- **Windows OS**

  ユーザーIDを設定するには、 C:\LogtailData\usersディレクトリにアカウントIDの名前の付いたファイルを作成します。 ユーザー IDを削除するには、このファイルを直接削除してください。

  C:\LogtailData\users\15591**********。

  注：

- マシン上でアカウント ID を設定した後、 クラウドアカウントには、Logtail を使用してマシン上のログを収集する権限が与えられます。 適宜、マシンから不要なアカウント ID ファイルをクリアします。
- ユーザー ID の追加または削除は 1 分以内に有効になります。

3.4.3 Logtail マシングループの作成

Log Service は、マシングループの形式で Logtail クライアントを使用してログを収集する必要のあるすべての ECS（ Elastic Compute Service ）インスタンスを管理します。

Logtail 設定を作成した後、マシングループページでマシングループを作成するか、 データインポートウィザードのマシングループに適用ページでマシングループの作成をクリックしてください。

以下を使用してマシングループを定義できます。

- マシングループ名を定義し、マシングループのイントラネット IP アドレスを追加します。

  マシングループに複数の ECS インスタンスを追加して、Alibaba Cloud ECS インスタンスのイントラネット IP アドレスを追加して Logtail 設定を統一することができます。 非 ECS インスタンス用のマシングループの作成方法については、Alibaba Cloud ECS インスタンス以外または他のアカウントの ECS インスタンスからログを収集するを参照してください。
- ユーザー定義の ID：マシングループの ID を定義し、対応するマシン上で ID を設定して関連付けを行います。

  システムは複数のモジュールで構成されています。 各モジュールの各部分は水平方向にスケーラブルであり、複数のマシンを含むことができます。 ログを個別に収集するには、モジュールごとにマシングループを作成します。 したがって、モジュールごとにユーザー定義の ID を作成し、各モジュールのサーバー上で ID を構成する必要があります。 たとえば、一般的に、ウェブサイトは、フロントエンド HTTP のリクエスト処理モジュール、キャッシュモジュール、ロジック処理モジュール、およびストレージモジュールで構成されいます。 ユーザー定義の ID は、http_module、cache_module、logic_module、および store_module として定義できます。

1. [Log Service コンソール]にログインします。 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。 ログストアリストページを開きます。

2. ログストアリストページで、左側のナビゲーションウィンドウで **LogHub -**ログ収集 > **Logtail** マシングループをクリックします。マシングループページが表示されます。 右上のマシングループの作成をクリックします。

   また、マシングループの作成 （データインポートウィザードのマシングループに適用ページにあります）をクリックすることもできます。

3. グループ名を入力します。

   名前は 3〜128 文字で、小文字、数字、ハイフン（ - ）、およびアンダースコア（_）を含むことができ、小文字または小文字の先頭または末尾に入力する必要があります。

4. マシングループの識別子を選択します。

   - **IP**

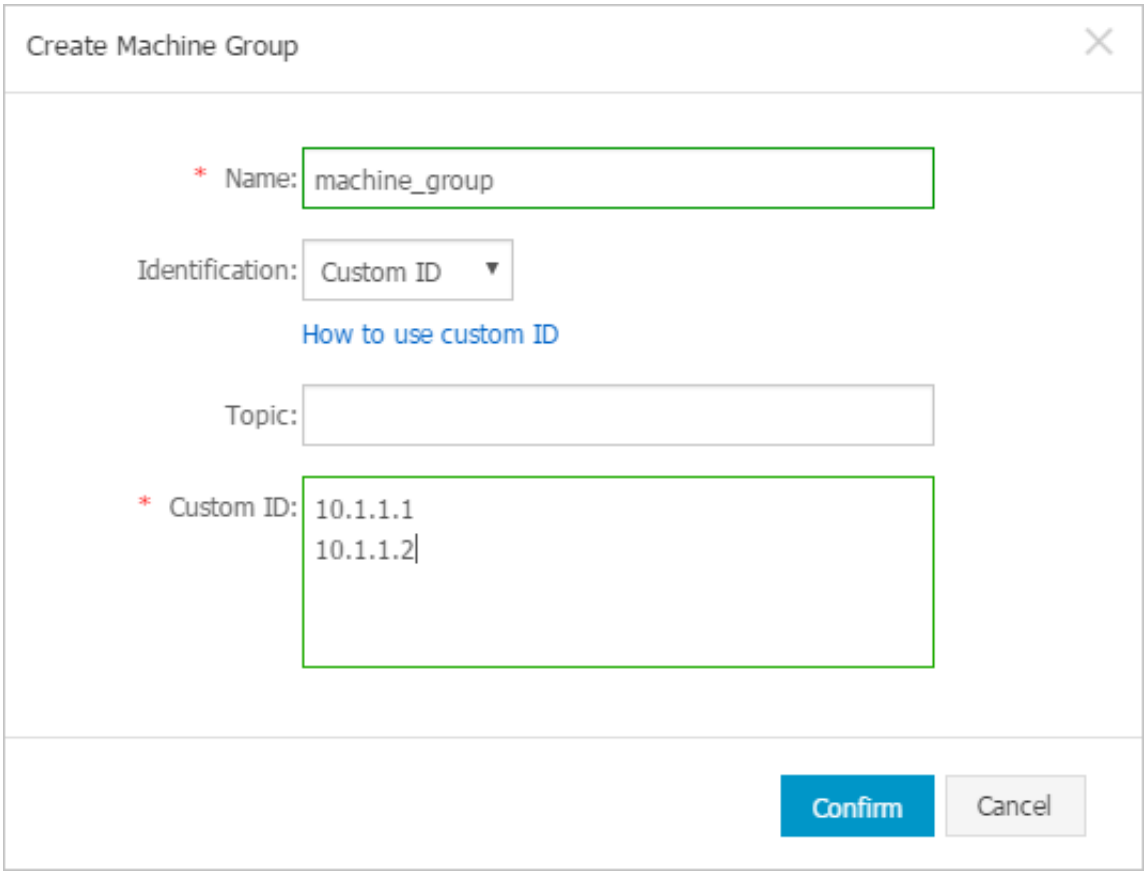
     このオプションを選択して、 IP フィールドに ECS インスタンスのイントラネット**IP** アドレスを入力します。

     注 **:**

     - 入力された ECS インスタンスが現在のログイン Alibaba Cloud アカウントに属していることを確認します。
     - 入力された ECS インスタンスと現在の Log Service プロジェクトが同じ Alibaba Cloud リージョンにあることを確認します。
     - ECS インスタンスのイントラネット IP アドレスを使用していることを確認します。 複数の IP アドレスを区切るには、改行を使用します。
     - Windows ECS インスタンスと Linux ECS インスタンスを同じマシングループに追加することはできません。

- 現在、Log Service は、Cloudtail を使用して Logtail クライアントをリモートにインストールする機能を無効にしています。 Logtail をインストールするには、Logtail の Linux へのインストールを参照してください。

図 3-5：IPアドレス



- ユーザー定義の固有設定

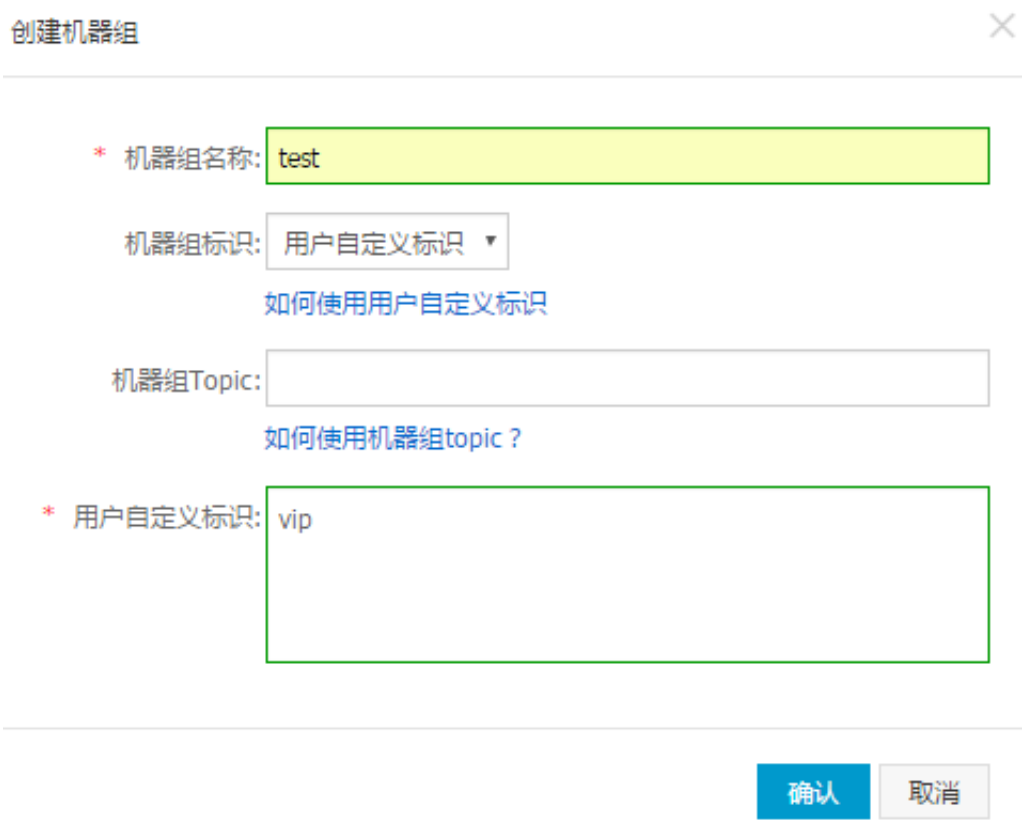
  このオプションを選択した状態で、ユーザー定義の **ID** フィールドに カスタムID を入力します。

  この手順を実行する前に、ログを収集するサーバー上にユーザー定義の ID を作成したことを確認してください。ユーザー定義の ID の使用方法については、マシングループにユーザー定義 ID を設定するを参照してください。

  たとえば、フロントエンドモジュール用のサーバーを追加するために、モジュールのマシンを展開するには、Logtail をインストールし、追加するサーバー上でユーザー定義 ID が http_moduleのファイルを作成して、異なるマシングループの構成を自動的に同期させま

す。 操作が成功したら、マシンステータスをクリックして、追加されたマシンを表示します。

図 3-6：カスタムID



5. マシングループトピックを入力します。
6. 確認をクリックします。

   作成したマシングループは、マシングループページで表示できます。

図 3-7：マシングループリスト

マシングループの作成後、マシングループのリストの表示、マシングループの変更、ステータスの表示、構成の管理、マシングループの削除ができます。

3.4.4 マシングループにユーザー定義 ID を設定する

IP アドレスの他に、ラベルのユーザー定義 ID を使用して、マシングループを動的に定義することができます。

ユーザー定義のIDは、以下のシナリオで有利です。

- 仮想プライベートクラウド（VPC）などのカスタムネットワーク環境では、異なるマシンの IP アドレスが互いに競合する可能性があり、Log Service が Logtail の管理に失敗することがあります。 ユーザー定義の ID は、このような状況を回避するのに役立ちます。
- 複数のマシンは同じラベルを使用してマシングループの自動スケーリングを実装します。新しく追加されたマシンに対して同じユーザ定義 ID だけを設定する必要があります。 Log Service はそれを自動的に識別し、マシングループに追加することができます。

手順

ユーザ定義の ID を使用してマシングループを動的に定義するには、次の手順を実行します。

1. ユーザ定義の ID を有効にする
   - **Linux Logtail**

     /etc/ilogtail/user_defined_idファイルを使ってユーザ定義の ID を設定します。

     たとえば、次のようにマシンのユーザー定義の ID を設定します。

     ```
     #cat /etc/ilogtail/user_defined_id
     ```

   - **Windows Logtail**

     C:\LogtailData\user_defined_idファイルを使用してユーザ定義の ID を設定します。

     たとえば、次のようにマシンのユーザー定義の ID を設定します。

     ```
     C:\LogtailData>more user_defined_id
     aliyun-ecs-rs1e16355
     ```

     aliyun-ecs-rs1e16355をマシングループに追加します。 設定は 1 分後に有効になります。

     > 注：
     > ディレクトリ/etc/ilogtail/またはC:\LogtailData、ファイル/etc/ilogtail/user_defined_idまたは C:\LogtailData\user_defined_idが存在しない場合は、手動で作成してください。

2. マシングループを作成する

   a. マシングループページで、右上のマシングループの作成をクリックします。

   b. マシングループの設定を完了します。

   - グループ名：マシングループの名前を入力します。
   - マシングループ ID ： ユーザー定義 **ID**を選択します。
   - ユーザー定義 ID ：手順 1 で設定したユーザー定義 ID を入力します。

   c. 確認をクリックして、マシングループを作成します。 マシンを拡張するには、追加するサーバーでステップ 1 を完了します。

3. ステータスを表示する

   マシングループページで、マシングループの右側にあるマシンステータスをクリックして、同じユーザー定義の ID とハートビートステータスを使用するマシンのリストを表示します。

その他の操作

ユーザー定義の ID を無効にする

IP アドレスをマシングループの識別情報として使用するには、 user_defined_id ファイルを削除します。 設定は 1 分後に有効になります。

```
rm -f /etc/ilogtail/user_defined_id
```

- **Linux OS**

  ```
  rm -f /etc/ilogtail/user_defined_id
  ```

- **Windows Logtail**

  ```
  Del c: \ logtaildata \ user_defined_id
  ```

有効時間

user_defined_id ファイルを追加、削除、または変更すると、最新の設定がデフォルトで有効になります。

設定をすぐに有効にするには、次のコマンドを実行して Logtail を再起動します。

```
/etc/init.d/ilogtaild stop
/etc/init.d/ilogtaild start
```

- **Linux OS**

  ```
  /etc/init.d/ilogtaild stop
  ```

```
/etc/init.d/ilogtaild start
```

- **Windows Logtail**

  Windows コントロールパネル > 管理ツール > サービスに移動し、サービスリストの LogtailWorker サービスを右クリックし、再起動を選択して設定を有効にします。

例

一般に、システムは複数のモジュールで構成されています。 各モジュールは、複数のマシンを含めることができ、例えば、一般的なウェブサイトは、フロントエンド HTTP リクエスト処理モジュール、キャッシュモジュール、ロジック処理モジュール、およびストレージモジュールで構成されています。 各パートは水平方向に拡張できます。そのため、マシンを追加するときにログをリアルタイムで収集する必要があります。

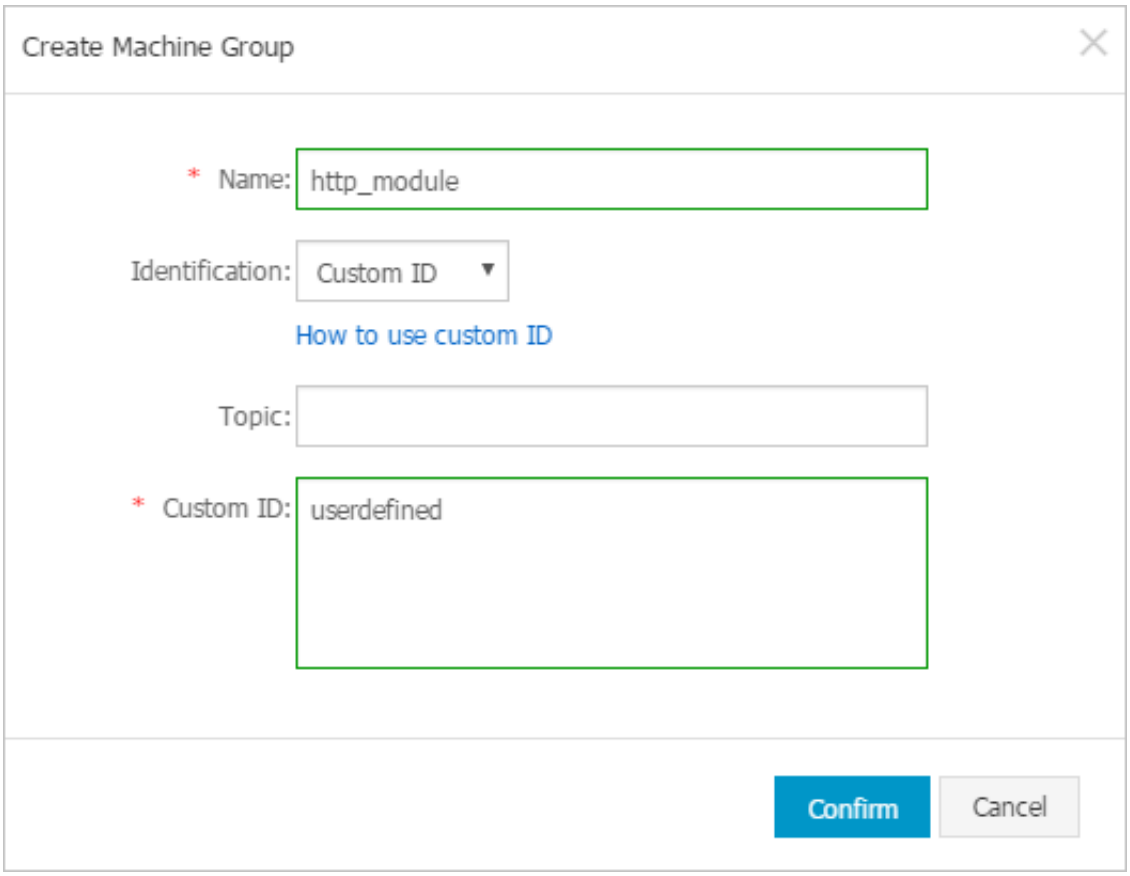
1. ユーザー定義の ID を作成します。

   Logtail クライアントをインストールした後、サーバーのユーザー定義 ID を有効にします。前述の例のモジュールの場合、ユーザー定義の ID は http_module、cache_module、logic_module、および store_module として定義できます。

2. マシングループを作成します。

   マシングループの作成時に、 **User-defined Identity**フィールドに、マシングループが対応するユーザー定義の ID を入力します。 http_module マシングループの次の設定を参照してください。 http_module マシングループは下図の通りです：

   図 3-8：マシングループを作成します。



3. マシングループの右側にあるマシンステータスをクリックして、同じユーザー定義の ID とそのハートビートステータスを使用するマシンのリストを表示します。
4. フロントエンドモジュールにマシン 10.1.1.3 が追加されている場合は、新しく追加されたマシンでステップ 1 を完了します。 操作が正常に終了すると、マシングループステータスダイアログボックスで追加したマシンを表示できます。

3.4.5 マシングループの管理

Log Service は、マシングループの形式で Logtail クライアントを使用してログを収集する必要のあるすべての Elastic Compute Service（ECS）インスタンスを管理します。 マシングループページに移動するには、プロジェクトリスト ページのプロジェクト名をクリックし、左側のナビゲーションで LogHub - ログ収集 > Logtail マシングループをクリックします。 マシングループ

の作成、変更、削除、マシングループのリストとステータスの表示、構成の管理、マシングループ ID の使用を Log Service のコンソールで行うことができます。

マシングループを作成する

以下を使用してマシングループを定義できます。

- IP：マシングループ名を定義し、マシングループのイントラネットIPアドレスを追加します。
- ユーザー定義の ID：マシングループの ID を定義し、対応するマシン上で ID を設定して関連付けを行います。

マシングループの作成方法については、Logtail マシングループの作成を参照してください。

マシングループリストを表示する

1. Log Service コンソールにログオンする。
2. **Logstore**ページで、左側のナビゲーションウィンドウで LogHub - ログ収集> **Logtail** マシングループをクリックします。 マシングループページが表示されます。

   プロジェクト内のすべてのマシングループを表示します。

   図 3-9：マシングループリストを表示する

マシングループを変更する

マシングループを作成したら、必要に応じてマシングループ内の ECS インスタンスを調整できます。

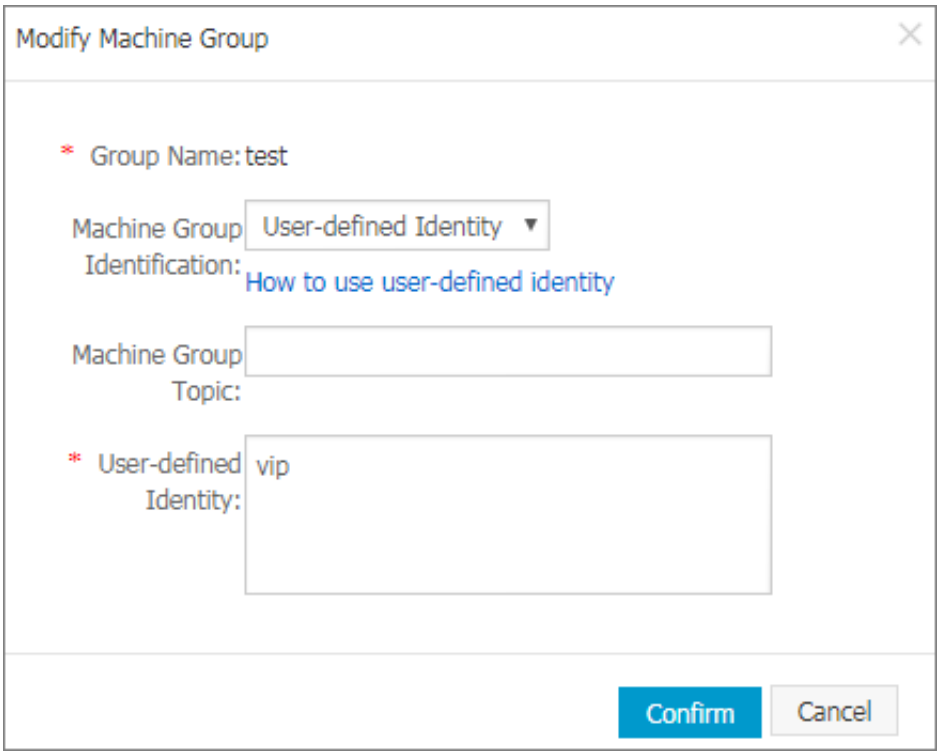
> 注：
> マシングループ名は、マシングループの作成後に変更することはできません。

1. ログサービスコンソールにログインする。
2. **Logstore** ページで、左側のナビゲーションウィンドウで LogHub - ログ収集> **Logtail** マシングループをクリックします。 マシングループページが表示されます。

   Project内のすべてのマシングループが表示されます。
3. マシングループの右側にある変更をクリックします。

4. 設定を変更し、確認をクリックします。

図 3-10：マシングループの変更



ステータス

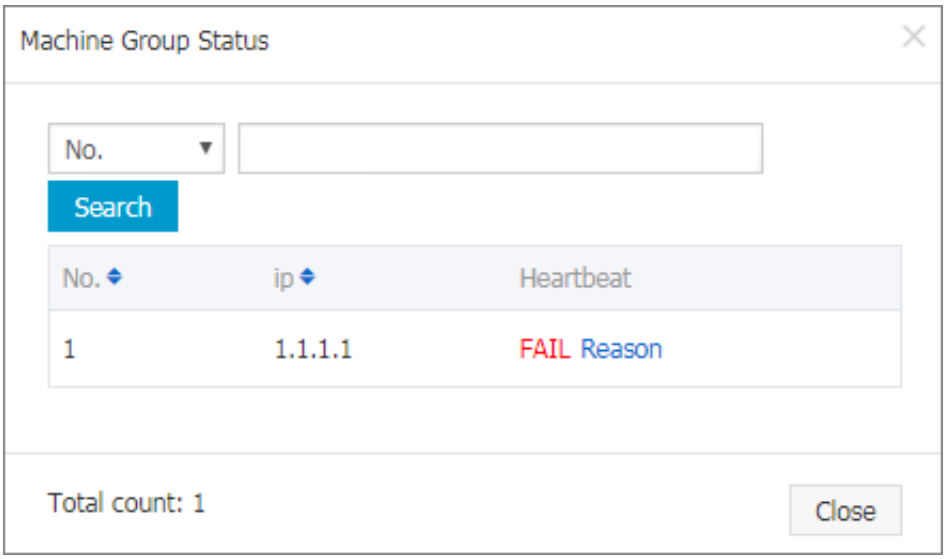
Logtail クライアントがマシングループ内のすべての ECS インスタンスに正常にインストールされたことを確認するには、Logtail クライアントのハートビートステータスを表示します。

1. Log Service コンソールにログインする。
2. プロジェクトページでプロジェクト名をクリックします。 **Logstore**ページで、左側のナビゲーションウィンドウで **LogHub -** ログ収集**> Logtail** マシングループをクリックします。 マシングループページが表示されます。
3. マシングループの右側にあるマシンステータスをクリックします。

   Logtailクライアントがすべての**ECS**インスタンスに正常にインストールされている場合、ECSインスタンスのハートビートステータスは**OK**です。 ハートビートのステータスが**FAIL**の

場合、ページで指示された理由を見つけることを推奨します。 問題が自分で解決できない場合は、チケットを起票し、サポートセンターへお問い合わせください。

図 3-11：マシングループのステータスの表示



注：

- ハートビートステータス**OK**は、 Logtail クライアントがログサービスに正しく接続していることを示します。 マシンをマシングループに追加した後、ハートビートのステータス**OK**を表示する前に、数分の遅延が存在する可能性があります。
- ECS インスタンスのハートビート状態が常に**FAIL**の場合、Logtail の Linux へのインストールとLogtail の Windowsへのインストールを参照してください。

設定の管理

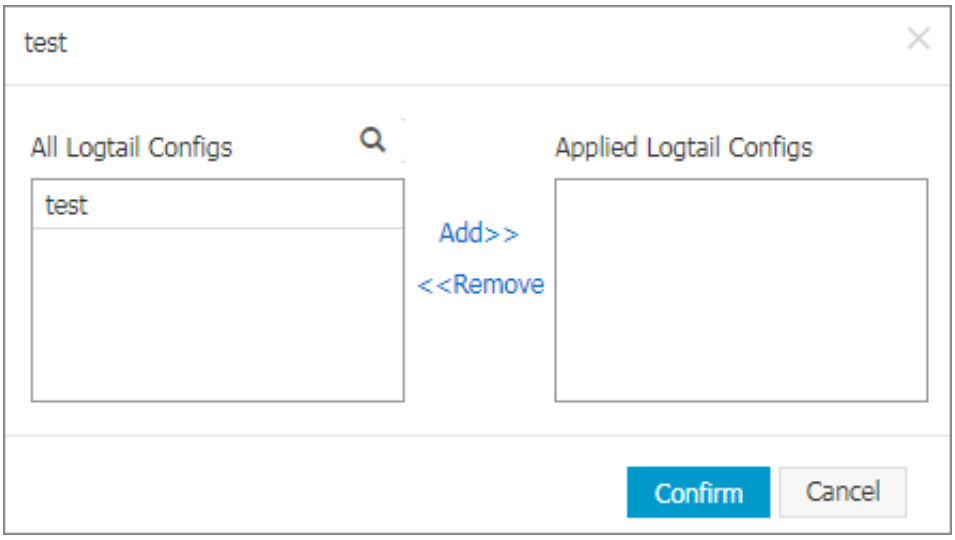
ログサービスは、マシングループを使用してログを収集する必要があるすべてのサーバーを管理します。 1つの重要な管理項目は、Logtail クライアントの収集設定です。 詳細については、テキストファイルの収集および#unique_45を参照してください。 マシングループとの間でLogtail 設定を適用または削除して、収集されるログ、ログの解析方法、各 ECS インスタンスの Logtail によってログが送信される Logstore を決定することができます。

1. Log Service コンソールにログする。
2. **Logstore**ページで、左側のナビゲーションウィンドウで**LogHub - ログ収集> Logtail** マシングループをクリックします。 マシングループページが表示されます。
3. マシングループの右側にある設定をクリックします。

4. Logtail 設定を選択し、追加または削除をクリックして、マシングループとの間で設定を追加または削除します。

   Logtail 設定が追加されると、 マシングループ内の各 ECS インスタンスの Logtail クライアントに発行されます。 Logtail 設定が削除されると、 Logtail クライアントから削除されます。

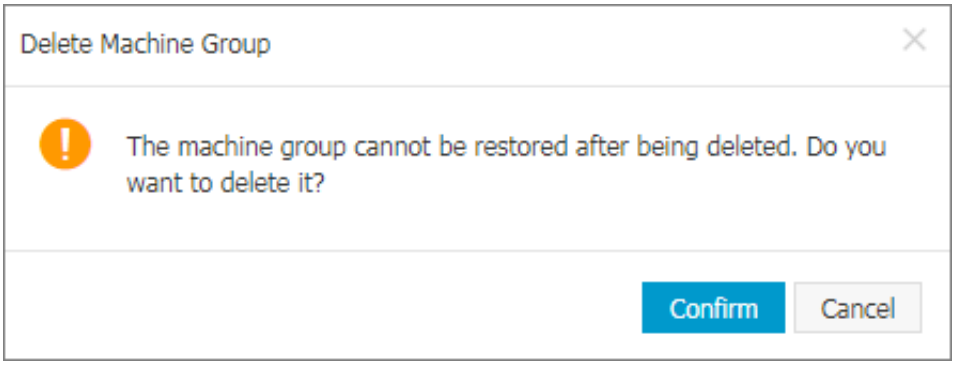
図 3-12：マシングループ設定の管理



マシングループを削除する

1. Log Service コンソールにログインする。
2. プロジェクトページでプロジェクト名をクリックします。 **Logstore**ページで、左側のナビゲーションウィンドウで**LogHub - ログ収集> Logtail** マシングループをクリックします。 マシングループページが表示されます。
3. マシングループの右側にある削除をクリックします。
4. 表示されたダイアログボックスで確認をクリックします。

図 3-13：マシングループの削除



3.4.6 Logtail 設定の作成

Logtail クライアントは、Log Service コンソールの Elastic Compute Service（ECS）インスタンスからログを簡単に収集する方法を提供します。 Logtail クライアントをインストールしたら、Logtail クライアントのログ収集設定を作成する必要があります。 Logtail をインストールする方法については、Logtail の Linux へのインストールとLogtail の Windowsへのインストールを参照してください。 Logstore List ページで Logstore の Logtail 設定を作成および変更することができます。

Logtail 設定を作成する

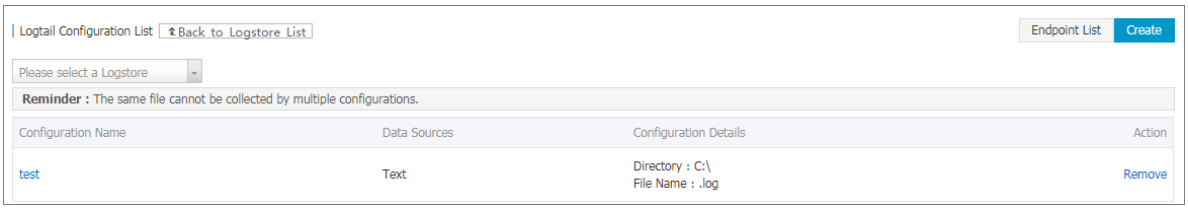
Log Service コンソールでLogtail 設定を作成する方法については、テキストファイルの収集および#unique_45 を参照してください。

Logtail 設定リストを表示する

1. Log Serviceコンソールにログインします。
2. プロダクトページでプロジェクト名をクリックして**Logstore** リストページへ移動します。
3. **Logstore** リストページで、Logstore の右側にある管理をクリックします。 **Logtail** 設定ページが 表示されます。

   このLogstore のすべての設定が、設定名、データソース、設定の詳細など、このページに表示されます。 データソースがテキストの場合、 ファイルパスとファイル名が構成の詳細に表示されます。

   図 3-14 : Logtail 設定リスト



> 注 :
> ファイルは、1つの設定だけで収集できます。

Logtail 設定の変更

1. Log Service コンソールにログインします。
2. プロダクトページでプロジェクト名をクリックします。
3. **Logstore** リストページで、Logstore の右側にある管理をクリックします。 **Logtail** 設定ページが表示されます。

4. 変更するLogtail の名前をクリックします。

   ログ収集モードを変更し、この変更された設定を適用するマシングループを指定できます。 設定変更プロセスは、設定作成プロセスと同じです。

Logtail 設定の削除

1. Log Serviceコンソールにログインします。
2. プロダクトページでプロジェクト名をクリックします。
3. **Logstore** リストページで、Logstore の右側にある管理をクリックします。 **Logtail** 設定ページが表示されます。
4. 削除しようとするLogtail 設定の右側にある削除をクリックします。

   設定が正常に削除されると、この構成を適用したマシングループからアンバインドされ、Logtailは削除された構成のログファイルの収集を停止します。

   > 注：
   > Logstoreを削除する前に、LogstoreのすべてのLogtail 設定を削除する必要があります。

3.5 テキストログ

3.5.1 テキストファイルの収集

Logtail クライアントは、Log Service ユーザーがコンソールで ECSインスタンスまたはローカルサーバーからログを収集する際に補助します。

前提条件

- ログを収集する前に Logtail をインストールする必要があります。 Logtail は、Windows および Linux オペレーティングシステムをサポートしています。インストール方法については、Logtail の Linux へのインストールとLogtail の Windowsへのインストールを参照してください。
- ECSインスタンスまたはローカルサーバーからログを収集するには、ポート80と443が開いていることを確認してください。

制限

- 単一のファイルは、1 つのみの設定で収集できます。 複数の設定を使用してファイルを収集する必要がある場合は、ソフトリンクが推奨されます。 例えば、/home/log/nginx/logの下にあるファイルは、2 つの設定を使用して収集する必要があります。一方は元のパスを設定し、

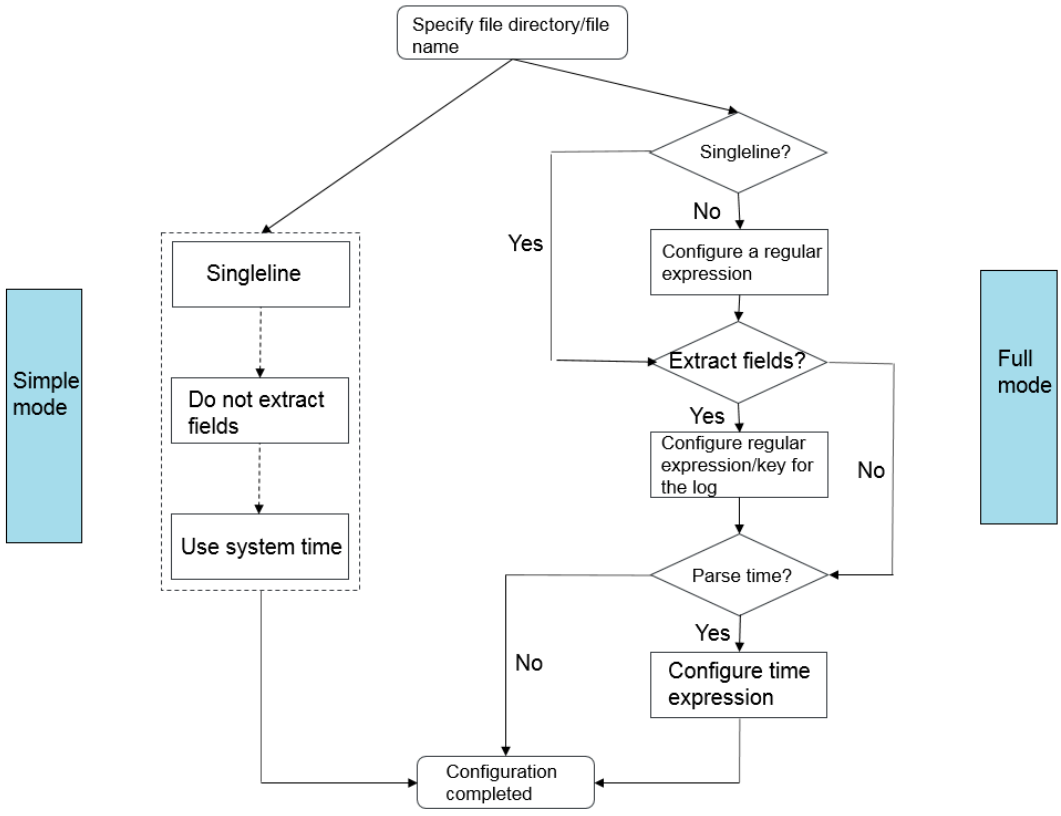
もう一方はフォルダ用に作成されたソフトリンク`ln -s /home/log/nginx/log /home/log/nginx/link_log`を設定します。

- オペレーティングシステムのバージョンの詳細は概要を参照してください。
- クラシックネットワークや仮想プライベートクラウド（VPC）、または Log Service プロジェクトの ECS インスタンスは、同一リージョンでなければなりません。 ソースデータがインターネットを通して送信される場合（IDC の使用法と同様）、リージョンサービスのプロジェクトが存在するリージョンをリージョンの説明に基づいて選択できます。

ログ収集の設定プロセス

ログサービスコンソールでは、シンプルモード、デリミタモード、JSON モード、 フルモードなどのモードでテキストログを収集するように Logtail を設定できます。 シンプルモードとフルモードを例に挙げます。 構成の手順は次の通りです。

図 3-15：手順



手順

1. プロジェクト名をクリックして**Logstore** リストを開きます。
2. 対象の Logstore の右側にあるデータ・インポート・ウィザードをクリックします。

3. データソースを選択します。

   他のソースの下のテキストを選択することで、データソースの設定に入ります。

4. 設定名を指定します。

   設定名の長さは 3～63 文字で、小文字、数字、ハイフン（-）、アンダースコア（_）を含めることができます。小文字の英字で開始し、終了する必要があります。

    注：
   設定名は、後から変更することはできません。

5. ログディレクトリとファイル名を指定します。

   ディレクトリ構造は、完全パスまたはワイルドカードを含むパスである必要があります。

    注：
   *と?のみがディレクトリ内でワイルドカードとして 使用できます。

   ログファイル名は、完全なファイル名またはワイルドカードを含む名前でなければなりません。 ファイル名の規則については、 Wildcard matchingを参照してください。

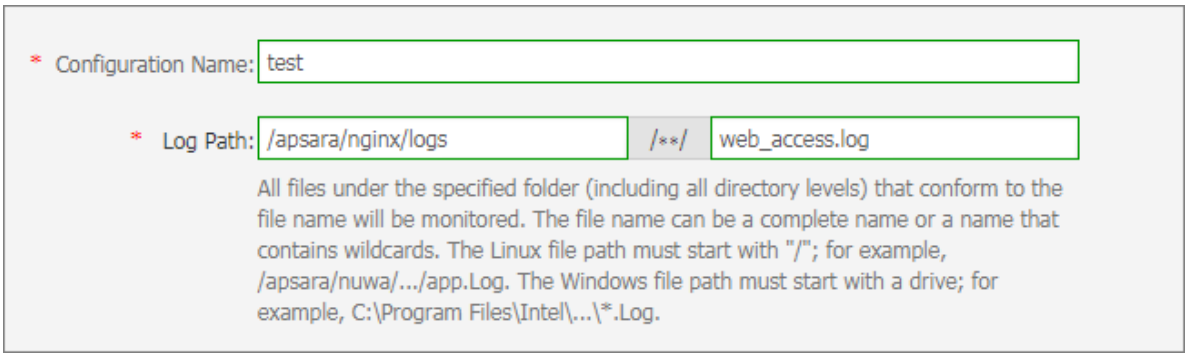
   ログファイルの検索モードはマルチレベルのディレクトリマッチングモードです。つまり、指定したフォルダ（このフォルダのすべてのサブディレクトリを含む）で、ファイル名検索モードに適合するすべてのファイルがモニターされます。こちらに2つの例があります：

   - /apsara/nuwa/ ... /*.Logは接尾辞が.Logで、/apsara/nuwa ディレクトリ（再帰的なサブディレクトリを含む）に存在するファイルを意味します。
   - /var/logs/app_* ... /*.Log*は.Logを含むファイル名を持ち、app_* 検索モード（再帰的なサブディレクトリを含む）に準拠するすべてのディレクトリに存在するファイルを意味します。/var/logsディレクトリの下にあります。

    注：

1つのファイルは、1つの設定でしか収集できません。

図 3-16：ディレクトリとファイル名の指定



6. 収集モードを設定します。

Logtail はシンプルモード、デリミタモード、JSON モード、正規表現モード、およびその他のログ収集メソッドをサポートしています。 詳細はログ収集方法を参照してください。 この例では、シンプルモードと正規表現モードを使用して収集モードの設定を紹介しています。

- シンプルモード

  シンプルモード、つまりシングルラインモードでは、 デフォルトで 1 行のデータがログとして扱われます。すなわち、2行のログはログファイルの改行で区切られます。 システムはログフィールドを抽出しません（つまり、デフォルトでの正規表現は（.*）)。ログ生成時刻として現在のサーバのシステム時刻が使用されます。 より詳細な設定を行うには、設定

をフルモードに変更して設定を調整します。 Logtail 設定の変更方法については、Logtail 設定の作成を参照してください。

シンプルモードで、ファイルディレクトリとファイル名を指定するだけで、 Logtail は行ごとにログを収集します。Logtail はログコンテンツからフィールドを抽出しません。さらに、ログ時間は、ログがキャプチャされた時間に設定されます。

- モード

ログコンテンツ（クロスラインログやフィールド抽出など）のパーソナライズされたフィールド抽出設定を設定するには、フルモードを選択します 。 これらのパラメーターの具体的な意味と設定方法の詳細については、概要を参照してください。

a. Enter ログサンプルを入力します。

ログサンプルを提供する目的は、ログサービスコンソールがログ内の正規表現モードを自動的に抽出するのを容易にすることを目的としています。 実際の環境からログを使用してください。

b. Disable シングルラインを無効にします。

シングルラインモードがデフォルトのオプションです。これは、ログが1行ずつ区切られていることを意味します。 クロスラインログ（Java プログラムログなど）を収集する必要がある場合は、 シングルラインを無効にし、 正規表現を設定する必要があります。

c. 正規表現を設定します。

このオプションは、自動生成と手動入力という 2 つの機能を提供します。 ログサンプルを入力した後、 自動生成をクリックすると、 正規表現が自動的に生成されます。 失敗した場合は、手動モードに切り替えて正規表現を入力して検証することができます。

d. フィールドの抽出を設定します。

ログコンテンツでフィールドを1つずつ分析して処理する必要がある場合は、フィールドの抽出機能を使用して、 指定されたフィールドをキーと値のペアに変換してからサーバーに送信します。 したがって、ログコンテンツ（具体的には正規表現）を解析する方法を指定する必要があります。

ログサービスコンソールでは、構文解析のための正規表現を2通りの方法で指定できます。 最初のオプションは、簡単な対話を介して正規表現を自動的に生成することです。抽出するフィールドを示すために、ログサンプルの”ドラッグ選択”メソッドを使用し

て、” 正規表現の作成 “をクリックすると、ログサービスコンソールが自動的に正規表現を生成します。

このようにして、自らが書くことなく正規表現を生成することができます。 さらに、手動で正規表現を入力することもできます。 **[**手動で正規表現入力**]**をクリックすると、 手動入力モードに切り替えることができます。 式が入力されたら、右側の**[**検証**]**をクリックして、 式がサンプルログを解析して抽出できるかどうかを確認します。

ログ解析の正規表現が自動的に生成されるか手動で入力されるかにかかわらず、抽出された各フィールドに名前を付ける必要があります （つまり、フィールドのキーを設定する必要があります）。

図 3-17 :

e. [システム時間を使用] を設定します。

[システム時間を使用**]** はデフォルトで設定されています。 無効になっている場合は、フィールド抽出中の時間フィールドとして使用する特定のフィールド（値）を指定し、このフィールドの名前をtimeとする必要があります。timeフィールドを選択した後、 自動生成 （時間フォーマット内にあります） をクリックして、このフィールドを解析する

メソッドを生成できます。 ログの時刻形式の詳細については、時刻形式の設定を参照してください。

7. 状況に応じて高度なオプションを設定して、次へをクリックします。

ローカルキャッシュ、オリジナルログのアップロード、トピック生成モード、ログファイルエンコーディング、最大モニタディレクトリの深さ、タイムアウト、およびフィルター設定を要件に応じて設定します。それ以外の場合は、デフォルトのままにしておきます。

構成項目	詳細
ローカルキャッシュ	ローカルキャッシュを有効にするかどうかを選択します。 この機能を有効にすると、ログサービスが利用できないときにログがマシンのローカルディレクトリにキャッシュされ、サービスの復旧後にログサービスに引き続き送信されます。 デフォルトでは、最大で1GBのログをキャッシュできます。
オリジナルログのアップロード	オリジナルログをアップロードするかどうかを選択します。有効にすると、デフォルトで新しいフィールドが追加され、オリジナルログがアップロードされます。
トピック生成モード	• **Null -** トピックを生成しない：デフォルトオプションであり、トピックをヌル文字列として設定し、トピックを入力せずにログを照会できることを示します。 • マシングループのトピック属性：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を明確に区別するために使用されます。 • ファイルパスレギュラー：このオプションを選択すると、正規表現を使用してパスからコンテンツをトピックとして抽出するには、カスタム正規表現を入力する必要があります。 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ます。
カスタム正規表現	トピック生成モードとしてファイルパスレギュラーを選択すると、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使用。 • gbk: GBKエンコーディングを使用
最大モニターディレクトリの深さ	ログソースからログを収集するときに監視するディレクトリの最大深度を指定します。要するに、最大でモニターできるログレベルを指定します。 範囲は0~1000で、0を指定した場合は、現行ディレクトリレベルのみをモニターすることになります。

構成項目	詳細
タイムアウト	指定された時間内に更新がない場合、ログファイルはタイムアウトします。 タイムアウトの次の設定を構成できます。 • タイムアウトにならない：すべてのログファイルを永続的にモニターし、ログファイルがタイムアウトしないように指定します。 • 30分のタイムアウト：ログファイルが30分以内に更新がないければタイムアウトになり、モニターされなくなります。
フィルター構成	フィルター条件に完全に準拠したログのみ、収集できます。 たとえば： • 条件に一致するログを収集します：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示します。 • 条件に適合しないログをフィルターします： - Key:level Regex:^(?!. *(INFO|DEBUG))、INFOまたはDEBUGレベルのログを収集しないことを示します。 - Key:url Regex:. *^(?!.*(healthcheck)). *、urlにヘルスチェックを含むログをフィルターすることを示します。keyがurlでvalueが/inner/healthcheck/jiankong.htmlのログなどは収集されません。 同様の事例についてはregex-exclude-wordとregex-exclude-patternを参照してください。

8. 設定が完了したら、次へをクリックします。

マシングループを作成していなければ、1つを作成する必要があります。 マシングループの作成方法は、「マシングループの作成」 Logtail マシングループの作成を参照してください。

注：

- Logtail 設定が有効になるまでには最大3分かかります。
- IISアクセスログを収集するには、Logstash を使用した IIS ログの収集を参照してください・

- Logtail 設定を作成後、Logtail 設定リストを表示したり、Logtail 設定を変更したり、Logtail 設定を削除することができます。 詳細は、Logtail 設定の作成を参照してください。

図 3-18：マシングループに設定の適用

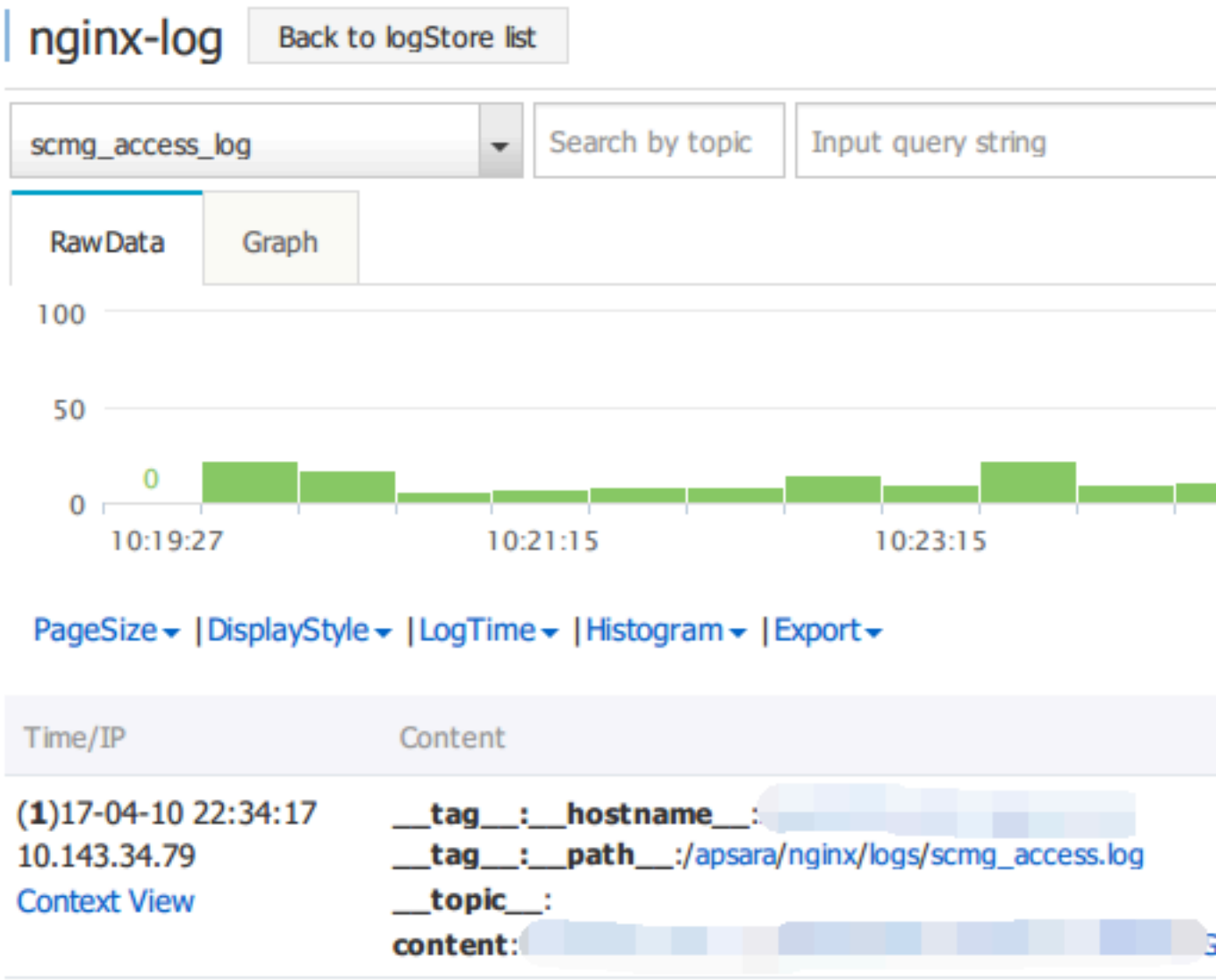
ログサービスは、設定が完了するとログを収集し始めます。

その他の操作

上記の設定が完了すると、ページの指示どおりに検索、分析、ビジュアライゼーション、シッパーとETLを設定することができます。

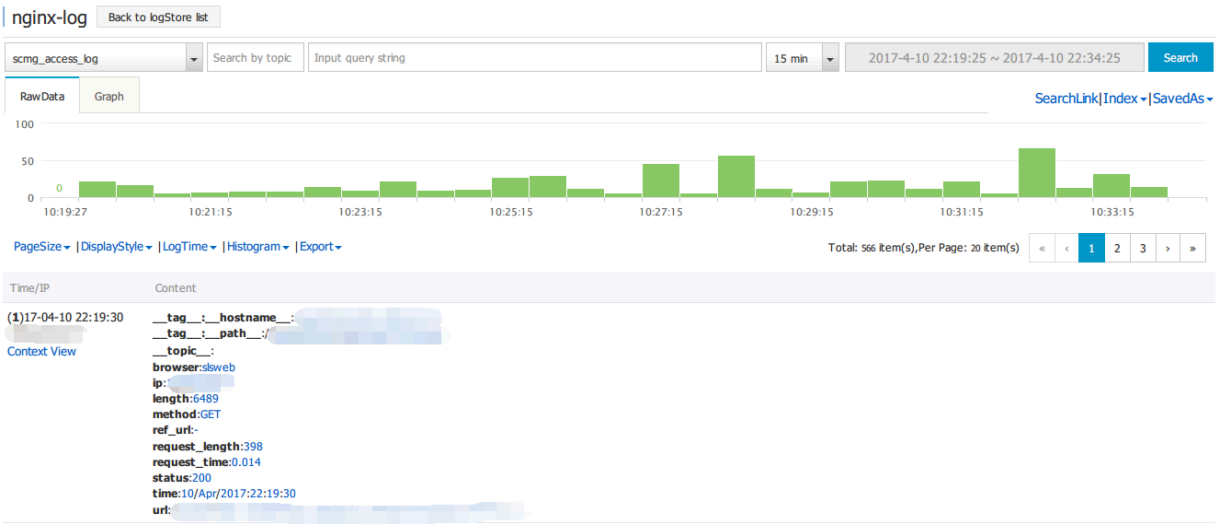
シンプルモードで Log Service に収集されたログは次のとおりです。 すべてのログの内容は、**content**という名前のキーの下に表示されます。

図 3-19 : プレビュー



フルモードでログサービスに収集されたログは、次のとおりです。 設定されたキー値に従って、各ログの内容がログサービスに収集されます。

図 3-20：プレビュー



設定項目をログに記録

Logtail を設定するときは、設定項目を完了する必要があります。 一般的に使用される構成項目の説明と制限は次のとおりです。

構成項目	説明
ログパス	ログ監視ディレクトリとログファイル名がマシン上のファイルと一致することを確認してください。 ディレクトリはファジー一致をサポートしておらず、絶対パスに設定する必要がありますが、ログファイル名はファジー一致をサポートしています。 ワイルドカードを含むパスは、複数のレベルのディレクトリと一致する可能性があります、つまり、指定されたフォルダ（すべてのレベルのディレクトリを含む）の下の、ファイル名に適合するすべてのファイルを監視することができます。
ログファイル名	収集されたログ・ファイルの名前を示します。名前は大文字と小文字を区別し、*.logのようにワイルドカードを含むことがあります。Linuxのファイル名ワイルドカードには、*、?、[...]があります。

ローカルストレージ	短期間のネットワーク中断のために送信できないログをローカルキャッシュに一時的に保存するかどうかを指定します。
First-line log header	正規表現による複数行ログの開始行を示します。 複数行のログ収集（たとえば、スタック情報を持つアプリケーションログ）では、個々のログを区切るために行を使用することはできません。 この場合、複数行ログの開始行を指定する必要があります。 この行が検出されると、最後のログが終了し、新しいログが開始されたことを示します。 そのため、開始ヘッダーに一致するルール、つまり正規表現を指定する必要があります。
ログ解析式	ログ情報を抽出し、Log Serviceでサポートされているログ形式に変換する方法を示します。必要なログフィールドを抽出し、各フィールドに名前を付けるには、正規表現を指定する必要があります。
ログ時刻形式	ログデータのタイムスタンプ文字列の時刻形式を解析する方法を定義します。 詳細は時刻形式の設定を参照してください。

ログの書き込み方法

ログを収集するために Logtail を使用することに加えて、Log Service はログを書くのに役立つ APIとSDKも提供します。

- API を使用してログを書き込む

  ログサービスには、ログを書き込むのに役立つ RESTful API が用意されています。 PostLogStoreLogs インターフェイスを PutLogStoreLogs 使用してデータを書き込むことができます。 完全な API リファレンスについては、Overviewを参照してください。

- SDK を使用してログを書き込む

  ログサービスは、API に加えて、複数の言語（Java、.NET、PHP、およびPython）で SDK を提供しているため、簡単にログを書き込むことができます。 SDK リファレンスについては、SDKリファレンス を参照してください #unique_4。

3.5.2 区切り文字ログ

概要

区切り文字ログは、各ログが改行で区切られています。1 行 1 ログになります。各ログのフィールドは、タブ、スペース、縦線 (|)、カンマ (,)、セミコロン (;)、といった定義されている区切り文字 (単一文字) で接続されます。フィールドのデータに区切り文字を使用する場合は、その文字を二重引用符 (") で囲みます。

一般的な区切り文字ログには、CSV ログ (コンマ区切り) および TSV ログ (タブ区切り) があります。

ログフォーマット

区切り文字ログは、区切り文字で各フィールドは分割されます。なお、単一文字および文字列の 2 つのモードがあります。

- 単一文字モード

  単一文字モードでは、タブ、スペース、縦線 (|)、コンマ (,)、セミコロン (;) といった定義された単一の文字により、ログの各フィールドは分割されます。

  > 注：
  > 二重引用符(") を区切り文字にすることはできません (単一区切り文字のエスケープ文字のため)。

  ログフィールドの値に区切り文字が含まれることはよくあります。ログフィールドを二重引用符"で括ることで、ログフィールドが不適切に分割されないようにします。また、フィールドの値に二重引用符"が含まれる場合には、""と記述してエスケープします。二重引用符("")をフィールドの括りに使用することも、フィールドの値に二重引用符を使用することもできます。それ以外の場合は、区切り文字ログモードではなく、シンプルモードまたは完全モードでフィールドが解析されるようにします。区切り文字ログのログフォーマット定義では対応できません。

  - フィールドを括るために二重引用符 **(")** を使用

    区切り文字を含むフィールドは、二重引用符 (") で括る必要があります。二重引用符は、区切り文字の隣に記述します。二重引用符と区切り文字の間には、スペース、タブ、およびその他の文字が含まれないようにします。

    区切り文字にコンマ (,)、フィールドを括るために二重引用符 (") を使用するとします。ログフォーマットが 1997,Ford,E350,"ac, abs, moon",3000.00である場合、ログは、1997、

Ford、E350、ac, abs, moon、および3000.00の 5 つのフィールドに解析されます。二重引用符で囲まれたac, abs, moonは 1 つのフィールドとみなされます。

- ログフィールドの値に二重引用符 **("")** を使用

  フィールドを括るためにではなく、ログフィールドの値に二重引用符 ("") が含まれる場合は、二重引用符"を""と記述してエスケープする必要があります。フィールドが構文解析される際、""は"に復元されます。

  区切り文字にコンマ (,) を使用し、ログフィールドの値にコンマ (,) および二重引用符 ("") が含まれているとします。フィールドの値のコンマは二重引用符で囲みます。また、二重引用符 (") は""と記述してエスケープします。ログ1999,Chevy,"Venture ""Extended Edition, Very Large""","",5000.00は、1999、Chevy、Venture "Extended Edition, Very Large"、空白フィールド、および5000.00の 5 つのフィールドに解析されます。

- 文字列モード

  文字列モードでは、||、&&&、^_^といった 2 ～ 3 文字を区切り文字に指定します。文字列区切りのログが解析されます。各ログを二重引用符で括る必要はありません。

  > 注：
  > ログフィールドの値に、区切り文字列が含まれないようにします。フィールドの値に区切り文字列が含まれる場合、ログの各フィールドは適切に分割されません。

  区切り文字列が&&のログ1997&&Ford&&E350&&ac&abs&moon&&3000.00は、1997、Ford、E350、ac&abs&moon、および3000.00の 5 つのフィールドに解析されます。

ログサンプル

- 単一文字区切りのログ

```
05/May/2016:13:30:28,10.10.10.1,"POST /PutData? Category=YunOsAccountOpLog
&AccessKeyId=U0Ujpek**********&Date=Fri%2C%2028%20Jun%202013%2006%3A53
%3A30%20GMT&Topic=raw&Signature=******************************** HTTP/1.1",200,
18204,aliyun-sdk-java
05/May/2016:13:31:23,10.10.10.2,"POST /PutData? Category=YunOsAccountOpLog
&AccessKeyId=U0Ujpek**********&Date=Fri%2C%2028%20Jun%202013%2006%3A53
%3A30%20GMT&Topic=raw&Signature=******************************** HTTP/1.1",401,
23472,aliyun-sdk-java
```

- 文字列区切りのログ

```
05/May/2016:13:30:28&&10.200.98.220&&POST /PutData? Category=YunOsAccou
ntOpLog&AccessKeyId=U0Ujpek**********&Date=Fri%2C%2028%20Jun%202013%2006%
3A53%3A30%20GMT&Topic=raw&Signature=******************************** HTTP/1.1&&
200&&18204&&aliyun-sdk-java
05/May/2016:13:31:23&&10.200.98.221&&POST /PutData? Category=YunOsAccou
ntOpLog&AccessKeyId=U0Ujpek**********&Date=Fri%2C%2028%20Jun%202013%2006%
```

```
3A53%3A30%20GMT&Topic=raw&Signature=******************************** HTTP/1.1&&
401&&23472&&aliyun-sdk-java
```

区切り文字ログを収集するように Logtail を設定

Logtail を使用して区切り文字ログを収集する手順については、「Python ログ」をご参照ください。ネットワーク構成およびネットワーク設定に合わせて構成を選択します。

1. **Logstore**リストページで、データインポートウィザードのアイコンをクリックします。
2. データソースを選択します。

   テキストファイルを選択し、次へをクリックします。

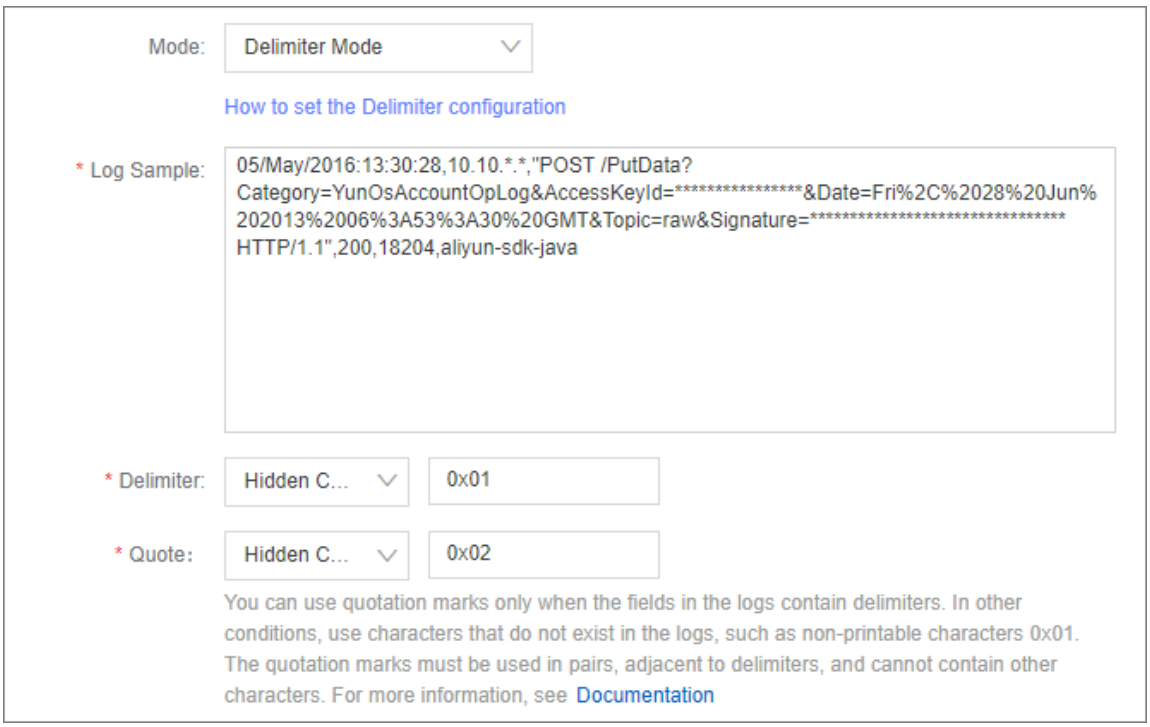
3. データソースを設定します。

   a. 構成名およびログパスを入力します。また、ログ収集モードには区切り文字モードを選択します。

   b. ログサンプルを入力し、区切り文字を選択します。

      ログフォーマットと同じ区切り文字を選択します。ログフォーマットと異なる区切り文字が選択された場合、ログの解析に失敗します。

      図 3-21：データソースを選択



   c. ログ抽出フィールドにキーを割り当てます。

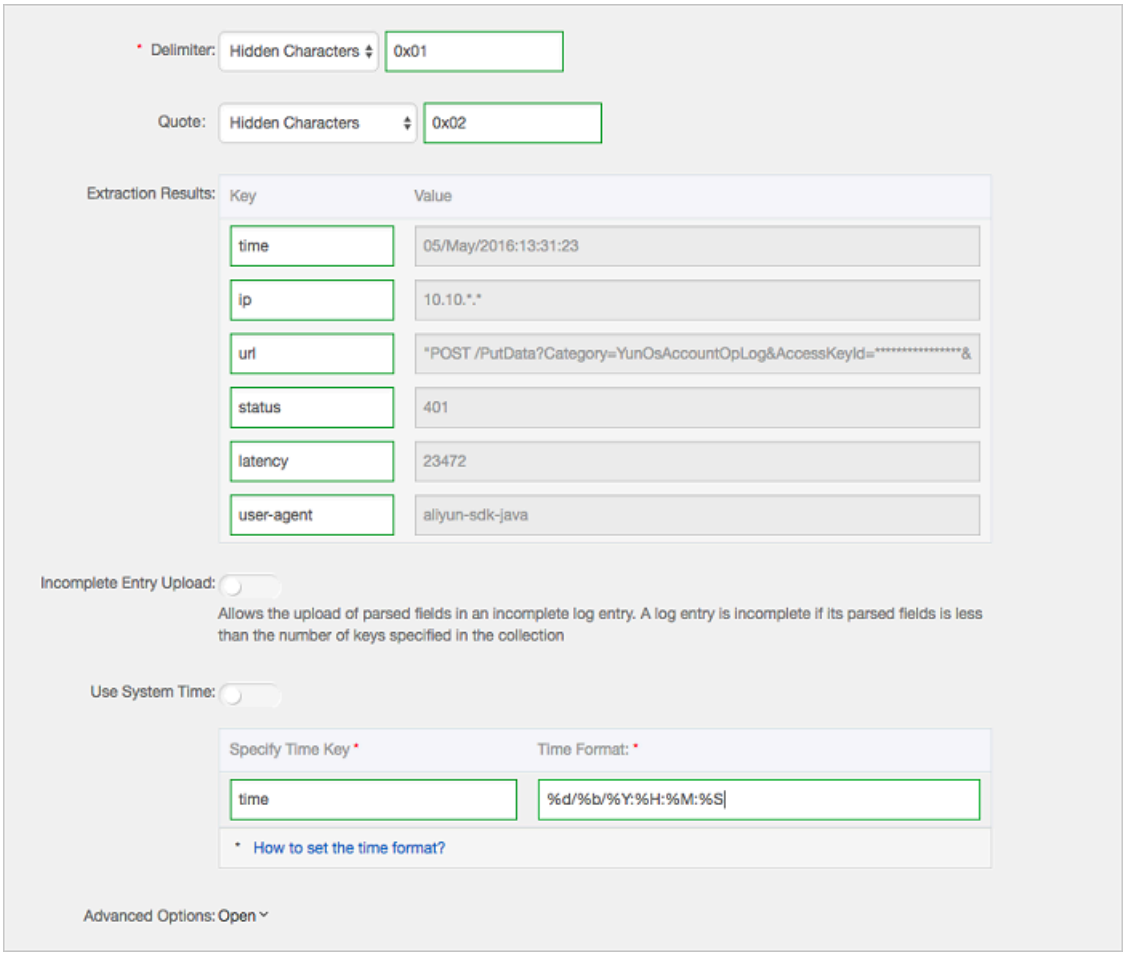
      ログサンプルを入力し、区切り文字を選択したら、選択した区切り文字に基づいてログの各フィールドが抽出されます。抽出された各フィールドは値です。値のキーを指定します。

      上記のログサンプルでは、区切り文字にコンマ (,) が用いられ、6 つのフィールドがログに記録されています。各フィールドのキーをそれぞれ time、ip、url、status、latency、user-agent に指定します。

   d. ログ時間を指定します。

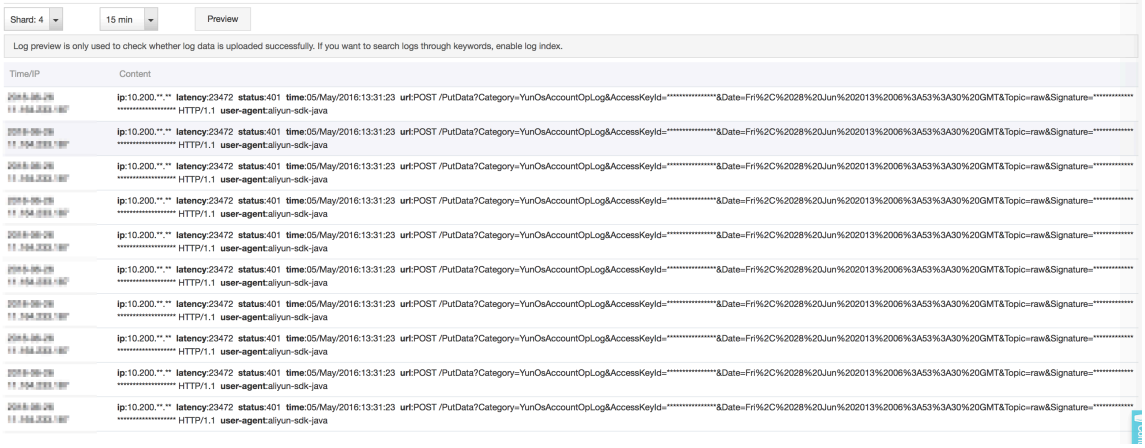
ログ時間には、システム時間またはログのフィールド (例: time フィールド 「05/May/2016:13:30:29」) のいずれかを指定します。時間の書式を設定する方法については、テキストログー時間書式を設定をご参照ください。

図 3-22 : ログ時間の指定



e. コンソールにログをプレビュー表示し、ログが正常に収集されているかどうかを確認します。

図 3-23：ログのプレビュー



3.5.3 JSON ログ

JSON ログは、次の 2 種類の構造体にします。

- オブジェクト: キーと値のペアの集合
- 配列: 値のリスト

Logtail は、Object 型の JSON ログに対応しています。オブジェクトの第 1 階層からキーおよび値が抽出され、フィールド名および値になります。フィールド値には、オブジェクト型、配列型、および、文字列型や数値型といった単純なデータ型が入ることができます。また、JSON ログは、\nで行は区切られます。抽出される各行が、1 つのログになります。

なお、JSON 配列といった Object 型でないデータは自動解析されません。正規表現でフィールドを抽出し、また、シンプルモードで行ごとにログを収集します。

ログサンプル

```
{"url": "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId
>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature
=<yourSignature> HTTP/1.1", "ip": "10.200.98.220", "user-agent": "aliyun-sdk-java", "
request": {"status": "200", "latency": "18204"}, "time": "05/May/2016:13:30:28"}
{"url": "POST /PutData? Category=YunOsAccountOpLog&AccessKeyId=<yourAccessKeyId
>&Date=Fri%2C%2028%20Jun%202013%2006%3A53%3A30%20GMT&Topic=raw&Signature
```

```
=<yourSignature> HTTP/1.1", "ip": "10.200.98.210", "user-agent": "aliyun-sdk-java", "
request": {"status": "200", "latency": "10204"}, "time": "05/May/2016:13:30:29"}
```

Logtail を使用して JSON ログを収集

Logtail を使用して JSON ログを収集する手順については、「5 分でクイックスタート」をご参照ください。 本ドキュメントでは、Logtail の**Log** 収集モードを設定する方法について説明します。

1. **Logstore** リストのデータインポートウィザードをクリックします。
2. データの型を選択します。

   テキストファイルを選択し、次へをクリックします。
3. データソースを設定します。

   a. 構成名およびログパスを入力し、ログ収集モードに**JSON** モードを選択します。

   b. ログ時間にシステム時間を使用するかどうかを選択します。システム時間を使用を有効または無効にします。

   - システム時間を使用を有効にする場合

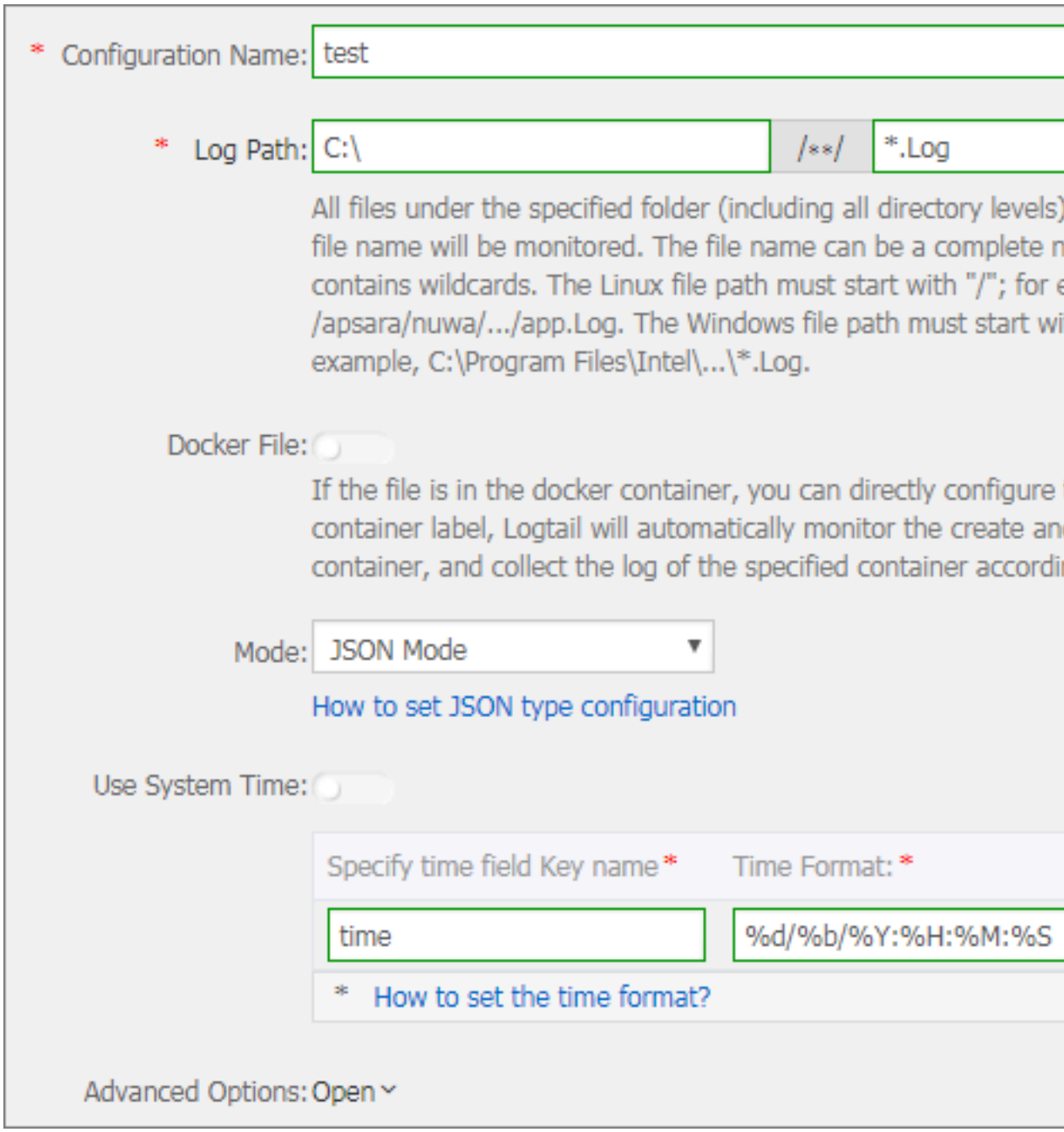
     システム時間を使用すると、ログ時間は、ログ内の time フィールドではなく、Log Service のログを収集した時間が採用されます。
   - システム時間を使用を無効にする場合

     システム時間を使用しない場合、ログ内の time フィールドがログ時間になります。

     システム時間を使用を無効にする場合は、time フィールドのキーおよび時間変換形式を定義する必要があります。たとえば、JSON オブジェクトの`time`フィールド「05/

May/2016:13:30:29」を抽出して、ログ時間にすることができます。ログ時間の書式設定については、テキストログー時間書式の設定をご参照ください。

図 3-24 : JSONログ



3.5.4 Nginx ログ

Nginx のログフォーマットおよびディレクトリは/etc/nginx/nginx.conf設定ファイルに指定します。

Nginx のログフォーマット

ログ設定ファイルに、Nginx ログの出力フォーマットが定義されています (main フォーマット)。

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

次の宣言文には、 main ログフォーマットおよびログデータの書き込み先ファイル名が定義されています。

```
access_log /var/logs/nginx/access.log main
```

フィールド説明

フィールド名	定義
remote_addr	クライアントの IP アドレス
remote_user	クライアントのユーザー名
request	リクエスト URL および HTTP プロトコル
status	リクエストステータス
body_bytes_sent	クライアントへの送信バイト数 (応答ヘッダーを含まない。本変数は、Apache モジュールの modlogconfig で bytes_sent と併せて使用)
connection	接続のシリアル番号
connection_requests	接続で受信したリクエストの数
msec	ログ書き込み時間（ミリ秒単位）
pipe	リクエスト送信を HTTP パイプライン経由にするかどうか (HTTP パイプライン経由でリクエストを送信する場合はp、パイプライン経由でリクエストを送信しない場合は.を指定)
http_referer	Web ページの参照元
“http_user_agent”	クライアントのブラウザ情報 (http_user_agent は、二重引用符で囲むこと)
request_length	リクエストの長さ (リクエスト行、リクエストヘッダー、およびリクエスト本文を含む)

フィールド名	定義
request_time	リクエストの処理時間 (単位: ミリ秒単位、最初のバイトがクライアントに送信されてから、最後の文字がクライアントに送信されるまでの時間)
[$time_local]	一般的なログフォーマットが適用されるローカル時刻 (本変数は角かっこで囲むこと)

ログサンプル

```
192.168.1.2 - - [10/Jul/2015:15:51:09 +0800] "GET /ubuntu.iso HTTP/1.0" 0.000 129 404 168 "-" "Wget/1.11.4 Red Hat modified"
```

Logtail を使用した Nginx ログの収集

1. **Logstore**リストページのデータインポートウィザードアイコンをクリックすると、データインポートウィザードが起動します。
2. データソースを選択します。

   テキストファイルを選択し、次へをクリックします。
3. データソースを選択します。

   a. 構成名およびログパスを入力します。

   b. Nginx のログフォーマットを入力します。

   Nginx 標準のログフォーマット設定を入力します。通常、log_formatで始まります。Log Service によって、ご利用の Nginx キーが自動的に読み込まれます。

   c. 必要に応じて詳細オプションを設定します。設定したら、次へをクリックします。

   詳細については、「詳細オプション」をご参照ください。

   Logtail を設定したら、設定をマシングループに適用します。Nginx ログの収集が開始されます。

3.5.5 Apache ログ

Apache のログフォーマットおよびディレクトリは、/etc/apache2/httpd.conf設定ファイルに指定します。

ログフォーマット

Apache ログの設定ファイルには、出力フォーマットがデフォルトで 2 つ定義されています (結合フォーマットおよび一般フォーマット)。なお、必要に応じてログの出力フォーマットをカスタマイズすることもできます。

- 結合フォーマット

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
```

- 一般フォーマット

```
LogFormat "%h %l %u %t \"%r\" %>s %b"
```

- カスタム化フォーマット

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

Apache のログ設定ファイルに、ログの出力フォーマット、ファイルパス、およびログ名を指定する必要があります。たとえば、次のログ設定ファイルは、結合フォーマットを出力するように指定されており、ログのパスおよび名前は、`/var/log/apache2/access_log`です。

```
CustomLog "/var/log/apache2/access_log" combined
```

フィールドの説明

フォーマット	キー名	説明
%a	client_addr	クライアント IP アドレス
%A	local_addr	ローカルプライベート IP アドレス
%b	response_size_bytes	レスポンスのサイズ (バイト)。空の場合は、ハイフン (-) となります。
%B	response_bytes	レスポンスのサイズ (バイト)。空の場合は、ハイフン (-) となります。
%D	request_time_msec	リクエストの時間 (マイクロ秒単位)
%h	remote_addr	リモートホスト名
%H	request_protocol_supple	リクエストプロトコル
%l	remote_ident	identd のクライアント側のログ名
%m	request_method_supple	リクエストメソッド
%p	remote_port	サーバーのポート
%P	child_process	子プロセス ID
%q	request_query	クエリ文字列。クエリ文字列が存在しない場合は、このフィールドが空文字列となります。

"%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のコンテンツ
%s	status	HTTP ステータスコード
%>s	status	最終の HTTP ステータスコード
%f	filename	ファイル名
%k	keep_alive	keep alive リクエストの数
%R	response_handler	サーバーのハンドラ
%t	time_local	サーバー時間
%T	request_time_sec	リクエストの時間 (秒単位)
%u	remote_user	クライアントのユーザー名
%U	request_uri_supple	リクエストした URL のパス (クエリを含まない)
%v	server_name	サーバー名
%V	server_name_canonical	UseCanonicalName 設定に従ったサーバー名
%I	bytes_received	サーバーの受信バイト数 (mod_logio モジュールが有効な場合のみ)
%O	bytes_sent	サーバーの送信バイト数 (mod_logio モジュールが有効な場合のみ)
%{User-Agent}i	http_user_agent	クライアント情報
"%{Rererer}i"	http_referer	ソースページ

サンプルログ

```
192.168.1.2 - - [02/Feb/2016:17:44:13 +0800] "GET /favicon.ico HTTP/1.1" 404 209 "http://localhost/x1.html" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/48.0.2564.97 Safari/537.36"
```

Logtail を使用した Apache ログの収集

1. **Logstore**ページで、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. データタイプを選択します。

   APACHE アクセスログを選択します。

3. データソースを設定します。

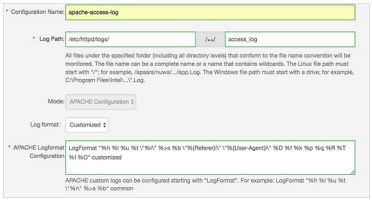
   a. 構成名およびログパスを入力します。

   b. ログフォーマットを選択します。

   c. ログフォーマットをカスタマイズする場合は、**APACHE** のログフォーマット構成に入力します。

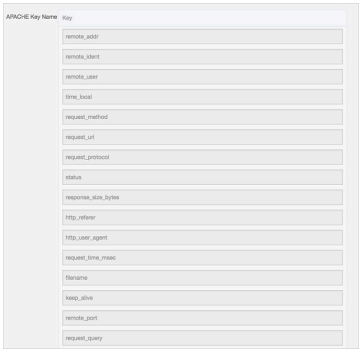
      Apache 標準の設定ファイルのログフォーマット設定欄に入力します。通常、設定ファイルは「LogFormat」で始まります。

      > 注：
      > ログフォーマットドロップダウンリストより一般または結合を選択した場合には、**APACHE** のログフォーマット設定欄に該当するログフォーマットが自動的に入力されます。入力された内容が、ローカルの Apache 設定ファイルに定義されているフォーマットと同じであることを確認します。

   d. **APACHE** キー名を確認します。

      Log Service によって、お客様の Apache キーが自動的に読み込まれます。現在のページの APACHE キー名を確認します。

   e. （オプション）詳細オプションを設定します。

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。

パラメーター	説明
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。 • 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。 • 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。 • 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。 • 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。 • 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 例： • 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。 • 条件に適合しないログをフィルターします： - 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。 - 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。 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。

4. 次へをクリックします。
5. 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。

   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。

   Logtail 設定をマシングループに適用すると、Log Service は設定に従って Apache ログを収集します。データインポートウィザードの手順に従って、インデックスおよびログ送信を設定します。

3.5.6 テキストログの設定と解析

ログラインの分離方法を指定する

一般に、完全なアクセスログ（例えば、 Nginx アクセスログ）は 1 行を占めます。2 つのログは改行で区切られています。 たとえば、次の 2 つのシングルラインアクセスログを参照してください。

```
10.1.1.1 - - [13/Mar/2016:10:00:10 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"
```

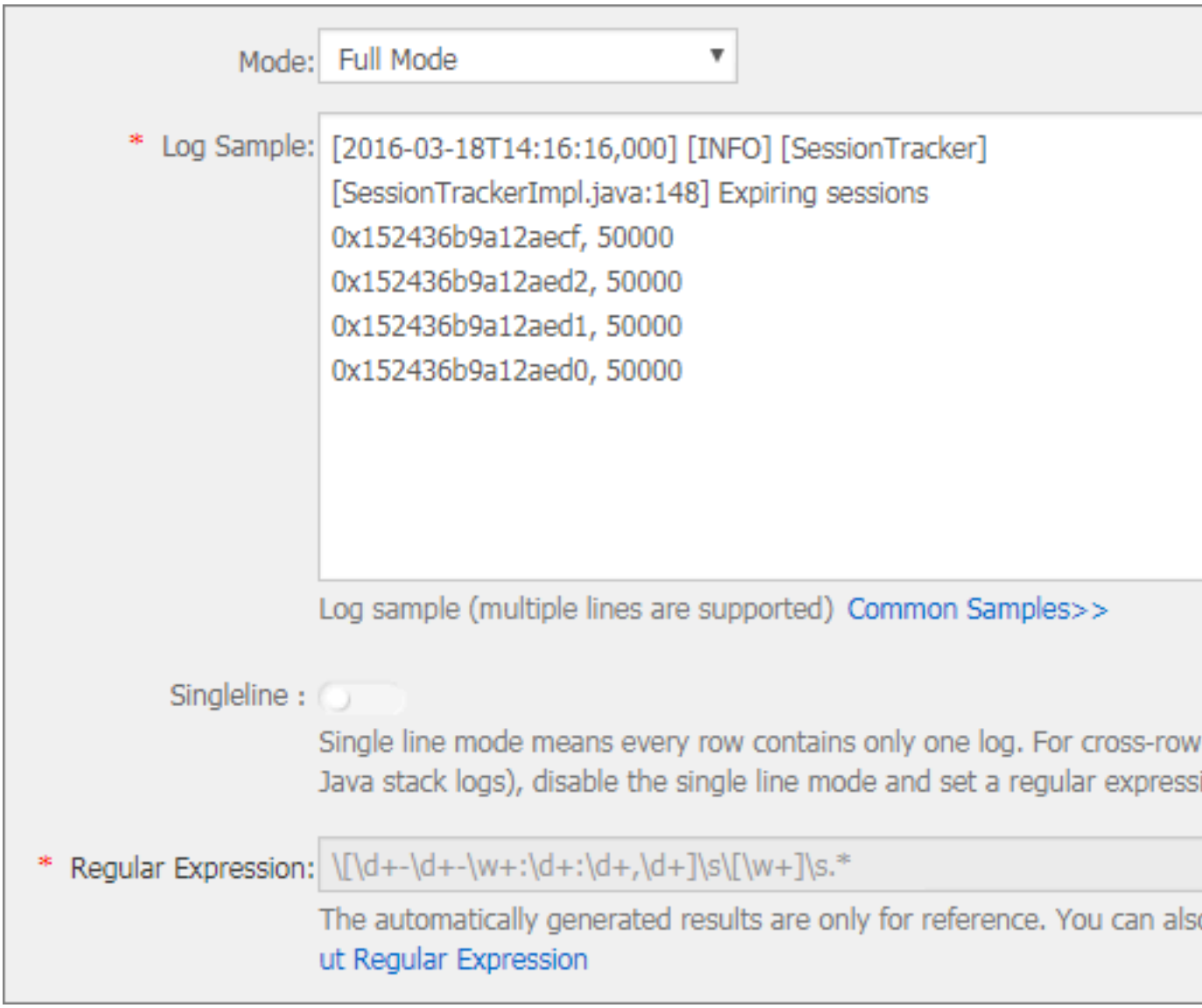
```
10.1.1.1 - - [13/Mar/2016:10:00:11 +0800] "GET / HTTP/1.1" 0.011 180 404 570 "-" "
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; 360se)"
```

Java アプリケーションの場合、プログラムログは通常複数の行にまたがります。 特性ログヘッダーは、 2 つのログを分離するために使用されます。 たとえば、次の Java プログラムログを参照してください。

```
[2016-03-18T14:16:16,000] [INFO] [SessionTracker] [SessionTrackerImpl.java:148] Expiring
 sessions
0x152436b9a12aecf, 50000
0x152436b9a12aed2, 50000
0x152436b9a12aed1, 50000
0x152436b9a12aed0, 50000
```

前述の Java ログには、日時形式の開始フィールドがあります。 正規表現は \[\d+-\d+-\w+:\d+:\d+,\d+]\s.* です。 以下のように、コンソールで設定を完了してください。

図 3-25：フルモード解析の正規表現



ログフィールドを抽出する

Log Service data modelsによれば、ログには 1 つ以上のキーと値のペアが含まれています。 分析のために指定したフィールドを抽出するには、正規表現を設定する必要があります。 ログコンテンツを処理する必要がない場合、ログはキーと値のペアと見なすことができます。

上記のアクセスログの場合：

- フィールドが抽出されるとき

  Regular expression: `(\S+)\s-\s-\s\[(\S+)\s[^]]+]\s"(\w+). *`, Extracted contents: `10.1.1.1`, `13/Mar/2016:10:00` and `GET`.
- フィールドが抽出されない場合

  Regular expression: `(. *)`, Extracted contents: `10.1.1.1 - - [13/Mar/2016:10:00:10 +0800] " GET / HTTP/1.1" 0.011 180 404 570 "-" "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5 .1; 360se)""`

ログ時間を指定する

Log Service data models によると、ログには UNIX のタイムスタンプ形式の時間フィールドが必要です。 現在、ログ時間は、 Logtail がログ内容を収集する時のシステム時刻に設定することができます。

上記のアクセスログの場合：

- ログコンテンツの時間フィールドを抽出する`Time: 13/Mar/2016:10:00:10`Time expression: `%d/%b/%Y:%H:%M:%S`
- ログが収集されるシステム時刻 Time: Timestamp when the log is collected.

3.5.7 時刻形式の設定

Log Service の各ログには、ログ生成時間を記録するタイムスタンプがあります。 ログファイルからログデータを収集する場合、Logtail は各ログのタイムスタンプ文字列を抽出し、タイムスタンプに解析する必要があります。 したがって、解析用のタイムスタンプ形式を指定する必要があります。

Linux では、Logtail は strftime 関数が提供するすべての時刻形式をサポートしています。 Logtail は、strftime 関数によって定義されたログ形式で表現できるタイムスタンプ文字列のみを解析し、使用します。

 注：

- ログタイムスタンプは秒単位で正確です。 したがって、ミリ秒やマイクロ秒などのその他の情報を必要とせずに、時間形式を秒に設定するだけで済みます。
- また、他の情報ではなく、時間フィールドを構成するだけで済みます。

Logtail でサポートされる一般的なログ時刻形式

ログのタイムスタンプ文字列には、さまざまな形式があります。 Logtail では次の表で示される複数のログ時刻形式をサポートして、設定を容易にしています。

形式	説明	例:
%a	1 週間のうちの 1 日の省略形。	金
%A	1 週間に 1 日の名前。	金曜日
%b	1 か月の省略形。	Jan
%B	1 ヶ月の名前。	1 月
%d	10 進式の月の日 [01, 31]	07、31
%h	月の省略形。%b と同じ。	Jan
%H	24 時間形式の時間	22
%I	12 時間形式の時間	11
%m	10 進形式の月。	08
%M	10 進形式の分。 [00, 59]	59
%n	改行。	改行
%p	12 時間形式の期間の省略。	AM または PM
%r	12 時間形式の時刻 (%I:%M:%S %p に相当)	11:59:59 AM
%R	時間と分で表される時間 (%H:%Mに相当)	23:59
%S	10 進形式の秒。 [00, 59]	59
%t	タブ文字	タブ文字
%y	10 進形式ではない年 [00, 99]	04 または 98
%Y	10 進形式の年。	2004 または 1998
%C	10 進形式の世紀。 [00, 99]	16
%e	10 進形式の月の日。 [1, 31] 1 桁の前にスペースがあります。	7 または 31

形式	説明	例:
%j	10 進形式の年の日。 [00, 366]	365
%u	10 進形式の曜日。 [1, 7] 1 は月曜日を表します。	2
%U	週の週番号 (日曜始まり) [00, 53]	23
%V	週の週番号 (月曜始まり)。 1 月初めの週が 4 日以上であれば、その週がその年の最初の週と見なされます。 1 月初めの週が 4 日未満の場合、その次の週がその年の最初の週と見なされます。 [01, 53]	24
%w	10 進形式の月の日 [0, 6] 0 は日曜日を表します。	5
%W	週の週番号 (日曜始まり) [00, 53]	23
%c	標準日時	長い日付や短い日付などの詳細情報を指定するには、サポートされている前述の形式を使用してより正確な式を使用することを推奨します。
%x	標準日付	長い日付や短い日付などの詳細情報を指定するには、サポートされている前述の形式を使用してより正確な式を使用することを推奨します。
%X	標準時刻	長い日付や短い日付などの詳細情報を指定するには、サポートされている前述の形式を使用してより正確な式を使用することを推奨します。
%s	UNIX タイムスタンプ	1476187251

例

以下は、共通のログ時刻形式、例、および対応する時刻表現式です。

ログ時刻形式	例	時間式
カスタマイズ	2017-12-11 15:05:07	%Y-%m-%d %H:%M:%S
カスタマイズ	[2017-12-11 15:05:07.012]	[%Y-%m-%d %H:%M:%S]
RFC822	02 Jan 06 15:04 MST	%d %b %y %H:%M
RFC822Z	02 Jan 06 15:04 -0700	%d %b %y %H:%M
RFC850	Monday, 02-Jan-06 15:04:05 MST	%A, %d-%b-%y %H:%M:%S
RFC1123	Mon, 02 Jan 2006 15:04:05 MST	%A, %d-%b-%y %H:%M:%S
RFC3339	2006-01-02T15:04:05Z07:00	%Y-%m-%dT%H:%M:%S
RFC3339Nano	2006-01-02T15:04:05.999999999Z07:00	%Y-%m-%dT%H:%M:%S

3.5.8 ログトピック

 注：

syslog データのトピックは設定できません。

トピック生成モード

Logtail を使用してログを収集したり、API/SDK を使用してデータをアップロードするときに、トピックを設定できます。 現在、次のトピック生成モードがコンソールでサポートされています。**Null -** トピックを生成しない、マシングループトピック属性、およびファイルパスレギュラー。

- **Null -** トピックを生成しない

  コンソールにテキストログを収集するために Logtail を設定するとき、デフォルトのログトピック生成モードは、 **Null -** トピックを生成しないです。つまり、トピックがヌル文字列であるときにトピックを入力せずにログをクエリできます。

- マシングループのトピック属性

  マシングループのトピック属性モードは、異なるサーバーで生成されたログデータを明確に区別するために使用されます。 異なるサーバーのログデータが同じファイルパスとファイル名に格納されている場合、異なるマシングループにマシンを分けて、異なるサーバーのログデータをトピックごとに区別することができます。 これを行うには、マシングループを作成するときに異なるマシングループに異なるトピック属性を設定し、トピック生成モードをマシングルー

プトピック属性に設定します。以前に作成した Logtail 設定をそれらのマシングループに適用して設定を完了します。

このモードを選択すると、データを報告するときに、現在のマシンが属しているマシングループのトピック属性がトピック名として Log Service にアップロードされます。 ログ索引分析機能を使用してログを照会するときは、トピックを指定する必要があります。つまり、ターゲット・マシン・グループのトピック属性を照会条件として指定する必要があります。

- ファイルパスレギュラー

  このモードは、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ます。サービスログがユーザーまたはインスタンスに基づいて異なるディレクトリに格納されているが、サブディレクトリとログファイル名が同じ場合、Log Service はログファイルを収集するときにログを生成するユーザーまたはインスタンスを明確に区別できません。 この場合、トピック生成モードをファイルパスレギュラーに設定し、ファイルパスの正規表現を入力してトピックをインスタンス名として設定することができます。

  このモードを選択すると、データを報告するときに Logtail がトピック名としてインスタンス名を Log Service にアップロードします。 ディレクトリ構造と設定に従って異なるトピックが生成されます。 ログ索引分析機能を使用してログを照会するときは、トピック名をインスタンス名として指定する必要があります。

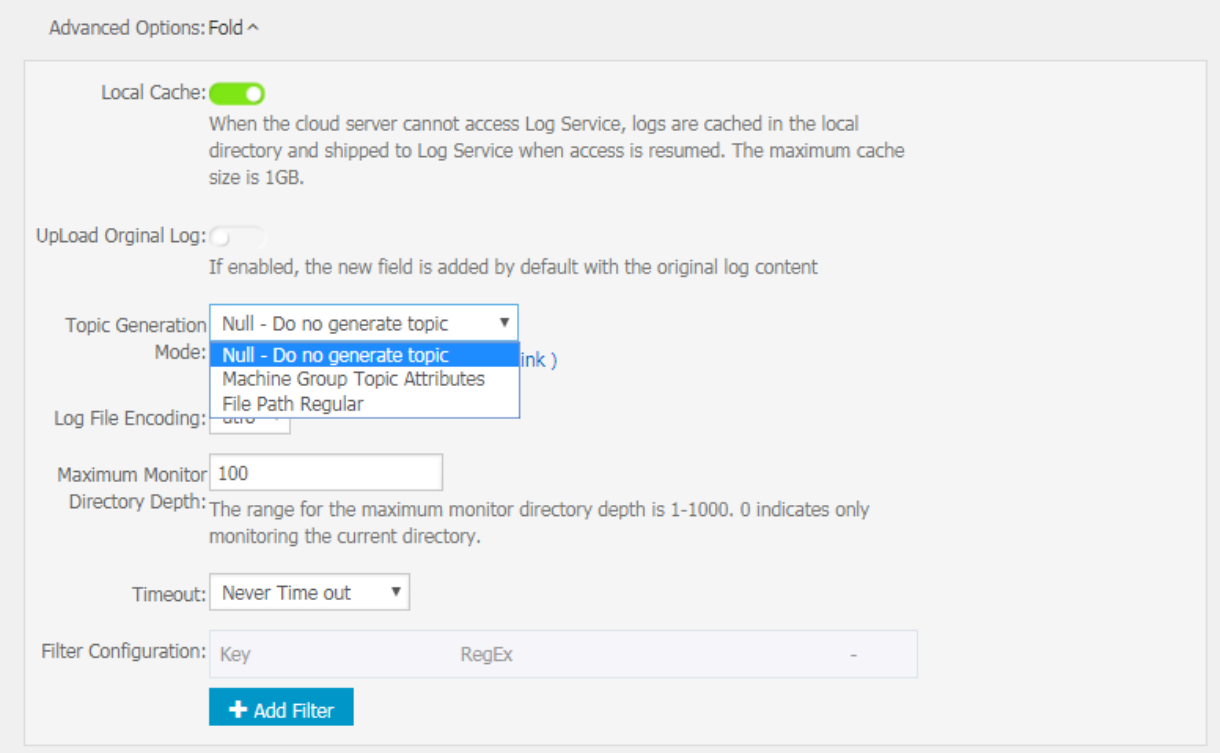
ログトピックの設定

1. テキストファイルの収集に従って、コンソールに Logtail を設定します。

   トピック生成モードをマシングループトピック属性に設定するには、マシングループの作成/変更時にマシングループトピックを設定します。

2. データインポートウィザードの詳細オプションを展開し、トピック生成モードドロップダウンリストからトピック生成モードを選択します。

図 3-26：ログトピックの設定



ログトピックの変更

ログトピックの生成モードを変更するには、データインポートウィザードでトピック生成モードオプションを直接変更します。

注：
変更された設定は、変更が有効になった後に収集されたデータにのみ適用されます。

3.5.9 履歴ログのインポート

Logtail はデフォルトでインクリメンタルログのみを収集します。 履歴ログをインポートする場合は、Logtail の履歴ログインポート機能を使用します。

- Logtail のバージョンは0.16.6以上であること。
- 対象の履歴ログは、収集設定に属していなければならず、Logtail によって収集さていないこと。
- 履歴ログの最終変更時刻は、Logtail 設定時刻よりも前であること。
- ローカル設定の生成とインポートの最大間隔は1分であること。

- ローカル設定をロードする特別なアクションのため、Logtail がお使いのサーバーに LOAD_LOCAL_EVENT_ALARMを送信してこの動作を通知すること。

Logtail は、リスニングをオンにして検出されたイベントに基づいてログを収集します。 Logtail はローカル設定をロードし、ログ収集をトリガーすることもできます。 Logtail は、ローカル設定をロードして履歴ログを収集します。

1. 収集設定情報の作成

   収集を設定し、マシングループに適用します。 対象のログが収集設定に属していることを確認します。 収集の設定についての詳細は、テキストファイルの収集を参照してください。

2. 設定の一意の ID を取得します。

   次の例のように、ローカル /usr/local/ilogtail/user_log_config.jsonから設定の一意のIDを取得します：

```
grep "##" /usr/local/ilogtail/user_log_config.json | awk '{print $1}'
                "##1.0##log-config-test$multi"
                "##1.0##log-config-test$ecs-test"
                "##1.0##log-config-test$metric_system_test"
                "##1.0##log-config-test$redis-status"
```

3. ローカルイベントを追加します。

   ローカルイベントを 次のフォーマットを使用して JSON ファイル /usr/local/ilogtail/local_event.jsonに保存します。

```
[
 {
  "config" : "${your_config_unique_id}",
  "dir" : "${your_log_dir}",
  "name" : "${your_log_file_name}"
 },
 {
  ...
  }
  ...
]
```

   - 構成項目

構成項目	説明	例
Config	ステップ2で取得した構成の一意のID。	##1.0#

構成項目	説明	例
dir	ログが存在するフォルダー。 注： フォルダーが/で終わることはできません。	/data/
name	ログの名前。	access

注：
Logite が無効な JSON ファイルを読み込まないようにするには、ローカルイベントの構成情報を一時ファイルに保存し、一時ファイルを編集した後に/usr/local/ilogtail/local_event.jsonにその内容をコピーすることをおすすめします 。

- 設定の例

```
$ cat /usr/local/ilogtail/local_event.json
[
  {
    "config" : "##1.0##log-config-test$ecs-test",
    "dir" : "/data/log/",
    "name" : "access.log. 2017-08-08"
  },
  {
    "config" : "##1.0##log-config-test$ecs-test",
    "dir" : "/tmp",
    "name" : "access.log. 2017-08-09"
  }
]
```

- **Logtail** が設定をロードしたかどうかを確認するには？

  ローカルファイルlocal_event.jsonを保存すると、Logtailはこのローカル構成ファイルを1分以内にメモリにロードし、local_event.jsonの内容をクリアします。

  Logtailがローカルイベントを読み込んでいるかどうかは、次の方法で確認できます：

  - local_event.json内のコンテンツが消去されているかどうかを確認します。 消去された場合、Logtail はローカル設定情報を読み込みます。
  - /usr/local/ilogtail/ilogtail.LOGファイルに次の情報が含まれているかどうかを確認します。 process local event キーワード。 local_event.jsonの内容が消去されたが、これらのキーワードが見つからない場合、ローカル設定ファイルが無効でフィルタされている可能性があります。
  - #unique_66でLOAD_LOCAL_EVENT_ALARMアラームがあるかどうかをクエリします。

- **Logtail** は設定情報をロードしましたが、それでもデータを収集できません。 この問題をどう処理しますか？

  この問題は、以下の理由により発生する可能性があります：

  - 設定情報が無効です。
  - ローカル設定のconfig項目は存在しません。
  - 対象のログは、収集設定内で指定されたパスにありません。
  - 対象のログは既に収集されています。

- 収集済みのデータをどのように収集できますか？

  収集済みのデータを収集するには、次の手順を実行します：

  1. /etc/init.d/ilogtaild stopコマンドを実行してLogtail を停止します。
  2. /tmp/logtail_check_pointファイルでログのパスを検索します。
  3. このログのチェックポイント（JSONオブジェクト）を削除し、変更を保存します。
  4. 手順3に従って、ローカルイベントを追加します。
  5. /etc/init.d/ilogtaild startコマンドを実行してLogtail を起動します。

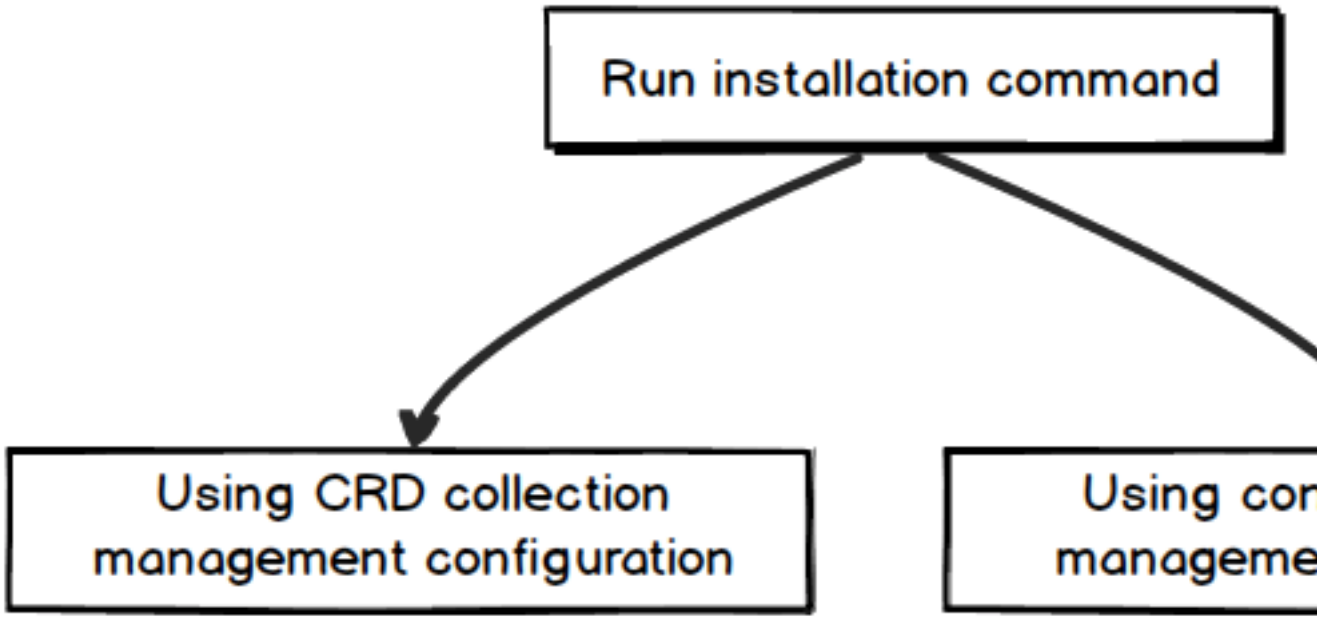
3.6 コンテナログの収集

3.6.1 Kubernetes のログ収集

Log Service は、 Logtail を使用して Kubernetes クラスターログを収集し、カスタムリソース定義 (CRD) を使用して、収集設定を管理します。 ここでは、Logtailのインストール方法、及びLogtailを使用してKubernetesクラスターログを収集する方法について説明します。

設定プロセス

図 3-27：設定プロセス



1. インストールコマンドを実行して、alibaba-log-controller Helm パッケージをインストールします。
2. 収集設定を管理するために、必要に応じて CRD またはコンソールを選択します。

手順 1. インストール

Alibaba Cloud Container Service に Kubernetes をインストールする

インストール手順

1. Alibaba Cloud Container Service Kubernetes のマスターノードにログインします。 ログインの詳細については、「#unique_88」をご参照ください。
2. ${your_k8s_cluster_id} を Kubernetes クラスタ ID に置き換え、次のコマンドを実行します。

```
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}
```

インストール後、 Log ServiceはKubernetesクラスタの同じリージョンにLog Serviceプロジェクトを自動的に作成します。 作成されるプロジェクトの名前はk8s-log-${your_k8s_c

luster_id} です。 プロジェクトに、マシングループk8s-group-${your_k8s_cluster_id}が自動的に作成されます。

> 注：
> Under k8s-log-${your_k8s_cluster_id} プロジェクトに、config-operation-log というという名前の Logstore が自動的に作成されます。 Logstore を削除しないでください。

インストールの例

実行が成功すると、次の情報が出力されます。

```
[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
名前年齢
aliyunlogconfigs.log.alibabacloud.com 0s
==> v1beta1/ClusterRole
alibaba-log-controller 0s
[SUCCESS] install helm package : alibaba-log-controller success.
```

helm Status alibaba-log-controllerを実行して、Pod の現在のステータスを確認できます。 すべてのステータスが成功すると、インストールは成功です。

インストールが正常に完了したら、Log Service コンソールにログインします。 自動的に作成された Log Service プロジェクトが、コンソールに表示されます。 (プロジェクトがたくさんある場合は、キーワードk8s-logで検索してください。)

自分で構築した Kubernetes をインストールする

制限事項

1. Kubernetes クラスタは、バージョン 1.8 以降でなければなりません。
2. Helm 2.6.4 以降がインストールされていなければなりません。

インストール手順

1. Log Service コンソールで、プロジェクトを作成します。 プロジェクト名は、k8s-log-custom-で始める必要があります。
2. 次のコマンドで、パラメーターをご自分のパラメーターに置き換え、コマンドを実行します。

```
wget http://logtail-release.oss-cn-hangzhou.aliyuncs.com/linux64/alicloud-log-k8s-custom-install.sh -O alicloud-log-k8s-custom-install.sh; chmod 744 ./alicloud-log-k8s-custom-install.sh; sh ./alicloud-log-k8s-custom-install.sh {your-project-suffix} {region-id} {aliuid} {access-key-id} {access-key-secret}
```

パラメーターと説明は次のとおりです。

名前	説明
{your-project-suffix}	2 番目の手順で作成したプロジェクト名のk8s-log-custom-の後の部分。 たとえば、作成したプロジェクトがk8s-log-custom-xxxxの場合 、xxxxを入力します。
{regionId}	プロジェクトが配置されているリージョンの ID。 Service endpointを表示できます。たとえば、中国 (杭州) のリージョン ID は、 cn-hangzhou です。
{aliuid}	ユーザー ID は、Alibaba Cloud マスターアカウントのユーザー IDで置き換えてください。 マスターアカウントのユーザー IDはString型で、ID の表示方法については、ユーザー ID 設定のセクション 2.1 をご参照ください。
{access-key-id}	アカウントアクセスキー ID。 サブアカウントアクセスキーを使用し、 権限を与えることを推奨します。#unique_89

名前	説明
{access-key-secret}	アカウントのアクセスキーシークレット。サブアカウントの AccessKey を使用し、AliyunLogFullAccess 権限を与えることを推奨します。 詳細については、#unique_89を参照してください。

インストール後、 Log Service はマシングループをプロジェクトに自動的に作成します。 マシングループ名は、k8s-group-${your_k8s_cluster_id}です。

注：

- Logstoreconfig-operation-logは、k8s-log-${your_k8s_cluster_id} プロジェクトに自動的に作成されます。 この Logstore は削除しないでください。
- 自分で構築した kubernetes のインストールの後、Logtail はprivileged権限を付与します。それは他のPOD を削除中にcontainer text file busyにならないようにするためです。詳細については、bug 1468249、bug 1441737 、およびissue 34538をご参照ください。

インストールの例

実行に成功すると、次のように出力されます。

```
[root@iZbp1dsxxxxxqfbiaZ ~]# wget http://logtail-release.oss-cn-hangzhou.aliyuncs.com/linux64/alicloud-log-k8s-custom-install.sh -O alicloud-log-k8s-custom-install.sh; chmod 744 ./alicloud-log-k8s-custom-install.sh; sh ./alicloud-log-k8s-custom-install.sh xxxx cn-hangzhou 165xxxxxxxx050 LTAxxxxxxxxxxx AIxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxe
....
....
....
NAME：alibaba-log-controller
LAST DEPLOYED: Fri May 18 16:52:38 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-ds 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-ds-7xf2d 0/1 ContainerCreating 0 0s
logtail-ds-9j4bx 0/1 ContainerCreating 0 0s
alibaba-log-controller-796f8496b6-6jxb2 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
[INFO] your k8s is using project : k8s-log-custom-xxx, region : cn-hangzhou, aliuid :
1654218*********, accessKeyId : LTAxxxxxxxxxx
[SUCCESS] install helm package : alibaba-log-controller success.
```

helm status alibaba-log-controllerを使用してPodの現行状態をチェックできます。 すべてのステータスが成功すると、インストールは完了です。

インストール後、Log Service コンソールにログインします。 自動的に作成された Log Service プロジェクトを表示できます。 プロジェクトがたくさんある場合は、キーワードk8s-logで検索してください。

手順 2. 設定

ログ収集は、デフォルトでコンソール設定モードをサポートしています。 また、Kubernetes マイクロサービス開発の CRD 設定モードも提供されています。 kubectl を使用して設定を管理できます。 2 つの設定の比較は次のとおりです。

-	CDRモード	コンソールモード
操作上の複雑さ	低	中
機能	コンソールモードを除く、高度な設定をサポート。	中
複雑さ	中	低
ネットワーク接続	Kubernetes クラスタに接続	インターネットに接続
デプロイメントコンポーネントとの連携	サポート	未サポート
認証方法	Kubernetes 認証	クラウドアカウント認証

Kubernetes のデプロイメント、および公開プロセスと連携されているため、収集設定管理に CRD メソッドを使用することを推奨します。

コンソール上でコレクションの設定を管理する

必要に応じてコンソールにLogtail収集設定を作成します。 設定手順については、以下をご参照ください。

- コンテナテキストログ（推奨）
- コンテナスタンダードアウトプット（推奨）

- ホストテキストファイル

  デフォルトでは、ホストのルートディレクトリはLogtail容器の/logtail_hostディレクトリにマウントされています。 パスを設定する際、このプレフィックスを追加する必要があります。たとえば、ホストの /home/logs/app_log/ディレクトリからデータを収集するには、構成ページのログパスを/logtail_host/home/logs/app_log/に設定します。
- #unique_45

CRD 管理による取得設定

kubernetes マイクロサービス開発モデルでは、ログサービスも CRD を設定する方法を提供し、kubectl を直接使用して設定を管理できます。また、kubernetes デプロイメントと連携し、公開プロセスをより完全に行うことができます。

詳細については、CRD での Kubernetes ログ収集の設定を参照してください。

その他の操作

Glasonset デプロイメントの移行手順

以前に使用した WebSphere の set メソッドを使用して Log Service をデプロイした場合、設定管理に CRD を使用することはできません。 次の方法で新しいバージョンに移行できます。

> 注：
> アップグレード中に、いくつかのログが複製されます。 CRD 管理設定は、CRD を使用して作成された設定に対してのみ使用できます。 履歴設定は、 CRD モードを使用して作成されていないため、CRD 管理モードはサポートしていません。

1. 新しいバージョンの形式でインストールすると、インストールコマンドには、以前のkubernetes クラスタで使用されていた Log Service プロジェクト名のパラメーターが最後に追加されます。

   たとえば、プロジェクト名がk8s-log-demoで、クラスタ ID が c12ba2028cxxxxxxxxxx6939f0bの場合、インストールコマンドは次のようになります。

   ```
   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 c12ba2028cxxxxxxxxxx6939f0b k8s-log-demo.
   ```
2. インストールが成功したら、Log Service コンソールで、履歴収集設定を新しいマシングループk8s-group-${your_k8s_cluster_id}に適用します。
3. 1 分後、履歴収集設定はマシングループの履歴にバインドされます。

4. ログ収集が正常であれば、以前にインストールした Logtail DaemonSet を削除できます。

同じ Log Service プロジェクトで複数のクラスタを使用する

複数のクラスタを使用して、同じ Log Service プロジェクトにログを収集することができます。他のクラスタ Log Service コンポーネントをインストールする際、インストールパラメーターの${your_k8s_cluster_id}を、最初にインストールしたクラスタ ID に置き換える必要があります。

たとえば、 ID が abc001、abc002、および abc003 の 3 つのクラスタがあるとします。 3 つのクラスタのインストールパラメーター${your_k8s_cluster_id}は、すべてabc001にする必要があります。

 注：
リージョン間の Kubernetes マルチクラスタ共有では、この方法はサポートしていません。

Logtail コンテナログ

Logtail ログは、Logtail コンテナの/usr/local/ilogtail/ディレクトリに格納されており、ファイル名は、ilogtail.LOGおよびilogtail.pluginです。コンテナの stdout に参照の意味はありません。そのため、次の stdout の出力を無視できます。

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fb
e2bdb95d13b1e110172ef57fe840c82155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009
528daa749c1bf8c16edff44beab6e69718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fa
c800cd56c6e880dc4e8a640b1e16c22dbe/merged: must be superuser to unmount
......
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
ilogtail is running
```

Kubernetes クラスタ内のログ関連コンポーネントのステータスを表示する

```
helm status alibaba-log-controller
```

alibaba-log-controller の起動に失敗した場合は？

次のようにインストールを実行してください。

1. インストールコマンドは、kubernetes クラスタのマスターノードで実行されます
2. インストールコマンドパラメーターは、クラスタ ID に入力されます。

これらの問題によりインストールが失敗した場合は、 helm del —purge alibaba-log-controller rを使用して、インストールパッケージを削除し、再度インストールを実行してください。

インストールの失敗が続く場合は、チケットを起票し、サポートセンターへお問い合わせください。

Kubernetes クラスタの Logtail DaemonSet のステータスを確認する

kubectl get ds -n kube-systemコマンドを実行し、Logtail の実行状態を確認できます。

 注：
Logtail のデフォルトの namespace は、kube-systemです。

Logtail のリソース制限を調整する方法

デフォルトでは、Logtail は最大 40% の CPU と 200M の RAM しか占有できません。 処理速度を上げる必要がある場合は、次の 2 つのセクションでパラメーターを調整する必要があります。

- YAMLテンプレートのresourcesのlimitsとrequests。
- Logtail 起動設定ファイルのパスは、YAML テンプレートのALIYUN_LOGTAIL_CONFIG環境変数です。 変更方法については、「Logtail 起動設定パラメーター」 をご参照ください。

Logtail DaemonSet の強制更新

logtail-daemonset.yamlファイルを修正後、次のコマンドを実行して、Logtail DaemonSet を強制的に更新します。

```
kubectl --namespace=kube-system delete ds logtail
kubectl apply -f ./logtail-daemonset.yaml
```

 注：
強制更新中にデータの重複が発生することがあります。

Logtail DaemonSet の設定情報を確認する

```
'kubectl describe ds logtail -n kube-system'
```

Logtail のバージョン番号、IP、開始時刻などを確認する

例は次のとおりです。

```
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl get po -n kube-system -l k8s-app=logtail
NAME READY STATUS RESTARTS AGE
logtail-gb92k 1/1 Running 0 2h
logtail-wm7lw 1/1 Running 0 4d
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-gb92k -n kube-system cat /usr/
local/ilogtail/app_info.json
{
```

```
  "UUID" : "",
  "hostname" : "logtail-gb92k",
  "instance_id" : "*********************************",
  "ip" : "*.*.*.*",
  "logtail_version" : "0.16.2",
  "os" : "Linux; 3.10.0-693.2.2.el7.x86_64; #1 SMP Tue Sep 12 22:26:13 UTC 2017; x86_64
",
  "update_time" : "2018-02-05 06:09:01"
}
```

Logtail の実行ログを表示する

Logtail 実行ログは、/usr/local/ilogtail/ディレクトリに格納されます。 ファイル名は、ilogtail.LOGです。 ファイルは圧縮され、ilogtail.LOG.x.gzとして保存されます。

例は次のとおりです。

```
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-gb92k -n kube-system tail /usr/
local/ilogtail/ilogtail.LOG
[2018-02-05 06:09:02.168693] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:
104] logtail plugin Resume:start
[2018-02-05 06:09:02.168807] [INFO] [9] [build/release64/sls/ilogtail/LogtailPlugin.cpp:
106] logtail plugin Resume:success
[2018-02-05 06:09:02.168822] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.
cpp:369] start add existed check point events, size:0
[2018-02-05 06:09:02.168827] [INFO] [9] [build/release64/sls/ilogtail/EventDispatcher.
cpp:511] add existed check point events, size:0 cache size:0 event size:0 success count:0
```

pod の Logtail を再起動する

例は次のとおりです。

```
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-gb92k -n kube-system /etc/init.
d/ilogtaild stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 9
stop success
[root@iZbp1dsu6v77zfb40qfbiaZ ~]# kubectl exec logtail-gb92k -n kube-system /etc/init.
d/ilogtaild start
ilogtail is running
```

3.6.2 コンテナテキストログ

Logtail は、コンテナで生成されたテキストログを収集し、収集されたログをコンテナメタデータと一緒に Log Service にアップロードします。

機能特徴

基本的なログファイル収集と比較して、Docker ファイル収集は、次の機能もサポートしています。

- コンテナ内のログパスの設定。このパスからホストへのマッピングを気にする必要はありません。
- Label を使用した収集するコンテナの指定。
- Label を使用した特定のコンテナの除外。
- 収集対象のコンテナを指定する environment。
- 特定のコンテナを除外する environment。
- 複数行ログ (Java スタックログなど) 。
- コンテナデータの自動タグ付け。
- Kubernetes コンテナの自動タグ付け。

制限事項

- 収集停止のポリシー：コンテナが停止すると、Logtail は (1-3 秒の遅延で) コンテナの die イベントをリッスンした後、コンテナからのログの収集を停止します。 この間に収集遅延が発生すると、停止前にログの一部を失う可能性があります。
- **Docker**ストレージドライブ ：現在、オーバーレイドライブと overlay2 ドライブのみがサポートされています。 他のドライブタイプの場合、ローカル PC にログディレクトリをマウントする必要があります。
- **Logtail**の実行方法：Logtail はコンテナとして実行し、Logtail のデプロイ方法に従う必要があります。
- **Label**：ラベルは、Docker inspect のラベル情報であり、Kubernetes 設定のラベルではありません。
- **Environment**：Environment は、コンテナの起動時に設定された環境情報です。

手順

1. Logtail コンテナをデプロイして設定します。
2. Log Service で収集設定を行います。

1. Logtail のデプロイメントと設定

- **Kubernetes**

  Kubernetes ログ収集の詳細については、「Kubernetes のログ収集」をご参照ください 。
- その他のコンテナ管理メソッド

  Swarm や Mesos などの他のコンテナ管理メソッドの詳細については、「標準の Docker ログの収集」をご参照ください。

2. Log Service の収集設定

1. **Logstore** リスト ページで、**[**データインポートウィザード**]** アイコンをクリックして設定プロセスを入力します。
2. データソースを選択します。

   [サードパーティ製のソフトウェア**]** で **[Docker** ファイル**]** を選択し、[次へ] をクリックします。
3. データソースを設定します。

設定項目	必須	説明
Docker ファイル	必須	収集する該当ファイルが Docker ファイルであるかどうかを確認します。
ホワイトリストにラベルを付ける	オプション	LabelKey は必須です。 LabelValue が空でない場合、LabelKey = LabelValue がラベルに含まれているコンテナのみが収集されます。 LabelValue が空の場合、LabelKey がラベルに含まれているすべてのコンテナが収集されます。 注： a. キーと値のペアには互いに OR 関係があります。つまり、ラベルにキーと値のペアのいずれかが含まれていればコンテナは収集されます。 b. ここでは、ラベルは Docker inspect のラベル情報です。
ブラックリストに Label を付ける	オプション	LabelKey は必須です。 LabelValue が空でない場合、LabelKey = LabelValue がラベルに含まれているコンテナのみが除外されます。 LabelValue が空の場合、LabelKey がラベルに含まれているすべてのコンテナが除外されます。 注： a. キーと値のペアは相互に OR 関係を持ちます。つまり、 ラベルにキーと値のペアのいずれかが含まれている場合、コンテナは除外されます。 b. ここでは、ラベルは Docker inspect のラベル情報です。

設定項目	必須	説明
環境変数ホワイトリスト	オプション	EnvKey は必須です。 EnvValue が空でない場合、EnvKey = EnvValue が環境変数に含まれているコンテナのみが収集されます。 EnvValue が空の場合、EnvKey が環境変数に含まれているすべてのコンテナが収集されます。  注： • キーと値のペアは相互に OR 関係を持ちます。つまり、環境変数にキーと値のペアのいずれかが含まれている場合、コンテナは収集されます。 • ここでは、環境変数とは、コンテナの起動時に設定される環境情報です。
環境変数ブラックリスト	オプション	EnvKeyis は必須です。 EnvValue が空でない場合、EnvKey = EnvValue が環境変数に含まれているコンテナのみが除外されます。 EnvValue が空の場合、EnvKey が環境変数に含まれているすべてのコンテナが除外されます。  注： a. キーと値のペアは相互に OR 関係を持ちます。つまり、環境変数にキーと値のペアのいずれかが含まれている場合、コンテナは収集されます。 b. ここでは、環境変数とは、コンテナの起動時に設定される環境情報です。
その他の設定	-	その他の収集設定とパラメーターの説明については、「テキストファイルの収集」をご参照ください。

4. 説明

   • このトピックでは、ラベルは Docker 検査に含まれるラベル情報を指します。

   • Kubernetes の名前空間とコンテナ名は、Docker 内のラベル io.kubernetes.pod.namespace および io.kubernetes.container.name に割り当てられます。 たとえば、作成した Pod は backend-prod 名前空間に属し、コンテナ名は worker-server とします。この場合、2 つのホワイトリストラベル io.kubernetes.pod.namespace : backend-pro

および io.kubernetes.container.name : worker-server を設定して、worker-server コンテナ内のログのみの収集を 指定ことができます。

- Kubernetes ではio.kubernetes.pod.namespace と io.kubernetes.container.name ラベルのみの使用を推奨します。 他のシナリオでは、環境のホワイトリストまたはブラックリストを使用できます。

5. マシングループに適用します。

   [マシングループに適用] ページで、収集対象の Logtail マシングループを選択し、**[マシングループに適用]** をクリックして、選択したマシングループに設定を適用します。 マシングループを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. コンテナのテキストログにアクセスするプロセスを完了します。

   検索、分析、可視化、および Shipper と ETL 機能を設定するために、ページの指示どおりに設定を完了します。

設定例

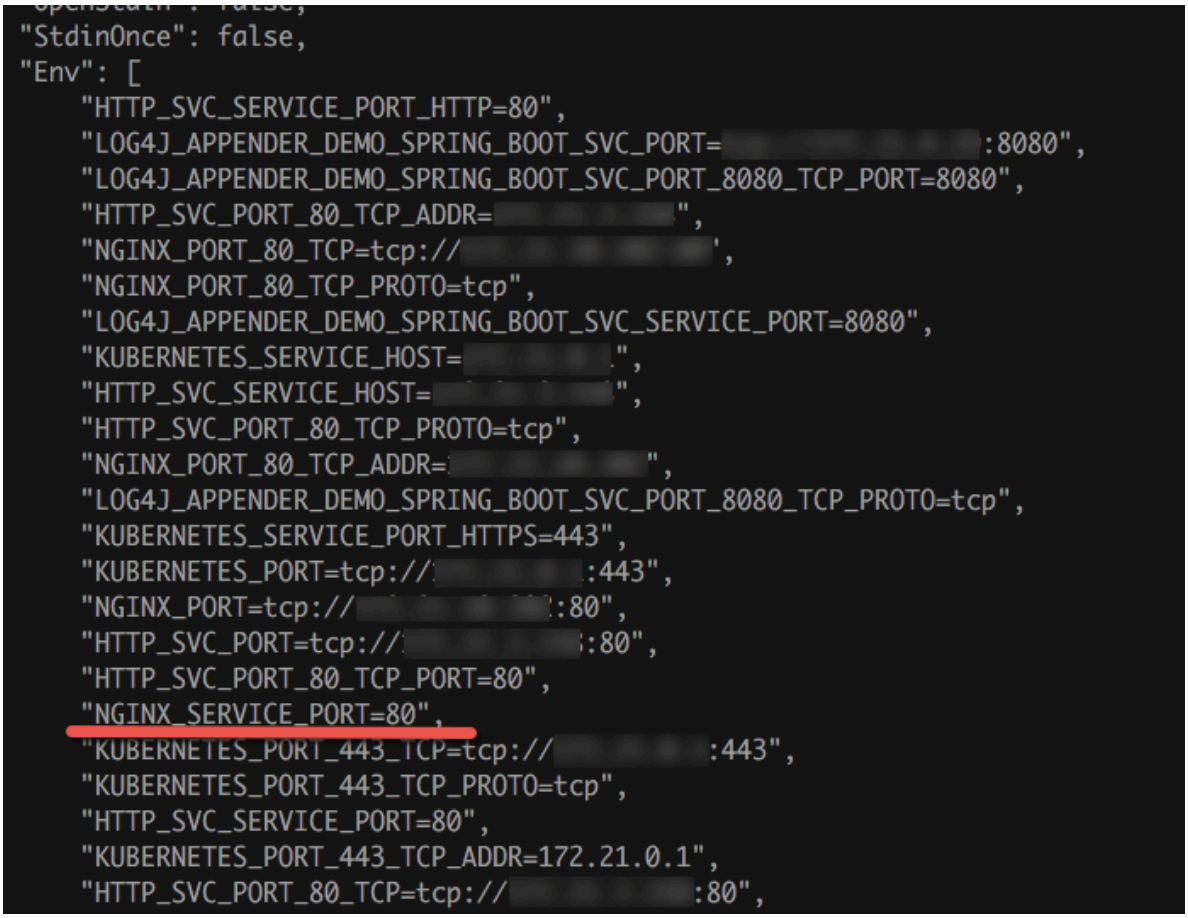
- **Environment** 設定

  environment が POD_NAMESPACE=kube-system ではなく、 NGINX_PORT_80_TCP_PORT=80 であるコンテナのログを収集します。 ログファイルパスは、/var/log/nginx/access.log で、ログはシンプルモードで解析されます。

   **注 :**

Environmentは、コンテナの起動時に設定された環境情報です。

図 3-28：Environment 設定の例



この例のデータソースの設定は、次のとおりです。 その他の収集設定とパラメーターの説明については、「テキストファイルの収集」をご参照ください。

- **Label** 設定

  Label が io.kubernetes.container.name=nginx で、type=pre ではないコンテナのログを収集します。 ログファイルパスは、/var/log/nginx/access.logで、ログはシンプルモードで解析されます。

  注：

Label は、Docker inspect のラベル情報であり、Kubernetes 設定のラベルではありません。

図 3-29：Label モードの例

この例のデータソースの設定は、次のとおりです。 その他の収集設定とパラメーターの説明については、「テキストファイルの収集」をご参照ください。

デフォルトフィールド

標準 Docker

各ログによって次のフィールドが、デフォルトでアップロードされます。

フィールド	説明
_image_name_	Image name.
_container_name_	コンテナ名
_container_ip_	コンテナ IP アドレス

Kubernetes

クラスターが Kubernetes クラスターの場合、各ログによって次のフィールドがデフォルトでアップロードされます。

フィールド	説明
_image_name_	イメージ名
_container_name_	コンテナ名
_pod_name_	pod 名
namespace	pod が存在する名前空間
_pod_uid_	pod の一意の識別子
_container_ip_	pod の IP アドレス

3.6.3 コンテナ標準出力

Logtail では、コンテナの標準出力ストリームを入力ソースとして使用し、標準出力ストリームをコンテナメタデータと一緒に Log Service にアップロードすることができます。

機能の特徴

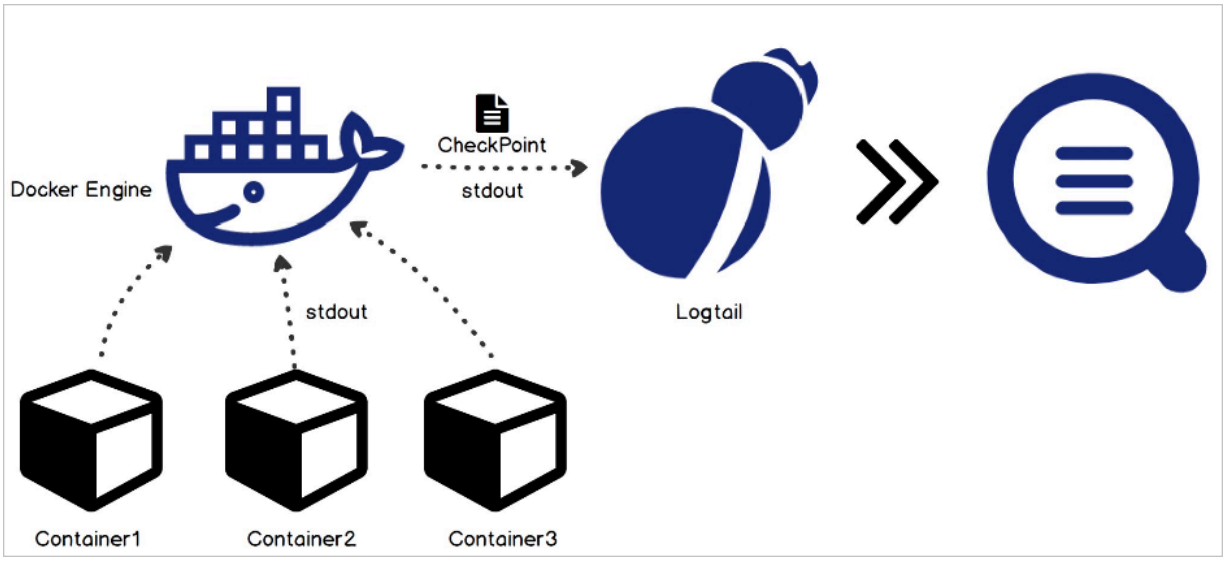
- stdout と stderr の収集をサポートします。
- ラベルを使用して収集するコンテナを指定することをサポートします。
- ラベルを使用して特定のコンテナを除外することをサポートします。
- environments を使用して収集するコンテナを指定することをサポートします。
- environments を使用して収集するコンテナを除外することをサポートします。
- 複数行ログ（Java スタックログなど）をサポートします。
- コンテナデータの自動タグ付けをサポートします。
- Kubernetes コンテナの自動タグ付けをサポートします。

実装の原則

次の図に示すように、Logtail は Docker の Domain Socket と通信し、 Docker 上で実行されているすべてのコンテナを照会し、ラベル情報に従って収集するコンテナを特定します。 Logtail は、Docker log コマンドを使用して、指定されたコンテナログを取得します。

Logtail は、コンテナの標準出力を収集するときに定期的に収集したポイント情報をチェックポイントファイルに保存します。 Logtail が停止後に再起動されると、ログは最後に保存されたポイントから収集されます。

図 3-30：実装の原則



制限

- 現在、この機能は Linux のみをサポートしており、Logtail 0.16.0 以降のバージョンに依存しています。 バージョンの確認とアップグレードについては、Logtail の Linux へのインストールをご参照ください。
- デフォルトでは、Logtail は/var/run/docker.sockを使用して Docker Engine にアクセスします。 ドメインソケットが存在し、アクセス権があることを確認します。
- 複数行ログの制限。 複数の行で構成されるログが出力遅延のために分割されないようにするため、最後に収集された複数行のログはデフォルトで短時間キャッシュされます。 デフォルトのキャッシュ時間は 3 秒ですが、BeginLineTimeoutMsパラメータを使用して変更できます。 ただし、この値は 1000 未満にすることはできません。 そうしないと、エラーが発生する可能性があります。
- 収集を停止するための戦略。 コンテナが停止すると、Logtail はコンテナの die イベントをリスニング後、コンテナからの標準出力の収集を停止します。 この間に収集遅延が発生すると、停止前に出力の一部を失う可能性があります。
- **Context limit**。 デフォルトでは、コレクション構成は同じコンテキストにあります。 コンテナの各タイプごとに異なるコンテキストを設定するには、各タイプの収集設定を作成します。
- データ処理。収集されたデータのデフォルトのフィールドは content で、これは共通の処理設定をサポートします。
- **Label**。 Label は、Docker 検査のラベル情報であり、Kubernetes 構成のラベルではありません。
- **Environment**。Environment は、コンテナの起動時に構成された環境情報です。

構成プロセス

1. Logtail コンテナをデプロイして構成します。
2. ログサービスで収集の構成を行います。

1.Logtail コンテナをデプロイして構成する

- **Kubernetes**
- その他コンテナ管理方法

2.ログサービスで収集の構成を行う

1. **Logstore List** ページで、**Data Import Wizard** アイコンをクリックし、設定プロセスに入ります。
2. データソースを選択します。

   サードパーティ製ソフトウェアの **Docker Stdout** を選択してから、**Next** をクリックします。

3. データソースを設定します。

   データソースの設定ページで、コレクションの設定を完了します。 次の例をご参照ください。

```
{
 "inputs": [
    {
       "type": "service_docker_stdout",
       "detail": {
          "Stdout": true,
          "Stderr": true,
          "IncludeLabel": {
             "io.kubernetes.container.name": "nginx"
          },
          "ExcludeLabel": {
             "io.kubernetes.container.name": "nginx-ingress-controller"
          },
          "IncludeEnv": {
             "NGINX_SERVICE_PORT": "80"
          },
          "ExcludeEnv": {
             "POD_NAMESPACE": "kube-system"
          }
       }
    }
 ]
}
```

4. マシングループに適用します。

   マシングループに適用ページで、収集する Logtail マシングループの作成 を選択し、マシングループに適用をクリックして、選択したマシングループに設定を適用します。マシングループを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。

構成項目の説明

入力ソースタイプは `service_docker_stdout` です。

設定項目	型	必須	説明
IncludeLabel	マッピングタイプ。キーと値の両方がStringです。	はい	デフォルトでは空です。 空の場合、すべてのコンテナが収集されます。 キーが空ではなく値が空の場合、ラベルにこのキーが含まれているすべてのコンテナが収集されます。  注： 1. キーと値のペアには OR 関係があります。つまり、ラベルにキーと値のペアが含まれていればコンテナが収集されます。 2. ここで、ラベルは Docker inspect のラベル情報です。
ExcludeLabel	マッピングタイプ。キーと値の両方が String です。	いいえ	デフォルトでは空です。 空の場合、コンテナは除外されません。 キーが空ではなく値が空の場合、ラベルにこのキーが含まれるすべてのコンテナが除外されます。  注： 1. キーと値のペアには互いに OR 関係があります。 つまり、ラベルにキーと値のペアが含まれている場合、コンテナは除外されます。 2. ここで、ラベルは Docker inspect のラベル情報です。

設定項目	型	必須	説明
IncludeEnv	マッピングタイプ。キーと値の両方が String です。	いいえ	デフォルトでは空です 空の場合、すべてのコンテナが収集されます。キーが空ではなく値が空の場合、このキーを含む環境を持つすべてのコンテナが収集されます。  注： 1. キーと値のペアは相互に OR 関係を持ちます。つまり、環境にキー/値のペアが含まれている場合、コンテナが収集されます。 2. 環境は、コンテナの起動時に構成された環境情報です。
ExcludeEnv	マッピングタイプ。キーと値の両方が String です。	いいえ	デフォルトでは空です 空の場合、コンテナは除外されません。 キーが空ではなく値が空の場合、このキーを含む環境を持つすべてのコンテナが除外されます。  注： 1. キーと値のペアには OR 関係があります。つまり、環境にキーと値のペアが含まれている場合、コンテナは除外されます。 2. 環境は、コンテナの起動時に構成された環境情報です。
Stdout	ブール	いいえ	デフォルトでは True です。 false の場合、stdout データは収集されません。
Stderr	ブール	いいえ	デフォルトでは True です。 false の場合、stderr データは収集されません。
BeginLineRegex	String	いいえ	デフォルトでは空です。 空でない場合は、行の先頭に一致する正規表現です。 この正規表現が行と一致する場合、その行は新しいログとして扱われます。 それ以外の場合、データの行は前のログに接続されます。

設定項目	型	必須	説明
BeginLineTimeoutMs	int	いいえ	行の先頭と一致するためのタイムアウト（ミリ秒単位）。デフォルト値は 3000 です。 新しいログが 3 秒以内に表示されない場合は、最後のログが出力されます。
BeginLineCheckLength	int	いいえ	正規表現との一致に使用される行の先頭の長さ（バイト単位）。 デフォルト値は 10 \ 1024 です。 正規表現が最初の N バイト以内の行と一致する場合は、このパラメータを設定して一致効率を上げます。
MaxLogSize	int	いいえ	ログの最大長（バイト単位）。 デフォルト値は 512 \ 1024 です。 ログの長さが設定された値を超えると、一致した行の先頭を検索することなく、ログが直接アップロードされます。

デフォルトフィールド

ノーマルドッカー

次のフィールドは、デフォルトで各ログによってアップロードされます。

フィールド名	説明
time	データ時間。 たとえば、2018-02-02T02：18：41.979147844Z となります。
source	入力ソースタイプ（stdout または stderr）。
_image_name_	イメージ名。
_container_name_	コンテナ名。

Kubernetes

クラスタが Kubernetes クラスタの場合、デフォルトで各ログによって次のフィールドがアップロードされます。

フィールド名	説明
time	データ時間。 たとえば、2018-02-02T02：18：41.979147844Z となります。

フィールド名	説明
source	入力ソース・タイプ（stdout または stderr のいずれか）。
_image_name_	イメージ名。
_container_name_	コンテナ名。
_pod_name_	ポッド名。
namespace	ポッドが存在する名前空間。
_pod_uid_	ポッドの一意の識別子。

設定例

一般的な設定

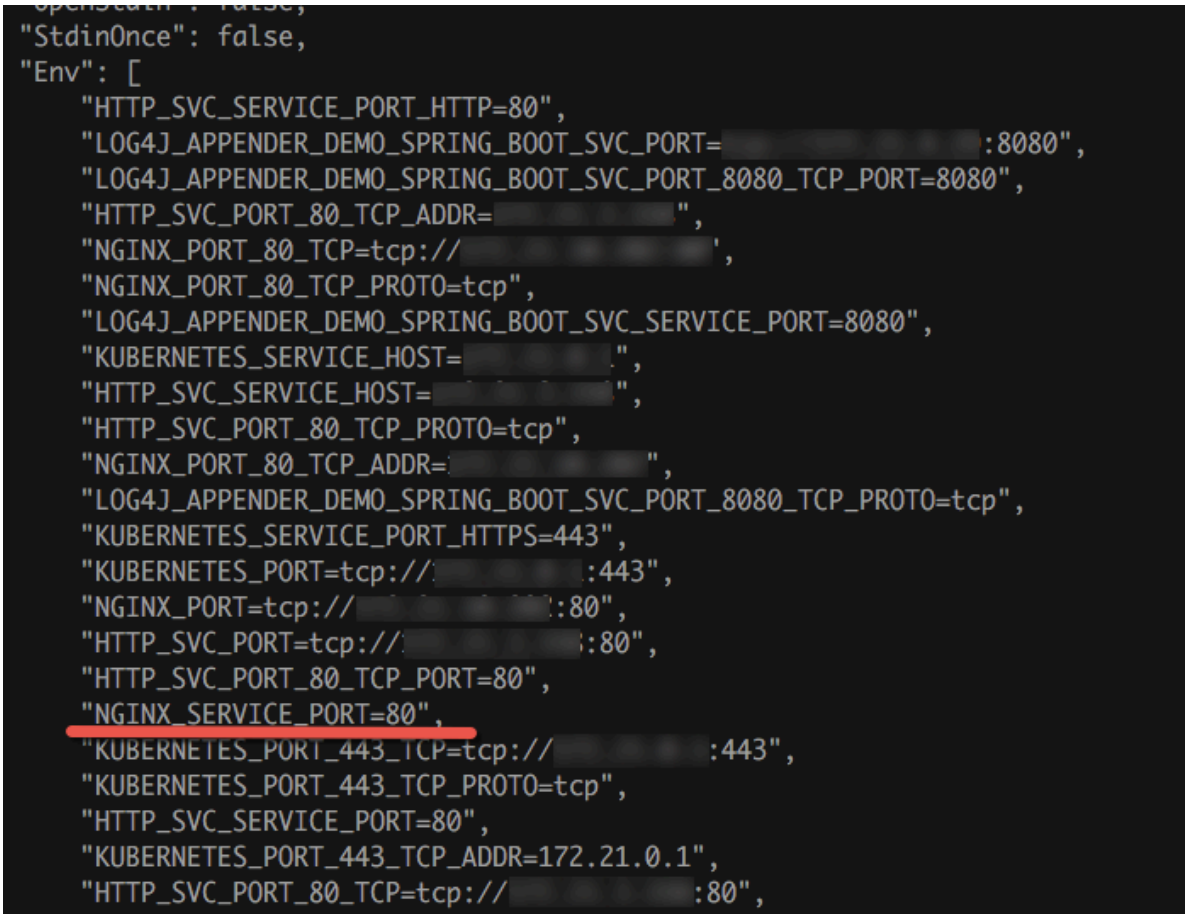
- **Environment** の構成

  Environment が NGINX_PORT_80_TCP_PORT = 80 で、POD_NAMESPACE = kube-system ではないコンテナの stdout ログと stderr ログを収集する：

   注：

Environment は、コンテナの起動時に構成された環境情報です。

図 3-31：Environment 構成の例



収集の構成

```
{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "IncludeEnv": {
          "NGINX_PORT_80_TCP_PORT": "80"
        },
        "ExcludeEnv": {
          "POD_NAMESPACE": "kube-system"
        }
      }
    }
  ]
```

```
}
```

- **Label** 構成

  Label が io.kubernetes.container.name=nginx で type=pre ではないコンテナの stdout ログと stderr ログを収集します。

  > 注：
  > こちらの label は Docker です Kubernetes 構成のラベルではありません。

  図 3-32：Label 構成の例


```
{
  "inputs": [
    {
      "type": "service_docker_stdout",
      "detail": {
        "Stdout": true,
        "Stderr": true,
        "IncludeLabel": {
          "io.kubernetes.container.name": "nginx"
        },
        "ExcludeLabel": {
          "type": "pre"
        }
      }
    }
  ]
}
```

複数行ログの収集設定

マルチラインログ収集は、Java 例外スタック出力の収集にとって特に重要です。 ここでは、Java 標準出力ログの標準収集設定を紹介します。

- ログサンプル：

```
2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.
controller.DemoController : service start
```

```
2018-02-03 14:18:41.969 ERROR [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.
controller.DemoController : java.lang.NullPointerException
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(Applicatio
nFilterChain.java:193)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.
java:166)
at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.
java:199)
at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java
:96)
...
2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web.
controller.DemoController : service start done
```

- コレクションの構成：

  ラベルが app = monitor で、行の先頭が日付型のコンテナの入力ログを収集する（マッチング効率を上げるために、行の最初の 10 バイトだけが正規表現との一致をチェックするために使用される）。

```
{
"inputs": [
 {
  "detail": {
   "BeginLineCheckLength": 10,
   "BeginLineRegex": "\\d+-\\d+-\\d+. *",
   "IncludeLabel": {
    "app": "monitor"
   }
  },
  "type": "service_docker_stdout"
 }
]
}
```

収集したデータを処理する

Logtail は収集された Docker の標準出力に対して common data processing methods をサポートしています。 前のセクションの複数行のログ形式に基づいて、正規表現を使用して時刻、モジュール、スレッド、クラス、および情報のログを解析することをお勧めします。

- 収集の構成：

  ラベルが app = monitor で、行の先頭が日付型のコンテナの入力ログを収集する（マッチング効率を上げるために、行の最初の 10 バイトだけが正規表現との一致をチェックするために使用される）

```
{
"inputs": [
 {
  "detail": {
   "BeginLineCheckLength": 10,
   "BeginLineRegex": "\\d+-\\d+-\\d+. *",
   "IncludeLabel": {
    "app": "monitor"
   }
```

```
    },
    "type": "service_docker_stdout"
  }
],
"Processors ":[
  {
    "type": "processor_regex",
    "detail": {
      "SourceKey": "content",
      "Regex": "(\\d+-\\d+-\\d+ \\d+:\\d+:\\d+\\.\\d+)\\s+(\\w+)\\s+\\[([^]]+)]\\s+
\\[([^]]+)]\\s+:\\s+(. *)",
      "Keys": [
        "time",
        "module",
        "thread",
        "class",
        "info"
      ],
      "NoKeyError": true,
      "NoMatchError": true,
      "KeepSource": false
    }
  }
]
}
```

- サンプル出力：

  ログ 2018-02-03 14:18:41.968 INFO [spring-cloud-monitor] [nio-8080-exec-4] c.g.s.web .controller.DemoController：service start done を処理した後の出力は、次のようになります。

```
__tag__:__hostname__:logtail-dfgef
_container_name_:monitor
_image_name_:registry.cn-hangzhou.aliyuncs.xxxxxxxxxxxxxxx
_namespace_:default
_pod_name_:monitor-6f54bd5d74-rtzc7
_pod_uid_:7f012b72-04c7-11e8-84aa-00163f00c369
_source_:stdout
_time_:2018-02-02T14:18:41.979147844Z
Time: 2018-02-02 02:18:41. 968
level:INFO
module:spring-cloud-monitor
Thread: fig
Class: c.g.s. web. Controller. demcontroller
message:service start done
```

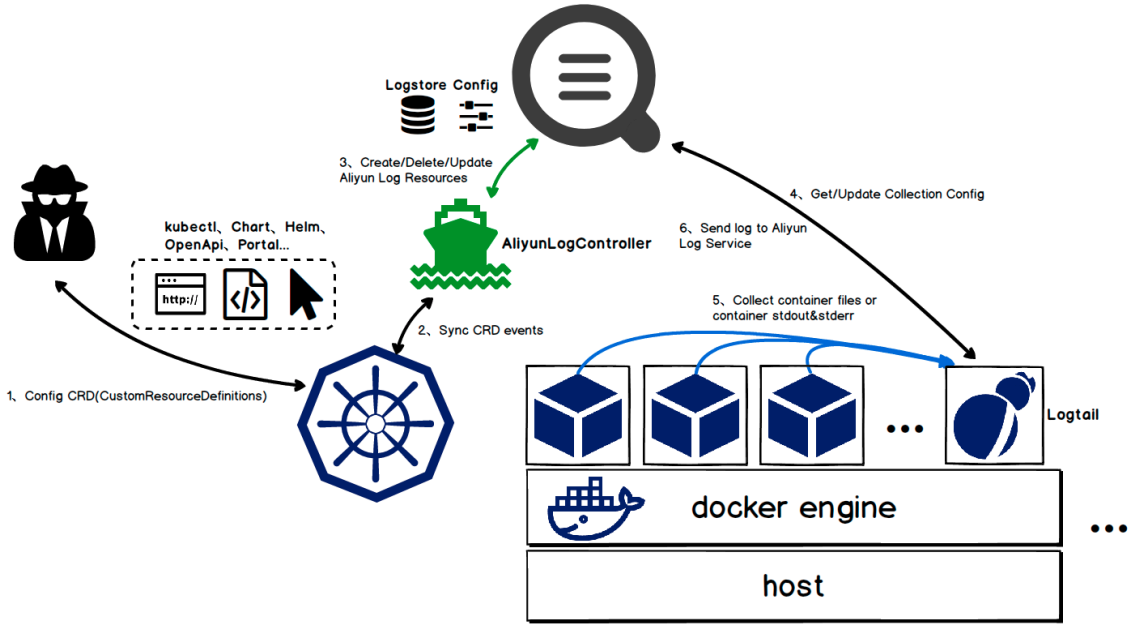
3.6.4 CRD での Kubernetes ログ収集の設定

ログ収集は、デフォルトでコンソールに設定されます。 Log Service では、Kubernetes マイクロサービス開発用のログ収集の CRD 設定も提供します。 これにより、kubectl を使用して設定を管理できます。

この方法は、Kubernetes のデプロイメント、および公開プロセスと連携されているため、収集設定管理に CRD メソッドを使用することを推奨します。

実装の原則

図 3-33：実装の原則



インストールコマンドを実行して、 alibaba-log-controllerHelm パッケージをインストールします。 Helm パッケージは、主に次の操作を実行します。

1. aliyunlogconfigs CRD (カスタムリソース定義) を作成します。
2. alibaba-log-controller をデプロイします。
3. Logtail DaemonSet をデプロイします。

設定の内部ワークフローは次のとおりです。

1. kubectl または他のツールを使用して、aliyunlogconfigs CRD 設定を適用します。
2. alibaba-log-controller は、設定のアップデートを検出します。
3. alibaba-log-controller は、CRD の内容とサーバーのステータスに基づき、Logstore の作成、設定の作成、およびマシングループのアプリケーション設定のリクエストを自動的に送信します。
4. DaemonSet モードで実行される Logtail は、サーバー設定のリクエストを定期的に送信し、新しい設定または更新された設定を取得し、高速ロードを実行します。
5. Logtail は、設定情報に基づき、各コンテナ (pod) から標準出力またはファイルを収集します。
6. Logtail は、処理され集められたデータを Log Service に送信します。

設定方法

> 注：
> DaemonSet モードでデプロイされた Logtail を使用した場合、CRD モードでは設定を管理できません。 詳細は、このドキュメントの **DaemonSet** デプロイメントモードの移行プロセスをご参照ください。

設定を作成するには、AliyunLogConfig の CRD を定義する必要があります。設定を削除するには、対応する CRD リソースを削除する必要があります。 CRD は次のように設定されています。

```
apiVersion: log.alibabacloud.com/v1alpha1 ## デフォルト値。変更する必要はありません。
kind: AliyunLogConfig ## デフォルト値。変更する必要はありません。
metadata:
  name: simple-stdout-example ## リソース名。ラスター内で一意である必要があります。
spec:
  logstore: k8s-stdout ## Logstore 名。存在しなければ自動的に作成されます。
  shardCount: 2 ## [オプション] Logstore のパーティション数。 デフォルト値は 2 です。 値の範囲は 1～10 です。
  lifeCycle: 90 ## [オプション] Logstore の保存期間。 デフォルト値は 90 です。 値の範囲は 1～7300 です。 値 7300 は、永久的な保存を示しています。
  logtailConfig: ## 詳細な設定
    inputType: plugin ## 収集の入力タイプ。 通常、値はファイルかプラグインです。
    configName: simple-stdout-example ## 収集設定名。 値はリソース名 (metadata.name) と同じでなければなりません。
    inputDetail: ## 詳細な設定情報。例をご参照ください。
      ...
```

設定が完了して適用されると、alibaba-log-controller が自動的に作成されます。

設定の表示

Kubernetes CRD またはコンソールで設定を表示できます。

コンソールでの設定の表示方法については、「Logtail 設定の作成」をご参照ください。

> 注：
> CRD 方式を使用して設定を管理すると、CRD で設定を更新する際、コンソールで行った設定変更が上書きされます。

- すべての設定を表示するには、 `kubectl get aliyunlogconfigs` を実行します。

```
[root@iZbp1dsbiaZ ~]# kubectl get aliyunlogconfigs
NAME AGE
regex-file-example 10s
regex-stdout-example 4h
```

```
simple-file-example 5s
```

- 詳細な設定とステータスを表示するには、kubectl get aliyunlogconfigs ${configname} -o yaml を実行します。

  設定のstatus フィールドには、設定の実行結果が表示されます。 設定が正常に適用された場合、 statusCode の値は status フィールドで 200 になります。 statusCode の値が 200 でない場合、設定の適用に失敗しています。

```
[root@iZbp1dsbiaZ ~]# kubectl get aliyunlogconfigs simple-file-example -o yaml
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
annotations:
  kubectl.kubernetes.io/last-applied-configuration: |

clusterName: ""
creationTimestamp: 2018-05-17T08:44:46Z
generation: 0
name: simple-file-example
namespace: default
resourceVersion: "21790443"
selfLink: /apis/log.alibabacloud.com/v1alpha1/namespaces/default/aliyunlogc
onfigs/simple-file-example
uid: 8d3a09c4-59ae-11e8-851d-00163f008685
spec:
lifeCycle: null
logstore: k8s-file
logtailConfig:
  configName: simple-file-example
  inputDetail:
    dockerFile: true
    dockerIncludeEnv:
      ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
    filePattern: simple.LOG
    logPath: /usr/local/ilogtail
    logType: common_reg_log
  inputType: file
machineGroups: null
project: ""
shardCount: null
status:
status: OK
statusCode: 200
```

設定例

コンテナ標準出力

コンテナ標準出力では、inputType を plugin に設定します。inputDetail の plugin フィールドの詳細情報を入力します 。 設定フィールドの詳細については、「コンテナ標準出力」をご参照ください。

- シンプル収集モード

  環境変数の設定 COLLECT_STDOUT_FLAG=false があるコンテナを除く、すべてのコンテナの標準出力 (stdout と stdeer) を収集します。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # your config name, must be unique in your k8s cluster
  name: simple-stdout-example
spec:
  # logstore name to upload log
  logstore: k8s-stdout
  # logtail config detail
  logtailConfig:
    # docker stdout's input type is 'plugin'
    inputType: plugin
    # logtail config name, should be same with [metadata.name]
    configName: simple-stdout-example
    inputDetail:
      plugin:
        inputs:
          -
            # input type
            type: service_docker_stdout
            detail:
              # collect stdout and stderr
              Stdout: true
              Stderr: true
              # collect all container's stdout except containers with "COLLECT_STDOUT_FLAG:
false" in docker env config
              ExcludeEnv:
```

```
            COLLECT_STDOUT_FLAG: "false"
```

- カスタム収集モード

  Grafana のアクセスログを収集し、アクセスログを構造化データに解析します。

  Grafana コンテナには環境変数 GF_INSTALL_PLUGINS=grafana-piechart-.... の設定があります。 Logtail がこのコンテナのみから標準出力を収集できるように IncludeEnv を GF_INSTALL_PLUGINS: '' に設定できます。

  図 3-34：カスタム収集モード

  Grafana のアクセスログの形式は次のとおりです。

```
t=2018-03-09T07:14:03+0000 lvl=info msg="Request Completed" logger=context
userId=0 orgId=0 uname= method=GET path=/ status=302 remote_addr=172.16.64.154
 time_ms=0 size=29 referer=
```

  正規表現を使用してアクセスログを解析します。 詳細な設定は次のとおりです。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # your config name, must be unique in your k8s cluster
  name: regex-stdout-example
spec:
  # logstore name to upload log
  logstore: k8s-stdout-regex
  # logtail config detail
  logtailConfig:
    # docker stdouts input type is plugin
    inputType: plugin
    # logtail config name, should be same with [metadata.name]
    configName: regex-stdout-example
    inputDetail:
      plugin:
        inputs:
          -
            # input type
            type: service_docker_stdout
            detail:
              # stdout 出力のみを収集し、stdeer 出力は収集しません。
              Stdout: true
              Stderr: false
```

```
        # コンテナの環境変数設定でキーが “GF_INSTALL_PLUGINS” である stdout 出力のみを収集します。
        IncludeEnv:
          GF_INSTALL_PLUGINS: ''
      processors:
        -
          # 正規表現を使用します。
          type: processor_regex
          detail:
            # docker によって収集されたデータには、デフォルトで、キー “コンテンツ” があります。
            SourceKey: content
            # 抽出の正規表現
            Regex: 't=(\d+-\d+-\w+:\d+:\d+\+\d+) lvl=(\w+) msg="([^"]+)" logger=(\w+) userId=(\w+) orgId=(\w+) uname=(\S*) method=(\w+) path=(\S+) status=(\d+) remote_addr=(\S+) time_ms=(\d+) size=(\d+) referer=(\S*). *'
            # 抽出されたキー
            Keys: ['time', 'level', 'message', 'logger', 'userId', 'orgId', 'uname', 'method', 'path', 'status', 'remote_addr', 'time_ms', 'size', 'referer']
            # 元のフィールドを保持します。
            KeepSource: true
            NoKeyError: true
            NoMatchError: true
```

設定が適用されると、 Log Service によって収集されるデータは次のようになります。

図 3-35：収集されるログデータ

コンテナファイル

- シンプルファイル

  環境変数設定にキー ALIYUN_LOGTAIL_USER_DEFINED_ID が含まれるコンテナからログファイルを収集します。 ログファイルのパスは/data/logs/app_1で、ファイル名は simple.LOG です。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # your config name, must be unique in your k8s cluster
  name: simple-file-example
spec:
  # logstore name to upload log
  logstore: k8s-file
  # logtail config detail
  logtailConfig:
```

```
    # log file's input type is 'file'
    inputType: file
    # logtail config name, must same with [metadata.name]
    configName: simple-file-example
    inputDetail:
      # 正規表現型のログについては、logType を “common_reg_log”に設定します。
      logType: common_reg_log
      # ログファイルフォルダ
      logPath: /data/logs/app_1
      # ワイルドカードをサポートするファイル名。たとえば、log.log
      filePattern: simple.LOG
      # コンテナからファイルを収集します。dockerFile フラグは true に設定されています。
      dockerFile: true
      # Only collect container with "ALIYUN_LOGTAIL_USER_DEFINED_ID" in docker env
config
      dockerIncludeEnv:
        ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

- 正規表現ファイルを完成する

  Java プログラムログの例を次に示します。

```
[2018-05-11T20:10:16,000] [INFO] [SessionTracker] [SessionTrackerImpl.java:148]
Expiring sessions
java.sql.SQLException: Incorrect string value: '\xF0\x9F\x8E\x8F",...' for column 'data'
at row 1
at org.springframework.jdbc.support.AbstractFallbackSQLExceptionTranslator.
translate(AbstractFallbackSQLExceptionTranslator.java:84)
at org.springframework.jdbc.support.AbstractFallbackSQLException
```

  ログにはエラースタック情報が含まれているため、ログエントリは複数の行に分割されることがあります。 したがって、行の先頭に正規表現を設定する必要があります。 各フィールドを抽出するには、正規表現を使用します。 設定の詳細は次のとおりです。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # your config name, must be unique in your k8s cluster
  name: regex-file-example
spec:
  # logstore name to upload log
  logstore: k8s-file
  logtailConfig:
    # log file's input type is 'file'
    inputType: file
    # logtail config name, should be same with [metadata.name]
    configName: regex-file-example
    inputDetail:
      # 正規表現型のログについては、logType を “common_reg_log” に設定します。
      logType: common_reg_log
      # ログファイルフォルダ
      logPath: /app/logs
      # ワイルドカードをサポートするファイル名。たとえば、log.log
      filePattern: error.LOG
      # 最初の行の正規表現
      logBeginRegex: '\[\d+-\d+-\w+:\d+:\d+,\d+]\s\[\w+]\s. *'
      # 正規表現を解析します。
      regex: '\[([^]]+)]\s\[(\w+)]\s\[(\w+)]\s\[([^:]+):(\d+)]\s(. *)'
      # 抽出されたキーのリスト
```

```
    key : ["time", "level", "method", "file", "line", "message"]
    # 正規表現のログ。 ログの timeはデフォルトで時間解析のために抽出されます。 時間
が必要ない場合は、フィールドを無視します。
    timeFormat: '%Y-%m-%dT%H:%M:%S'
    # コンテナからファイルを収集します。dockerFile フラグは true に設定されています。
    dockerFile: true
    # Only collect container with "ALIYUN_LOGTAIL_USER_DEFINED_ID" in docker env
config
    dockerIncludeEnv:
     ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

設定が適用されると、 Log Service によって収集されるデータは次のようになります。

図 3-36：収集されるログデータ

- 区切り文字パターンファイル

  Logtail は、区切り文字モードでのログ解析をサポートしています。例は次のとおりです。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
 # your config name, must be unique in your k8s cluster
 name: delimiter-file-example
spec:
 # logstore name to upload log
 logstore: k8s-file
 logtailConfig:
  # log file's input type is 'file'
  inputType: file
  configName: delimiter-file-example
  # logtail config name, should be same with [metadata.name]
  inputDetail:
   # 区切り文字タイプのログの場合は、logType を delimiter_log に設定します。
   logType: delimiter_log
   # ログファイルフォルダ
   logPath: /usr/local/ilogtail
   # ワイルドカードをサポートするファイル名。たとえば、 log.log
   filePattern: delimiter_log.LOG
   # 複数文字の区切り文字を使用します。
   separator: '|&|'
   # 抽出されたキーのリスト
   key : ['time', 'level', 'method', 'file', 'line', 'message']
   # 解析時間のキー。 時間の解析が不要な場合はフィールドを無視します。
   timeKey: 'time'
   # 時間解析メソッド。 時間の解析が不要な場合はフィールドを無視します。
   timeFormat: '%Y-%m-%dT%H:%M:%S'
   # コンテナからファイルを収集します。dockerFile フラグは true に設定されています。
```

```
    dockerFile: true
    # Only collect container with "ALIYUN_LOGTAIL_USER_DEFINED_ID" in docker env
config
    dockerIncludeEnv:
     ALIYUN_LOGTAIL_USER_DEFINED_ID: ''
```

- **JSON** モードファイル

  ファイル内の各データ行が JSON オブジェクトの場合、JSON メソッドを解析に使用できます。例は次のとおりです。

```
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
 # your config name, must be unique in you k8s cluster
 name: json-file-example
spec:
 # logstore name to upload log
 logstore: k8s-file
 logtailConfig:
  # log file's input type is 'file'
  inputType: file
  # logtail config name, should be same with [metadata.name]
  configName: json-file-example
  inputDetail:
   # 区切り文字タイプのログの場合は、logType を json_log に設定します。
   logType: json_log
   # ログファイルフォルダ
   logPath: /usr/local/ilogtail
   # ワイルドカードをサポートするファイル名。たとえば、log_*.log
   filePattern: json_log.LOG
   # 解析時間のキー。 時間の解析が不要な場合はフィールドを無視します。
   timeKey: 'time'
   # 時間解析メソッド。 時間の解析が不要な場合はフィールドを無視します。
   timeFormat: '%Y-%m-%dT%H:%M:%S'
   # コンテナからファイルを収集します。dockerFile フラグは true に設定されている
   dockerFile: true
   # Only collect container with "ALIYUN_LOGTAIL_USER_DEFINED_ID" in docker env
config
   dockerIncludeEnv:
    ALIYUN_LOGTAIL_USER_DEFINED_ID: ""
```

3.6.5 Kubernetes-Sidecar ログ収集モード

Logtail は Kubernetes から Sidecar モードでログを収集し、ログ収集を必要とするサービス容器ごとに Sidecar 容器を作成できるため、マルチ容器の分離が容易になり、収集パフォーマンスが向上します。

現在、Kubernetes クラスタにインストールされているデフォルトのログコンポーネントは DaemonSet です。これは、O&M の操作を簡素化し、リソースの占有が少なく、容器の標準出力と容器ファイルの収集をサポートし、柔軟に構成できます。

ただし、DaemonSet モードでは、Logtail はノード上のすべての容器からログを収集する必要があります。 これはパフォーマンスのボトルネックにつながり、サービスログ間の完全な分離

はできません。 この問題を解決するために、Logtail は Sidecar を提供します。これにより、Logtail はログ収集を必要とする各サービス容器に対して Sidecar 容器を作成できます。 このモードでは、マルチ容器間の分離が大幅に強化され、収集パフォーマンスが向上します。 大規模な Kubernetes クラスタ、および複数のサービスを提供する PaaS プラットフォームとして機能するクラスタには、Sidecar モードの使用をお勧めします。

機能

- Sidecar モードは、Kubernetes、オンプレミス ECS Kubernetes、および IDC のオンプレミス Kubernetes の容器サービスに適用できます。
- Sidecar モードでは、Logtail は Pod名、Pod IP アドレス、Pod ネームスペース、およびPod が属するノードの名前と IP アドレスを含む、Pod メタデータを収集できます。
- Sidecar モードでは、Logtail は CustomResourceDefinition（CRD）を介してプロジェクト、Logstores、インデックス、Logtail 構成、マシングループなどの Log Service リソースを自動的に作成できます。
- Sidecar モードは動的スケーリングもサポートします。 レプリカの数はいつでも調整でき、変更はすぐに有効になります。

コンセプト

Sidecar モードでは、ログ収集で Logtail がログディレクトリをサービス容器と共有する必要があります。 簡単に言うと、サービス容器はログをログディレクトリに書き込み、Logtail はログディレクトリ内のログファイルの変更をモニタリングしてログを収集します。 詳細については、以下の説明をご参照ください。

1. Sidecar ログ収集モードの紹介
2. Sidecar モードの例

前提条件

1. Log Service を有効化しました。

   まだ Log Service を有効にしていない場合は 、まず有効化します。
2. CRD ベースの設定用にKubernetes のログ収集をインストールしました。

制限事項

1. Logtail はログディレクトリをサービス容器と共有する必要があります。
2. Sidecar モードは容器の標準出力の収集をサポートしません。

Sidecar 構成

Sidecar 構成には以下が含まれます：

1. 基本操作パラメータの設定
2. マウントパスの設定

例は次となります：

```
apiVersion: batch/v1
kind: Job
metadata:
  name: nginx-log-sidecar-demo
  namespace: default
spec:
  template:
    metadata:
      name: nginx-log-sidecar-demo
    spec:
      restartPolicy: Never
      containers:
      - name: nginx-log-demo
        image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
        command: ["/bin/mock_log"]
        args: ["--log-type=nginx", "--stdout=false", "--stderr=true", "--path=/var/log/nginx/
access.log", "--total-count=1000000000", "--logs-per-sec=100"]
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
      ##### logtail sidecar container
      - name: logtail
        # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/
logtail/detail
        # this images is released for every region
        image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest
        livenessProbe:
          exec:
            command:
            - /etc/init.d/ilogtaild
            - status
          initialDelaySeconds: 30
          periodSeconds: 30
        resources:
          limits:
            memory: 512Mi
          requests:
            cpu: 10m
            memory: 30Mi
        env:
          ##### base config
          # user id
          - name: "ALIYUN_LOGTAIL_USER_ID"
            value: "${your_aliyun_user_id}"
          # user defined id
          - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
            value: "${your_machine_group_user_defined_id}"
          # config file path in logtail's container
          - name: "ALIYUN_LOGTAIL_CONFIG"
            value: "/etc/ilogtail/conf/${your_region_config}/ilogtail_config.json"
          ##### env tags config
          - name: "ALIYUN_LOG_ENV_TAGS"
            value: "_pod_name_|_pod_ip_|_namespace_|_node_name_|_node_ip_"
          - name: "_pod_name_"
            valueFrom:
              fieldRef:
                fieldPath: metadata.name
```

```
        - name: "_pod_ip_"
          valueFrom:
            fieldRef:
              fieldPath: status.podIP
        - name: "_namespace_"
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        - name: "_node_name_"
          valueFrom:
            fieldRef:
              fieldPath: spec.nodeName
        - name: "_node_ip_"
          valueFrom:
            fieldRef:
              fieldPath: status.hostIP
      volumeMounts:
      - name: nginx-log
        mountPath: /var/log/nginx
    ##### share this volume
    volumes:
    - name: nginx-log
      emptyDir: {}
```

構成 1：基本動作パラメータを設定します。

主なパラメータとその設定は次のとおりです。

```
##### base config
        # user id
        - name: "ALIYUN_LOGTAIL_USER_ID"
          value: "${your_aliyun_user_id}"
        # user defined id
        - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
          value: "${your_machine_group_user_defined_id}"
        # config file path in logtail's container
        - name: "ALIYUN_LOGTAIL_CONFIG"
          value: "/etc/ilogtail/conf/${your_region_config}/ilogtail_config.json"
```

パラメーター	説明
$ {your_region_config}	このパラメータは、プロジェクトのリージョンとネットワークの種類によって決まります。 ネットワークの種類に応じて適切な値に設定します。 有効な値： • インターネット：region-internet。 たとえば、中国（杭州）リージョンの値は cn-hangzhou-internet です。 • Alibaba Cloud イントラネット：region。 たとえば、中国（杭州）リージョンの値は cn-hangzhou です。 このパラメータでの region は#unique_35/unique_35_Connect_42_table_eyz_pmv_vdbです。 プロジェクトが属するリージョンに設定します。

パラメーター	説明
${your_aliyun_user_id}	このパラメータは、ユーザー ID を指定します。これは、文字列形式の **Alibaba Cloud** アカウント ID に置き換える必要があります。 IDのクエリの方法については、 ユーザー ID の設定のセクション 2.1 をご参照ください。  注： このパラメータ値は、 **Alibaba Cloud** アカウント ID である必要があります。 RAM ユーザー ID の場合は無効になります。
${your_machine_group_user_defined_id}	このパラメータは、クラスタ内のマシングループのカスタム ID を指定します。 ID は、Log Service がデプロイされているリージョン内で一意である必要があります。 詳細は、マシングループにユーザー定義 ID を設定するを参照ください。

構成 2：マウントパスを設定します。

1. Logtail とサービス容器は同じディレクトリにマウントする必要があります。
2. emptyDir のマウント方法をお勧めします。

マウントパスの例は、前述の構成例に示されています。

ログ収集設定

ログ収集は、CRD または Log Service コンソールを介して設定できます。 CRD ベースの設定は、プロジェクト、ログストア、インデックス、マシングループ、および Logtail 構成の自動作成をサポートし、Kubernetes と簡単に統合できます。 そのため、CRD ベースの設定をお勧めします。 Kubernetes のログ収集を初めて使用、またはデバッグするユーザーにとっても、コンソールベースの設定が簡単です。

CRD ベースの設定

詳細は、CRD での Kubernetes ログ収集の設定をご参照ください。 DaemonSet 収集モードと比較して、CRD ベースの設定には以下の制限があります。

1. ログ収集が必要なプロジェクトの名前を指定しなければなりません。 そうしないと、ログが収集され、ログコンポーネントがデフォルトでインストールされているプロジェクトに送信されます。
2. 設定を有効にするには、マシングループを指定する必要があります。 そうしないと、設定は DaemonSet が属するマシングループにデフォルトで適用されます。
3. Sidecar モードはファイル収集のみをサポートします、ファイル収集している間に dockerFile を false に設定する必要があります。

詳細は、次の例をご参照ください。

コンソールベースの設定

1. マシングループの構成。

   Log Service コンソールで、Pod IP アドレスの変更に動的に適応するために、ID がカスタム ID に設定された Logtail マシングループを作成します。 そうするには、次の手順を実行します：

   a. Log Service を有効にして、プロジェクトとログストアを作成します。 詳細は、#unique_94をご参照ください。
   b. マシングループリストページで、 マシングループの作成をクリックします。
   c. ID をカスタム ID ALIYUN_LOGTAIL_USER_DEFINED_ID に設定します。

2. 収集モードの設定。

   対象のファイルの収集詳細を設定します。 現在、シンプルモード、Nginx アクセスモード、区切り文字モード、JSON モード、通常モードなど、さまざまなモードがサポートされています。詳細は、テキストファイルの収集次をご参照ください。

   次の図にこの例の設定を示します。

   注：

Docker ファイルを無効にする必要があります。


* 配置名称:
nginx-log-sidecar
* 日志路径:
/var/log/nginx
/**/
access
指定文件夹下所有符合文件名称的文件都会被监控到(包含所有层次的目
符模式匹配。Linux文件路径只支持/开头，例：/apsara/nuwa/.../app.Lo
如：C:\Program Files\Intel\...*.Log
是否为Docker文件:
如果是Docker容器内部文件，可以直接配置内部路径与容器Tag，Logta
进行过滤采集指定容器的日志，具体说明参考文档链接
模式:
分隔符模式
如何设置Delimiter类型配置
日志样例:
2018-09-26T03:16:53.033307075Z 10.200.98.220 - - "POST /PutData
Category=YunOsAccountOpLog&AccessKeyId=Uxxxx45A&Date=Fri%
A53%3A30%20GMT&Topic=raw&Signature=pD12XYLmGxKQ%2Bm
18204 200 37 "-" "aliyun-sdk-java" 1
请贴入需要解析的日志样例(支持多条) 常见样例>>
* 分隔符:
空格
引用符:
双引号
双引号（"）作为Quote时，内部包含分隔符的字段需要被一对Quote包
包含空格、制表符等字符，请修改格式。

例

シナリオ：

1. Kubernetes クラスタは IDC のオンプレミスクラスタであり、Log Service がデプロイされているリージョンは中国（杭州）です。 ログはインターネットから収集されます。
2. 次の例では、マウントオブジェクトは nginx-log で、マウントタイプは emptyDir です。 これらはそれぞれ nginx-log-demo および logtail 容器内の /var/log/nginx ディレクトリにマウントされています。
3. アクセスログは /var/log/nginx/access.log で、保存先ログストアは nginx-access です。
4. エラーログは /var/log/nginx/error.log で、保存先ログストアは nginx-error です。

- **Sidecar** 設定：

```
apiVersion: batch/v1
kind: Job
metadata:
  name: nginx-log-sidecar-demo
  namespace: default
spec:
  template:
    metadata:
      name: nginx-log-sidecar-demo
    spec:
      restartPolicy: Never
      containers:
      - name: nginx-log-demo
        image: registry.cn-hangzhou.aliyuncs.com/log-service/docker-log-test:latest
        command: ["/bin/mock_log"]
        args: ["--log-type=nginx", "--stdout=false", "--stderr=true", "--path=/var/log/
nginx/access.log", "--total-count=1000000000", "--logs-per-sec=100"]
        volumeMounts:
        - name: nginx-log
          mountPath: /var/log/nginx
      ##### logtail sidecar container
      - name: logtail
        # more info: https://cr.console.aliyun.com/repository/cn-hangzhou/log-service/
logtail/detail
        # this images is released for every region
        image: registry.cn-hangzhou.aliyuncs.com/log-service/logtail:latest
        livenessProbe:
          exec:
            command:
            - /etc/init.d/ilogtaild
            - status
          initialDelaySeconds: 30
          periodSeconds: 30
        env:
          ##### base config
          # user id
          - name: "ALIYUN_LOGTAIL_USER_ID"
            value: "xxxxxxxxxx"
          # user defined id
          - name: "ALIYUN_LOGTAIL_USER_DEFINED_ID"
            value: "nginx-log-sidecar"
          # config file path in logtail's container
```

```
        - name: "ALIYUN_LOGTAIL_CONFIG"
          value: "/etc/ilogtail/conf/cn-hangzhou-internet/ilogtail_config.json"
        ##### env tags config
        - name: "ALIYUN_LOG_ENV_TAGS"
          value: "_pod_name_|_pod_ip_|_namespace_|_node_name_|_node_ip_"
        - name: "_pod_name_"
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        - name: "_pod_ip_"
          valueFrom:
            fieldRef:
              fieldPath: status.podIP
        - name: "_namespace_"
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        - name: "_node_name_"
          valueFrom:
            fieldRef:
              fieldPath: spec.nodeName
        - name: "_node_ip_"
          valueFrom:
            fieldRef:
              fieldPath: status.hostIP
      volumeMounts:
      - name: nginx-log
        mountPath: /var/log/nginx
    ##### share this volume
    volumes:
    - name: nginx-log
      emptyDir: {}
```

- **CRD** 設定：

```
# config for access log
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # your config name, must be unique in you k8s cluster
  name: nginx-log-access-example
spec:
  # project name to upload log
  project: k8s-nginx-sidecar-demo
  # logstore name to upload log
  logstore: nginx-access
  # machine group list to apply config, should be same with your sidecar' [ALIYUN_LOG
TAIL_USER_DEFINED_ID]
  machineGroups:
  - nginx-log-sidecar
  # logtail config detail
  logtailConfig:
    # log file's input type is 'file'
    inputType: file
    # logtail config name, should be same with [metadata.name]
    configName: nginx-log-access-example
    inputDetail:
      # Simple logs with logType set to common_reg_log
      logType: common_reg_log
      # Log folder
      logPath: /var/log/nginx
      # File name with wildcards supported, for example, log_*.log
      filePattern: access.log
```

```
    # Sidecar mode with dockerFile set to false
    dockerFile: false
    # Line start regular expression, which is set to .* is the log contains only a line
    logBeginRegex: '. *'
    # Regular expression for parsing
    regex: '(\S+)\s(\S+)\s\S+\s\S+\s"(\S+)\s(\S+)\s+([^"]+)"\s+(\S+)\s(\S+)\s(\d+)\s(\d
+)\s(\S+)\s"([^"]+)"\s. *'
    # List of the extracted keys
    key : ["time", "ip", "method", "url", "protocol", "latency", "payload", "status", "
response-size",ser-agent"]
# config for error log
```

```
# config for error log
apiVersion: log.alibabacloud.com/v1alpha1
kind: AliyunLogConfig
metadata:
  # your config name, must be unique in you k8s cluster
  name: nginx-log-error-example
spec:
  # project name to upload log
  project: k8s-nginx-sidecar-demo
  # logstore name to upload log
  logstore: nginx-error
  # machine group list to apply config, should be same with your sidecar' [ALIYUN_LOG
TAIL_USER_DEFINED_ID]
  machineGroups:
  - nginx-log-sidecar
  # logtail config detail
  logtailConfig:
    # log file's input type is 'file'
    inputType: file
    # logtail config name, should be same with [metadata.name]
    configName: nginx-log-error-example
    inputDetail:
      # Simple logs with logType set to common_reg_log
      logType: common_reg_log
      # Log folder
      logPath: /var/log/nginx
      # File name with wildcards supported, for example, log_*.log
      filePattern: error.log
      # Sidecar mode with dockerFile set to false
      dockerFile: false
```

- ログ収集エラーの表示

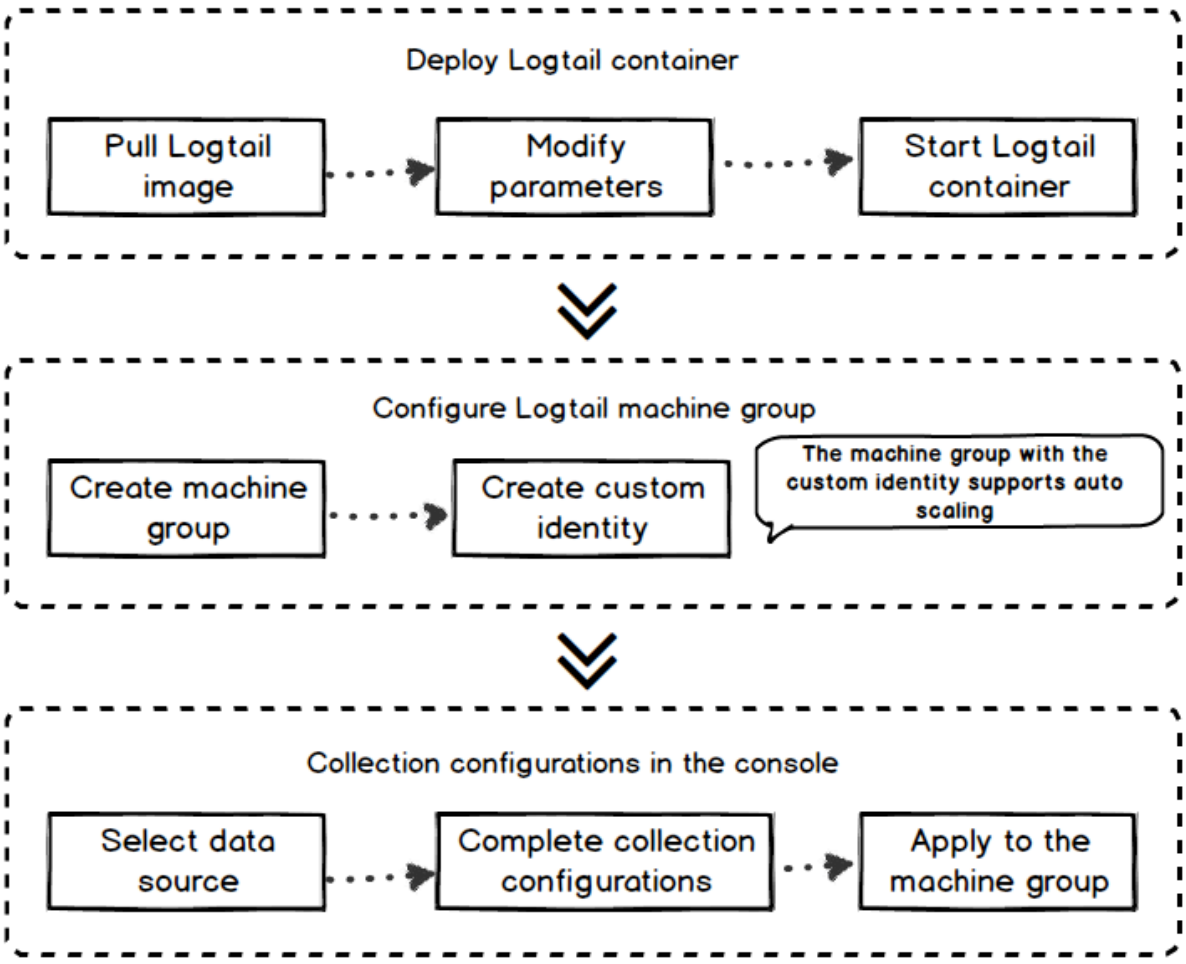
  上記の設定が Kubernetes クラスタに適用されると、Logtail 容器は対応するプロジェクト、ログストア、マシングループ、および Logtail 構成を自動的に作成し、収集されたログを Log Service に自動的に送信します。 Log Service コンソールにログインして詳細を表示できます。

3.6.6 標準の Docker ログの収集

Logtail は、標準の Docker ログを収集し、これらのログをコンテナ関連のメタデータ情報と一緒に Log Service にアップロードすることをサポートしています。

設定プロセス

図 3-37 : 設定プロセス



1. Logtail コンテナをデプロイします。
2. Logtail コンテナを構成します。

   ログサービスコンソールで、ユーザー定義の ID を持つマシングループを作成します。 その後、コンテナークラスターを拡張または縮小するために追加の操作および保守（O&M）は必要ありません。
3. コンソールにコレクション設定を作成します。

   ログサービスコンソールでコレクション構成を作成します。 すべてのコレクション構成はサーバー側用です。 ローカル設定は必要ありません。

ステップ 1 Logtail コンテナをデプロイする

1. Logtail 画像を引き出します。

```
docker pull registry.cn-hangzhou.aliyuncs.com/log-service/logtail
```

2. Logtail コンテナを開始します。

スタートアップテンプレートの${your_region_name}、 ${your_aliyun_user_id}、${your_machine_group_name}の 3 つのパラメータを置き換えてください。

```
docker run-d -v /:/logtail_host:ro -v /var/run/docker.sock:/var/run/docker.sock --env
ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/${your_region_name}/ilogtail_config.json

--env ALIYUN_LOGTAIL_USER_ID=${your_aliyun_user_id} --env
 ALIYUN_LOGTAIL_USER_DEFINED_ID=${your_machine_group_user_defined_id} registry.
cn-hangzhou.aliyuncs.com/log-service/logtail
```

 注：

構成パラメータの前に次の構成を実行してください。そうしないと、他のコンテナを削除する際に次のエラーが発生する可能性があります。container text file busy。

- CentOS バージョン 7.4 以降では fs.may_detach_mounts = 1が設定されています。詳細は、Bug 1468249、Bug 1441737およびissue 34538 をご参照ください。
- Logtail に privileged 権限を与え、起動パラメータに ---privileged を追加します。 詳細は、docker run command をご参照ください。

パラメータ	説明
${your_region_name}	リージョン名。 作成したログサービスプロジェクトが存在するリージョンに置き換えます。 リージョン名については、Logtail の Linux へのインストール で使われているリージョン名をご参照ください。  注： リージョン名をリストから直接コピーすることをお勧めします。

パラメータ	説明
$ {your_aliyun_user_id}	ユーザ ID。メインの Alibaba Cloud アカウントの ID で置き換えてください。 String 型の Alibaba Cloud メインアカウントの ID に置き換えてください。 ID の確認方法については、Alibaba Cloud ECS インスタンス以外または他のアカウントの ECS インスタンスからログを収集する の 2.1 をご参照ください。
${your_machine_group_user_defined_id}	クラスタマシングループのユーザー定義の ID。 ユーザー定義の ID がまだ有効になっていない場合は、マシングループにユーザー定義 ID を設定するの該当する手順に従って userdefined-id を有効にします。

```
docker run -d -v /:/logtail_host:ro -v /var/run/docker.sock:/var/run/docker.sock
--env ALIYUN_LOGTAIL_CONFIG=/etc/ilogtail/conf/cn_hangzhou/ilogtail_config.json
 --env
 ALIYUN_LOGTAIL_USER_ID=1654218******--env ALIYUN_LOGTAIL_USER_DEFINED_ID=
log-docker-demo registry.cn-hangzhou.aliyuncs.com/log-service/logtail
```

注：

次の条件が満たされている場合は、Logtail コンテナの起動パラメータ設定をカスタマイズできます。

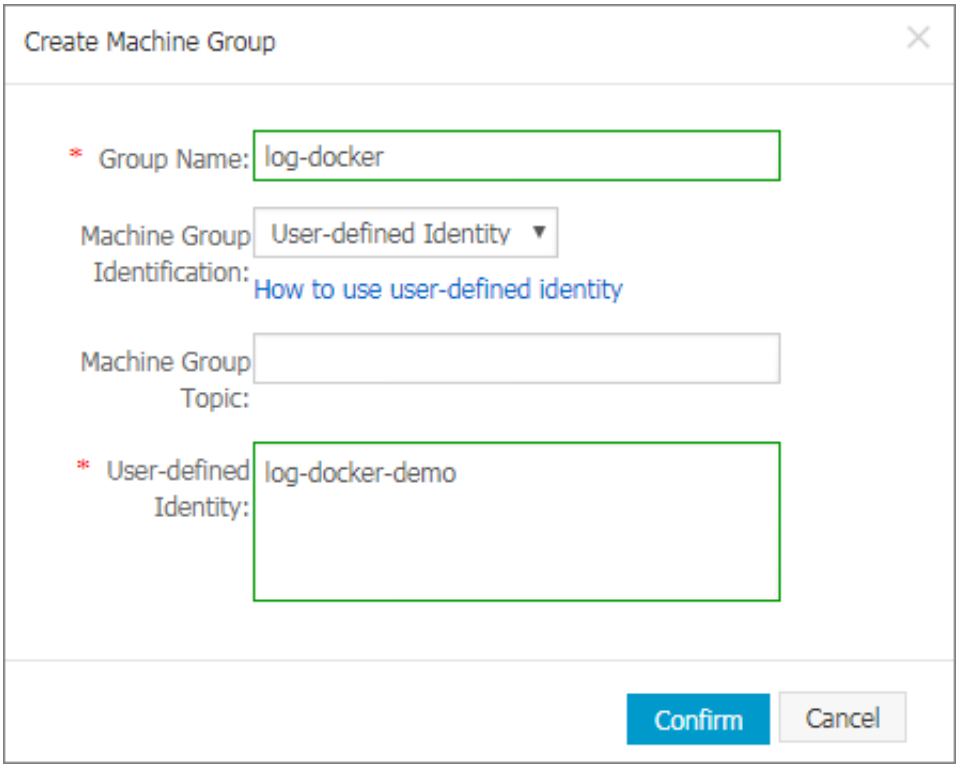
1. Logtail コンテナを起動するときは、ALIYUN_LOGTAIL_USER_DEFINED_ID、ALIYUN_LOGTAIL_USER_ID、および ALIYUN_LOGTAIL_CONFIG という 3 つの環境変数があります。
2. Docker のドメインソケットは /var/run/docker.sock にマウントされています。
3. 他のコンテナまたはホストファイルからログを収集する場合、ルートディレクトリは Logtail コンテナの / logtail_host ディレクトリにマウントされます。
4. Logtail ログ /usr/local/ilogtail/ilogtail.LOG にはパラメータ無効：uuid=none というエラーログがある場合、ホストマシンで product_uuid ファイルを作成し、有効な UUID（例：169E98C9-ABC0 -4A92-B1D2-AA6239C0D261）を入力して、Logtail コンテナの /sys/class/dmi/id/product_uuid パスにマウントします。

ステップ 2. マシングループを構成する

1. Log Service を有効にして、プロジェクトとログストアを作成します。 詳細については、#unique_94 をご参照ください。
2. Log Service コンソール内の[マシングループ]ページでLogtail マシングループの作成 をクリックします。

3. ユーザー定義の **ID** を 「マシングループの識別」ドロップダウンリストから選択します。 前の手順で設定した ALIYUN_LOGTAIL_USER_DEFINED_ID を[ユーザー定義の ID]フィールドに入力します。

図 3-38：マシングループの構成



[確認]をクリックして、マシングループを作成します。 1 分後、マシングループページの右側にあるマシンステータスをクリックして、デプロイされた Logtail コンテナのハートビートステータスを表示します。 詳細については、マシングループの管理のステータスの表示をご参照ください。

ステップ 3. 収集の構成情報を作成する

必要に応じてコンソールに Logtail 収集構成情報を作成します。 収集構成情報の作成方法については、以下をご参照ください。

- Container text log (recommended)
- Container standard output (recommended)
- ホストテキストファイル

  デフォルトでは、ホストのルートディレクトリは Logtail 容器の /logtail_host ディレクトリにマウントされています。 構成パスの先頭に/ logtail_hostを付ける必要があります。 たとえば、ホストの /home/logs/app_log/ ディレクトリからデータを収集するには、構成ページのログパスを /logtail_host/home/logs/app_log/ に設定します。

- #unique_45

その他の操作

- Logtail コンテナの動作状態を確認する

  docker exec $ {logtail_container_id} /etc/init.d/ilogtaild status コマンドを実行すると、Logtail の実行状態を確認できます。

- バージョン番号、IP、および Logtail の起動時間を確認する

  docker exec $ {logtail_container_id} cat / usr / local / ilogtail / app_info.json コマンドを実行して、Logtail に関する情報を確認できます。

- ログの実行ログを確認する

  Logtail 実行ログは、/usr/local/ilogtail/ ディレクトリに格納されます。 ファイル名は、ilogtail.LOG です。 rotation ファイルは圧縮され、ilogtail.LOG.x.gz として保存されます。

  例えば：

```
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 tail -n 5 /usr/
local/ilogtail/ilogtail.LOG
[2018-02-06 08:13:35.721864] [INFO] [8] [build/release64/sls/ilogtail/LogtailPlugin.
cpp:104] logtail plugin Resume:start
[2018-02-06 08:13:35.722135] [INFO] [8] [build/release64/sls/ilogtail/LogtailPlugin.
cpp:106] logtail plugin Resume:success
[2018-02-06 08:13:35.722149] [INFO] [8] [build/release64/sls/ilogtail/EventDispatcher.
cpp:369] start add existed check point events, size:0
[2018-02-06 08:13:35.722155] [INFO] [8] [build/release64/sls/ilogtail/EventDispatcher
.cpp:511] add existed check point events, size:0 cache size:0 event size:0 success
count:0
[2018-02-06 08:13:39.725417] [INFO] [8] [build/release64/sls/ilogtail/ConfigManager.
cpp:3776] check container path update flag:0 size:1
```

  コンテナの stdout は参照になりません。次の stdout の出力は無視してください。

```
start umount useless mount points, /shm$|/merged$|/mqueue$
umount: /logtail_host/var/lib/docker/overlay2/3fd0043af174cb0273c3c7869500fb
e2bdb95d13b1e110172ef57fe840c82155/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/d5b10aa19399992755de1f85d25009
528daa749c1bf8c16edff44beab6e69718/merged: must be superuser to unmount
umount: /logtail_host/var/lib/docker/overlay2/5c3125daddacedec29df72ad0c52fa
c800cd56c6e880dc4e8a640b1e16c22dbe/merged: must be superuser to unmount
......
xargs: umount: exited with status 255; aborting
umount done
start logtail
ilogtail is running
logtail status:
```

```
ilogtail is running
```

- Logtail の再起動

  Logtail を再起動するには、次の例をご参照ください。

```
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/
ilogtaild stop
kill process Name: ilogtail pid: 7
kill process Name: ilogtail pid: 8
stop success
[root@iZbp17enxc2us3624wexh2Z ilogtail]# docker exec a287de895e40 /etc/init.d/
ilogtaild start
ilogtail is running
```

3.7 制限

表 3-3：ファイル収集の制限

項目	能力と制限
ファイルエンコーディング	UTF-8 と GBK でエンコードされたログファイルがサポートされています。 他の形式でエンコードされたログファイルは、不器用さやデータ損失などの未定義の動作につながります。 処理パフォーマンスを向上させるには、UTF-8 エンコーディングを使用することをお勧めします。
ログファイルサイズ	無制限。
ログファイルのローテーション	.log *と .logファイルの両方がサポートされています。
ログ解析の輻輳時のログ収集動作	ログ解析で輻輳が発生すると、Logtail FD のオープン状態が維持されます。 輻輳中にログファイルのローテーションが複数回発生すると、Logtail は回転ログの解析シーケンスを保持しようとします。 解析されていない 20 以上のログが回転した場合、Logtail は後続のログファイルを処理しません。 ソフトリンクのサポート詳細は、こちらを参照してください。
シングルログサイズ	監視対象ディレクトリはソフトリンクにすることができます。

項目	能力と制限
シングルログサイズ	各ログのサイズは 512 KB を超えることはできません。 複数行のログが行頭の正規表現で分割されている場合、各ログの最大サイズは 512 KBです。ログサイズが 512 KBを超えると、ログは収集のために複数の部分に分割されます。たとえば、一つのログは 1025 KBです。 最初の 512 KBは最初に処理され、その後の 512 KBは 2 回目に処理され、最後の 1 KBは 3 回目に処理されます。
正規表現タイプ	Perl と互換性のある正規表現を使用します。
同じファイルの複数の収集設定	サポートされていません。 ログファイルをログストアに収集し、複数のサブスクリプションを構成することをお勧めします。この機能が必要な場合は、ログファイルのソフトリンクを設定してこの制限をバイパスします。
ファイルオープン動作	Logtail はファイルをオープン状態で収集します。 5 分間ファイルを変更しないと、Logtail はファイルを閉じます。
最初のログ収集動作	Logtail は、増分ログファイルのみを収集します。 最初にファイルに変更があり、ファイルサイズが 1 MB を超える場合、Logtail は最後の 1 MB からログを収集します。 それ以外の場合、Logtail は最初からログを収集します。 設定が発行された後にログファイルが変更されない場合、Logtail はこのファイルを収集しません。
非標準テキストログ	ログに' \0'を含む行。 ログは最初の '0'に切り捨てられます。

表 3-4 : チェックポイント管理

項目	能力と制限
チェックポイントタイムアウト期間	ファイルが 30 日以上変更されていない場合、チェックポイントは削除されます。
チェックポイントストレージポリシー	定期的に 15 分ごとに保存し、プログラムが終了すると自動的に保存されます。

項目	能力と制限
チェックポイント保存パス	デフォルトの保存パスは/ tmp / logtail_checkpointです。Logtail 起動設定パラメーターに従ってパラメータを変更できます。

表 3-5：構成の制限

項目	能力と制限
構成情報の更新	更新された構成は約 30 秒後に有効になります。
動的構成ロード	サポートされる。 構成の更新は他のコレクションには影響しません。
構成の数	理論的には無制限。 サーバーの収集構成の数は 100 以下にすることをお勧めします。
マルチテナント分離	収集構成情報間の分離。

表 3-6：リソースとパフォーマンスの制限

項目	能力と制限
ログ処理のスループット	raw ログトラフィックのデフォルトの制限は 2 MB/s です。（データはエンコードされ、圧縮された後にアップロードされますが、一般に圧縮率は 5〜10 倍です）。ログトラフィックが制限を超えると、ログが失われる可能性があります。 パラメータを調整するには、 Logtail 起動設定パラメーター configurations parameters を参照してください。
最大パフォーマンス	単一のコア条件での最大処理能力：単純なログファイルの場合は 100 MB/秒、正規表現を使用するログファイルの場合はデフォルトで 20 MB/秒（正規表現の複雑さに応じて）、a の場合は 40 MB/秒区切り文字ログファイル、JSON ログファイルの場合は 30 MB/秒です。 複数のログ処理スレッドを開始すると、パフォーマンスが 1.5〜3 倍向上します。
監視対象ディレクトリの数	Logtail は監視対象ディレクトリの深さを積極的に制限し、リソースを節約します。上限に達すると、Logtail はさらに多くのディレクトリとログファイルの監視を停止します。 Logtail は、最大 3,000 のディレクトリ（サブディレクトリを含む）を監視します。
既定のリソース制限	既定では、Logtail は CPU 使用率の最大 40% と 256 MB のメモリ使用量を占めます。 ログが高速に生成された場合は、Logtail 起動設定パラメーターを使用してパラメータを調整できます。
リソース制限を超える処理ポリシー	3 分で Logtail が占めるリソースが上限を超えた場合、Logtail は強制的に再起動され、データの損失や重複が発生する可能性があります。

表 3-7：エラー処理の制限

項目	能力と制限
ネットワークエラー処理	ネットワーク接続が異常な場合、Logtail は積極的に再試行し、自動的に再試行間隔を調整します。
リソースクォータの処理が最大割り当て量を超えた場合	データ転送速度が Logstore の最大割り当て量を超えた場合、Logtail はログ収集をブロックし、自動的に再試行します。
タイムアウトの最大リトライ時間	データ送信が連続して 6 時間以上失敗した場合、Logtail はデータを破棄します。
ステータス自己チェック	プログラムの異常終了やリソース制限を超えるなど、例外が発生した場合、Logtail は自動的に再起動します。

表 3-8：その他の制限

項目	能力と制限
ログ収集遅延	通常、ログがディスクにフラッシュされた後、Logtail によるログ収集の遅延は（輻輳を除いて）1 秒を超えません。
ログアップロードポリシー	Logtail は、ログをアップロードする前に自動的に同じファイルにログを集約します。 ログのアップロードは、2,000 を超えるログが生成されたり、ログファイルが 2 MB を超えたり、ログ収集が 3 秒を超えたりするという条件でトリガされます。

4 クラウドプロダクトのログ収集

4.1 クラウドサービスログ

Log Service は、Elastic Compute Service (ECS)、Object Storage Service (OSS)、Server Load Balancer (SLB) などのさまざまなクラウドサービスからログを収集できます。 ログには、運用情報、実行ステータス、ビジネスダイナミクスなどのクラウドサービス情報が記録されます。

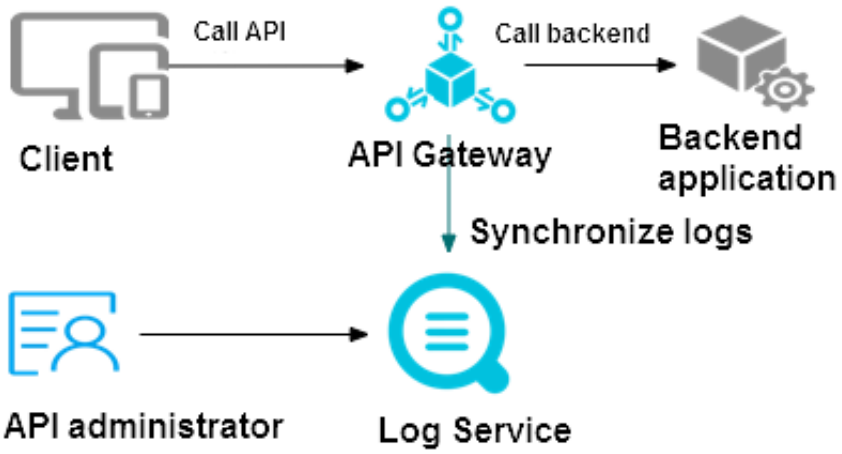
以下の表は、Log Service でログを収集できるクラウドプロダクトを示しています。

タイプ	クラウドサービス	有効化する方法	備考
エラスティックコンピューティング	ECS	Logtail をインストールします。	Logtail の紹介
	Container Service / Container Service for Kubernetes	Container Service または Container Service for Kubernetes コンソールにて有効化	テキストログ と 出力
ストレージ	OSS	OSS コンソールにて有効化	#unique_17
ネットワーク	SLB	SLB コンソールにて有効化	レイヤ 7 SLB のアクセスログ
	VPC	VPC コンソールにて有効化	#unique_19
	API Gateway	API Gateway コンソールにて有効化	API Gateway のアクセスログ
セキュリティ	ActionTrail	ActionTrail コンソールにて有効化	概要
	Anti-DDoS Pro /BGP-line Anti-DDoS Pro	Anti-DDoS Pro コンソールにて有効化	Anti-DDoS Pro 概要 と BGP-line Anti-DDoS Pro 概要
アプリケーション	Log service	Log Service コンソールにて有効化	#unique_24

4.2 API Gateway アクセスログ

Alibaba Cloud API Gateway は、マイクロサービスの集約、フロントエンドとバックエンドの分離、およびシステム統合を容易にする API ホスティングサービスを提供します。 アクセスログは、 Web サービスによって生成されるログです。 各 API リクエストは、呼び出し元 IP、要求された URL、応答待ち時間、返されたステータスコード、各要求と応答のバイト数、およびその他の情報を含むアクセスレコードに対応します。 前述の情報を使用すると、Web サービスの動作状況を理解できます。

図 4-1：APIゲートウェイ



Log Service を使用すると、 **Data Import Wizard** を使用して API Gateway アクセスログを収集できます。

特徴

1. オンラインログクエリ：ログ内の任意のキーワードを使用して、迅速で正確なファジークエリを実行できます。 この機能を使用すると、問題の特定やクエリのカウントに使用できます。
2. 詳細な通話記録： API 通話記録の詳細を検索することができます。
3. カスタマイズされた分析チャート：あなたのビジネス要件を満たす統計的な要件に応じて、任意のログ項目を統計チャートにカスタマイズすることができます。
4. 解析レポート： API Gateway は、要求量、成功率、失敗率、待ち時間、API を呼び出すアプリケーションの数、障害統計、TOP グループ化、 TOP API、および TOP 遅延を含むいくつかのグローバル統計図を事前定義します。

フィールドの説明

ログフィールド	説明
apiGroupUid	API グループ ID。
apiGroupName	API グループ名。
apiUid	API ID。
apiName	API 名。
apiStageUid	API ステージ ID。
apiStageName	API ステージ名。
httpMethod	呼び出される HTTP メソッド。
path	要求されたパス。
domain	呼び出されるドメイン名。
statusCode	HTTP ステータスコード
ErrorMessage	エラーメッセージ。
appId	呼び出し元のアプリケーション ID。
appName	呼び出し元のアプリケーション名。
clientIp	呼び出し側のクライアント IP。
Exception	バックエンドによって返された特定のエラーメッセージ。
providerAliUid	API プロバイダのアカウント ID。
region	リージョン、例えば cn-hangzhou。
requestHandleTime	リクエスト時間（GMT）。
RequestId	リクエスト ID。グローバルに一意です。
requestSize	リクエストのサイズ（バイト単位）。
Responsesize	返されるデータのサイズ（バイト単位）。
Servicelatency	バックエンドの待ち時間（ミリ秒単位）。

手順

1. 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、#unique_94をご参照ください。

   Logstore がすでに存在する場合は、この手順をスキップします。

2. データアクセスウィザードを開始します。

   Logstore を作成したら、**Logstore** リストページで Data Import Wizard アイコンをクリックしてください。

3. データタイプを選択します。

   クラウド製品ログの **API Gateway** をクリックし、 次へをクリックしてデータソース設定の手順に進みます。

4. データソースを設定します。

   データソース設定ステップで、次の設定を完了しているかどうかを確認します。

   a. **API Gateway** サービスを有効にする。

      API Gateway は完全な API ホスティングサービスを提供し、能力、サービス、およびデータを API の形式でパートナーに公開するのに役立ちます。

      API Gateway サービスを有効にしていない場合は、関連ページの指示に従ってサービスを有効にします。

   b. 完全な **Resource Access Management** （**RAM**） の承認ログ情報を **Logstore** に収集できるように、配布ルールを設定する前に **RAM** を使用して **Log Service** を認可します。

      配布ルールを設定する前に RAM を使用して Log Service を認可します。

      承認を迅速に行うには、右上隅の承認をクリックします。

   c. 配信ルールを確立する。

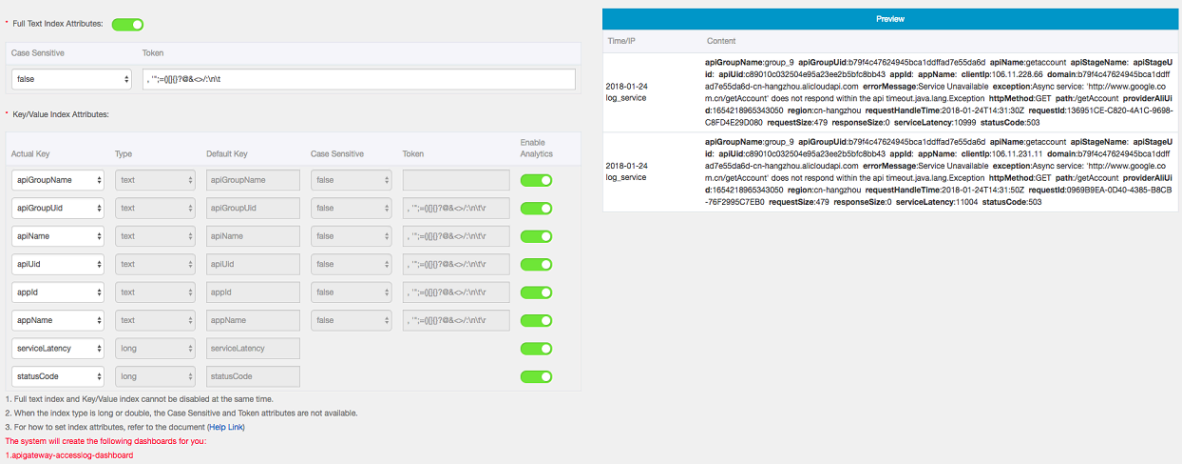
      この手順を初めて実行する場合、システムは自動的に API Gateway ログをインポートし、配布ルールを確立します。 以前に **API Gateway log collection** を設定していた場合、**Log distribution rules already exists** メッセージが表示されます。 既存の配布ルールを削除することもできます。

   次へをクリックして、検索と分析と視覚化ページに入ります。

5. 検索と分析と可視化を設定します。

   次の図に示すように、索引を構成します。 ダッシュボードでこの設定を使用するので、この設定を変更するときは注意する必要があります。

   図 4-2 : 索引の構成



   次へをクリックして設定を完了します。 ログシッパーは、必要に応じて個別に設定することができます。

ウィザードの初期化が完了しました。 設定した Logstore api-gateway-access-log を選択して、ログを照会および分析したり、ダッシュボードにアクセスしてレポートを表示することができます。

4.3 レイヤー 7 Server Load Balancer のアクセスログ

Alibaba Cloud の Server Load Balancer を使用することにより、複数の Elastic Compute Service (ECS) インスタンスにトラフィックを分散させることができます。Server Load Balancer には、レイヤー 4 Server Load Balancer (TCP)、および、レイヤー 7 Server Load Balancer (HTTP/HTTPS) があります。 Server Load Balancer を使用すると、1 つの ECS インスタンスに例外があった際、サービスへの影響が抑えられ、システム可用性は向上します。 Auto Scaling を併用し、 ECS のトラフィック量に応じてバックエンドサーバーを自動拡張/縮小させることもできます。

Server Load Balancer へのアクセス要求はすべて、アクセスログに記録されます。 アクセスログには、リクエスト時間、クライアントの IP アドレス、遅延、リクエストパス、およびサーバーレスポンスといった、Server Load Balancer に送信されるリクエストの詳細がすべて記録されます。 Server Load Balancer はインターネットアクセスポイントであるため、大量のアクセス要求

が処理されます。 そのアクセスログより、クライアントユーザーの行動パターンや地理的分布を分析することができます。また、問題のトラブルシューティングに役立てることもできます。

Server Load Balancer のアクセスログの収集には Log Service をご利用ください。 継続してレイヤー 7 (HTTP/HTTPS) のアクセスログをモニタリング、調査、診断、通知受信していくことにより、Server Load Balancer インスタンス全体を把握することができます。

 注：

Log Service は、レイヤー **7 Server Load Balancer** にのみ対応していますが、 全リージョンで利用できます。 詳細は、「#unique_96」をご参照ください。

利点

- シンプル: 開発者および管理者は、ログの処理に時間と手間をかける必要がなくなり、サービス開発および技術研究に専念できるようになります。
- 大容量処理: アクセスログのデータ量は、Server Load Balancer インスタンスのリクエスト PV に比例します。 一般的にデータ量が多いため、 アクセスログの処理に、コストパフォーマンスを考慮することは必須です。 Log Service は 1 秒で 1 億のログを分析することができるため、オープンソースのソリューションと比較してコスト面において確実に利点があります。
- リアルタイム性: DevOps、モニタリング、警告には、ログデータにリアルタイム性が求められます。 しかし、従来のデータ保存および分析ツールにはそのリアルタイム性がありません。 たとえば、Hive データの ETL は非常に時間がかかりますが、その大半はデータの統合処理です。 強力なコンピューティング機能を搭載した Log Service は、アクセスログを数秒で処理および分析します。
- 柔軟性: Server Load Balancer インスタンスごとにアクセスログの取得を有効/無効にすることができます。Server Load Balancer インスタンスごとにアクセスログの取得を有効/無効にすることができます。 また、保存期間 (1 ~ 365 日) も設定できます。なお、Logstore のサイズはサービスの成長に合わせて自動的に拡張されます。

Log Service にレイヤー 7 Server Load Balancer のアクセスログを収集するための設定

前提条件

1. Server Load Balancer および Log Service が有効になっていること。 また、作成した #unique_97、Log Service プロジェクト、および Logstore は同一リージョンであること。

    注：

   レイヤー 7 Server Load Balancer のアクセスログのみを収集することができます。 すべてのリージョンでご利用いただけます。

2. RAM ユーザーが SLB アクセスログへのアクセス権を有していること。 詳細は、「#unique_98」をご参照ください。

手順

1. Log Service コンソールにログインします。
2. プロジェクトおよび Logstore を作成後、ページの指示に従ってデータインポートウィザードを起動します。 (または、**Logstore** リストページ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. データソースを選択します。

   Cloud Services の **Server Load Balancer** 、次へを順にクリックします。
4. RAM の権限付与

   ページの指示に従って権限付与、 権限付与ポリシーの確認を順にクリックし、Server Load Balancer に Log Service へのアクセス権を付与します。
5. 送信ルールを設定します。 送信設定をクリックし、Server Load Balancer コンソールに移動します。

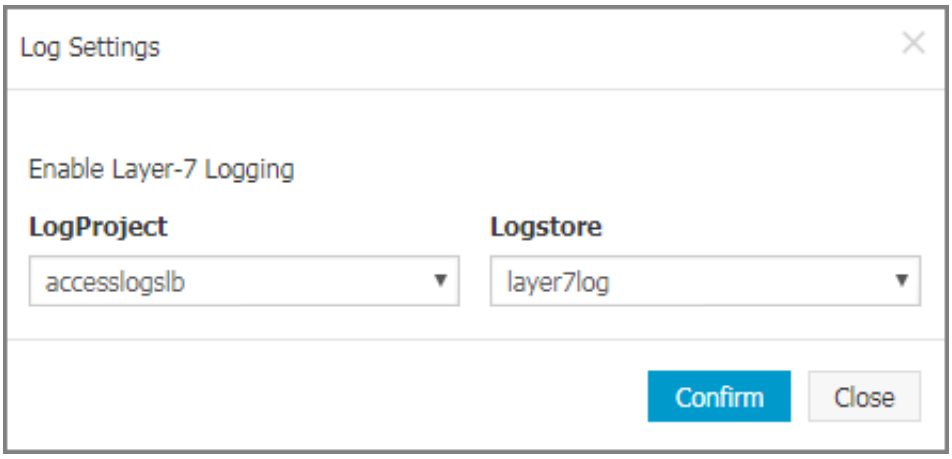
   a. 左側のナビゲーションメニューより、ログ > アクセスログを順にクリックします。

   b. Server Load Balancer インスタンスの右側の設定をクリックします。

   注 :

Log Service プロジェクトおよび SLB インスタンスが同じリージョンであること。

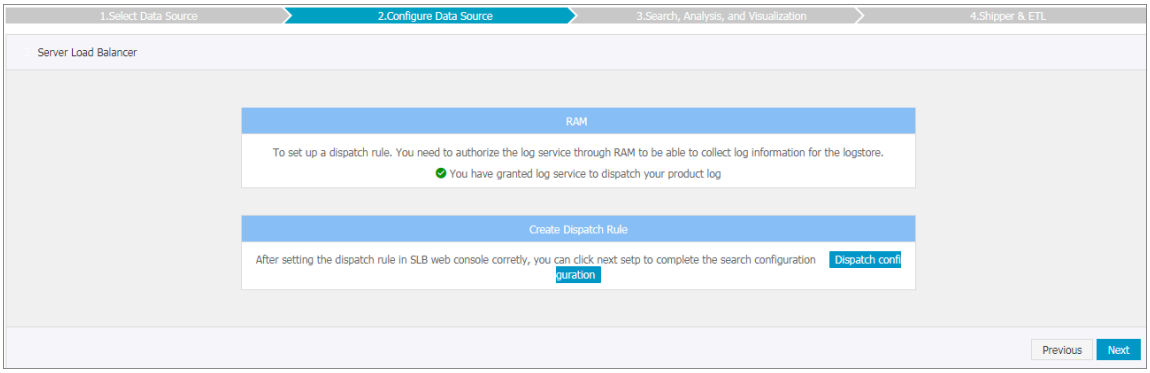
図 4-3：ログの設定



c. Log Service のプロジェクトおよび Logstore を選択し、 確認をクリックします。

d. 設定し終わったら、ダイアログボックスを閉じます。 データインポートウィザードに戻り、次へをクリックします。

図 4-4：データソースの設定



6. 照会/分析/可視化

Log Service には Server Load Balancer の照会フィールドがあらかじめ定義されています。 各フィールドの詳細は、下記「フィールドの説明」をご参照ください。 次へをクリックします。

注：

Logstore 名で始まる、{LOGSTORE}-slb_layer7_access_center ダッシュボード および {LOGSTORE}-slb_layer7_operation_center ダッシュボードが自動生成されます。 設定が完了すると、ダッシュボードページに表示されます。

7. 確認をクリックしてデータインポートウィザードを終了します。

その他の操作

- リアルタイムにログを照会

  ログ内のキーワードを使用して、迅速かつ正確なクエリまたはあいまいクエリを実行することができます。 問題の特定や、統計クエリに使用します。

- 分析レポートのテンプレート

  Server Load Balancer には、アクセス数の多いクライアント、リクエストステータスコードの分布、アクセス数の多い URI、リクエストのトラフィック量の変化、およびサーバーの応答時間の統計といった、包括的な統計グラフが事前に定義されています。

- 分析グラフの作成

  任意のログ項目に対してアドホッククエリを実行し、その実行結果をグラフとして保存し、サービスの運用管理に役立てることができます。

- ログモニタリングアラームの設定

  Server Load Balancer のアクセスログに対してカスタム分析を実行し、その実行結果をクイック照会として保存することができます。 また、クイック照会をアラーム通知として設定すると、 リアルタイムログの処理結果が、設定されているしきい値を超えた場合、システムよりアラーム通知が送信されます。

フィールド説明

フィールド	説明
body_bytes_sent	クライアントに送信された HTTP 本文のサイズ (単位: byte)
client_ip	リクエストを送信したクライアントの IP アドレス
host	リクエストパラメータの host 値 (リクエストパラメータに host 値がない場合は、host ヘッダーの host 値。 host ヘッダーにも host 値がない場合は、リクエストのバックエンドサーバーの IP アドレスが host 値)
http_host	リクエストの host ヘッダーのコンテンツ
http_referer	プロキシの受信したリクエストの HTTP リファラーヘッダーのコンテンツ
http_user_agent	プロキシの受信したリクエストの HTTP user-agent ヘッダーのコンテンツ

フィールド	説明
http_x_forwarded_for	プロキシの受信したリクエストの x-forwarded-for のコンテンツ
http_x_real_ip	送信元 IP アドレス
read_request_time	プロキシのリクエスト読み込み時間 (単位: ミリ秒)
request_length	startline、HTTP ヘッダー、および HTTP 本文を含めたリクエストメッセージの長さ
request_method	リクエストメソッド
Request_time	プロキシが最初のリクエストを受信してからレスポンスを返すまでの時間 (単位: 秒)
request_uri	プロキシの受信したリクエストの URI
scheme	リクエストスキーマ（http または https）
server_protocol	プロキシの受信した HTTP プロトコルのバージョン (例: HTTP/1.0 または HTTP/1.1)
slb_vport	Server Load Balancer のリスニングポート
slbid	Server Load Balancer インスタンスの ID
ssl_cipher	暗号スイート (例: ECDHE-RSA-AES128-GCM-SHA256/)
ssl_protocol	SSL/TSL 接続のプロトコル (例: TLS v1.2)
status	メッセージに対するプロキシのレスポンスステータス
tcpinfo_rtt	クライアントの tcp RTT (単位: マイクロ秒)
time	ログが書き込まれた時間
Upstream_addr	バックエンドサーバーの IP アドレスおよびポート
upstream_response_time	Server Load Balancer のバックエンドサーバーへの接続確立から、データを送信し、接続が切断されるまでの時間 (単位: 秒)
upstream_statu	プロキシの受信したバックエンドサーバーのレスポンスステータスコード
vip_addr	VIP アドレス
write_response_time	プロキシのレスポンス書き込み時間 (ミリ秒)

4.4 DDoS ログ収集

4.4.1 概要

Alibaba Cloud Anti-DDoS Pro は、インターネットサーバー（Alibaba Cloud 以外のホストを含む）向けの有料サービスです。 大トラフィック DDoS 攻撃後にサービスが利用できなくなるリスクを回避するために、有料サービスを適用できます。 Anti-DDoS Pro を構成し、攻撃トラフィックをハイプロテクション IP へ誘導することで、送信元の安定さ、及び信頼性を確保します。

背景情報

インターネットのセキュリティは常に課題に直面しています。 DDoS 攻撃を始めとしてのネットワークの脅威は、ネットワークセキュリティに深刻な影響を及ぼします。

DDoS 攻撃は、大規模化、モバイル化、及びグローバル化の方向に向かっています。 最近の調査報告によると、DDoS 攻撃の頻度は増加傾向です。 ハッカーの攻撃は隠密性が高く、セキュリティ対策が不十分なクラウドサービスプロバイダをコントロールして、IDC、そして大量なカメラに攻撃を仕掛けることができます。 その攻撃は既に闇産業チェーンに形成し、より組織的になってきています。 同時に、攻撃モードは極性化し、スロー攻撃、混合攻撃、特に CC 攻撃の割合が増加するため、防御の検出がより困難になります。 1Tbps を超える攻撃のピーク値がもはや一般的であり、100GB の攻撃の回数は倍で成長していく。 一方、アプリケーション層の攻撃も大幅に増加しています。

Kaspersky 2018Q1 DDoS リスクレポートによると、中国は依然として DDoS 攻撃の主な標的である。 攻撃を受けている主な産業は、インターネット、ゲーム、ソフトウェア、そして金融会社です。 DDoS 攻撃の 80% 以上が HTTP と CC 攻撃を組み合わせており、高レベルの隠密性を持っています。 そのため、ログを使用してアクセスと攻撃の動作を分析し、保護戦略を適用することが特に重要である。

Log Service は、Alibaba Cloud Anti-DDoS Pro の Web サイトアクセスログ、CC 攻撃ログのリアルタイム収集をサポートし、収集されたログデータのリアルタイムクエリと分析をサポートします。 クエリの結果はダッシュボードの形式で表示されます。

利点

- 簡単な構成：リアルタイムで保護されたログを取得するように簡単に構成できます。
- リアルタイム分析：ログサービスと連動することで、リアルタイムのログ分析とすぐに使用可能なレポートセンターを提供し、CC 攻撃のステータスとカスタマーアクセスの詳細に関する情報を提供します。

- リアルタイムアラーム：特定のインジケータに基づいたカスタムモニタリングとアラームをリアルタイムでサポートし、重要なビジネス例外にタイムリーに対応します。
- エコシステム：ストリームコンピューティング、クラウドストレージ、さらにデータ価値を探求するための視覚化ソリューションなど、他のエコシステムのドッキングをサポートします。
- 無料利用クォータ：無料のデータインポートクォータ、3 日間の無料ログ保存、クエリ、およびリアルタイム分析を提供します。 コンプライアンス管理、トレース、およびファイリングのために保管期間を自由に拡張できます。 無制限の保管時間をサポートし、保管コストは1ヶ月あたり0.35 USD / GBとなります。

制限と説明

- 専用ログストアは追加データの書込みをサポートしません。

  専用ログストアは Anti-DDoS Pro Web サイトのログの保存のみに使用されるため、他のデータの書き込みはサポートされていません。 クエリ、統計、アラーム、ストリーミングの消費など、他の機能に対する制限はありません。
- 従量課金。**DDoS** ログ収集保護が有効になっていない場合、料金は発生しません。

  DDOS ログ収集機能は、Log Service の請求項目に従って課金されます。 DDoS ログ収集保護が有効になっていない場合、料金は発生しません。 Log Service は従量課金をサポートし、無料利用クォータも提供しています。 詳細は、#unique_101をご参照ください。

シナリオ

- **Web**サイトアクセス例外のトラブルシューティング

  Log Service は DDoS ログを収集するように構成されると、収集したログをリアルタイムでクエリおよび分析できます。 SQL ステートメントを使用して DDoS アクセスログを分析するこ

とで、Web サイトのアクセス例外をすばやく確認して分析し、読み書きの遅延やキャリアの分布などの情報を表示できます。

たとえば、次のステートメントを使用して DDoS アクセスログを表示します。

```
__topic__: ddos_access_log
```

- **CC** 攻撃元の追跡

  CC 攻撃の分布と発生源は DDoS アクセスログに記録されます。 DDoS アクセスログに対してリアルタイムのクエリと分析を実行することで、ソースの追跡、CC 攻撃の追跡、および応答戦略の参照を提供できます。

  たとえば、DDoS アクセスログに記録されている CC 攻撃の国別分布を次の文で分析します。

```
__topic__: ddos_access_log and cc_blocks > 0| SELECT ip_to_country(if(real_client_ip ='-', remote_addr, real_client_ip)) as country, count(1) as "攻撃回数" group by country
```

- たとえば、次の文で PV アクセスを表示します。

```
__topic__: ddos_access_log | select count(1) as PV
```

- ウェブサイト運営分析

  DDoS アクセスログはリアルタイムでウェブサイトのアクセスデータを記録します。 収集したアクセスログデータの SQL クエリ分析を実行して、Web サイトの人気度、アクセスの発信元とチャネル、クライアントの分布などのリアルタイムのアクセス状況を取得し、Web サイトの運営分析をサポートできます。

  たとえば、複数のネットワーククラウドからの訪問者トラフィック分布を表示します。

```
__topic__: ddos_access_log | select ip_to_provider(if(real_client_ip='-', remote_addr, real_client_ip)) as provider, round(sum(request_length)/1024.0/1024.0, 3) as mb_in group by provider having ip_to_provider(if(real_client_ip='-', remote_addr, real_client_ip)) <> '' order by mb_in desc limit 10
```

4.4.2 収集手順

Anti-DDoS Pro コンソールで、Web サイトの DDos ログ収集機能を有効にできます。

1. Anti-DDoS Pro ログ収集機能を有効にし、Anti-DDoS Pro インスタンスを購入してから、オンライン構成を行います。
2. Anti-DDoS Pro ログ収集機能を有効にし、Anti-DDoS Pro インスタンスを購入します。
3. Log Service を有効化します。

Log Service は、**Alibaba Cloud Anti-DDoS Pro** Web サイトのアクセスログ、CC 攻撃ログのリアルタイム収集をサポートし、収集されたログデータのリアルタイムクエリと分析をサポートし

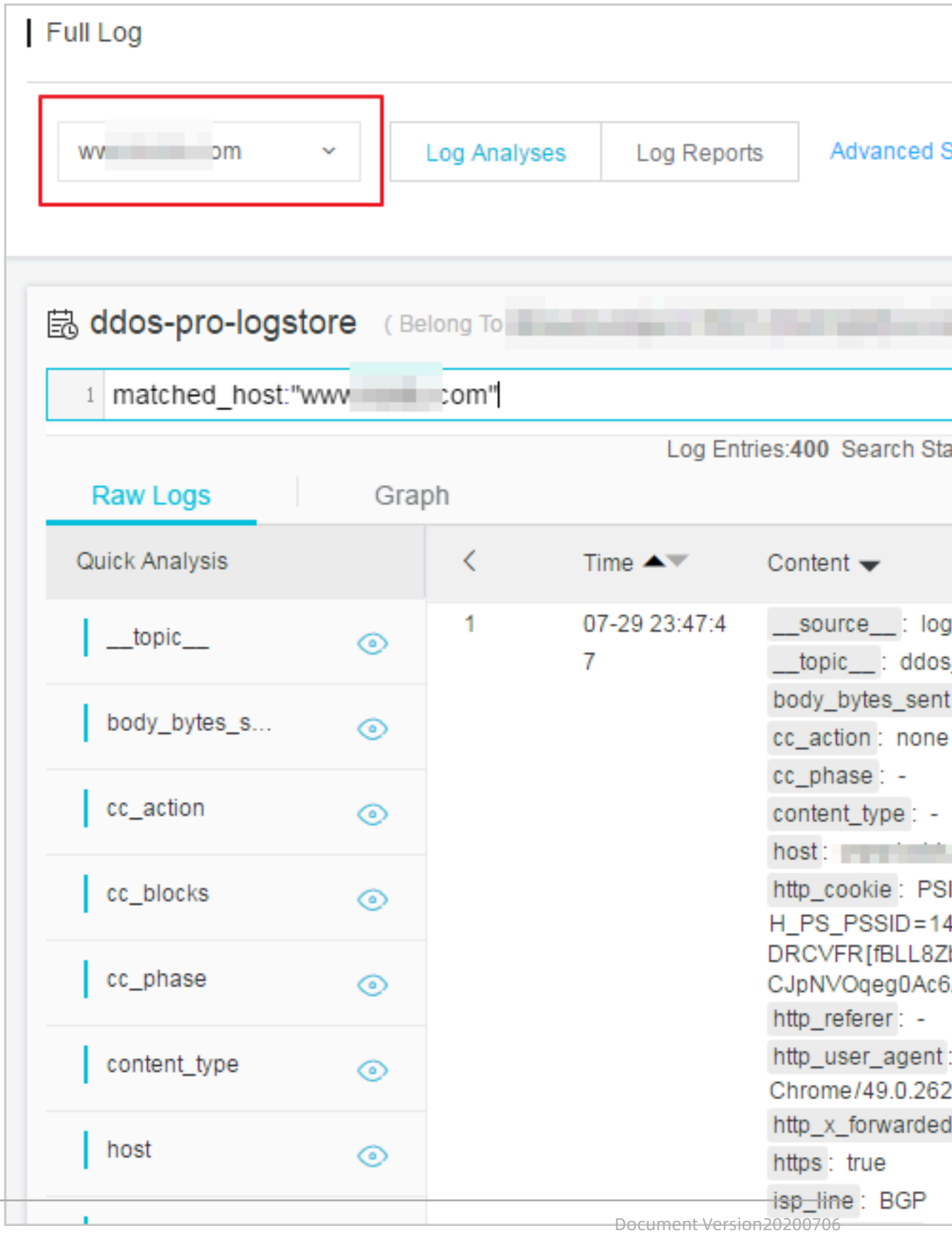
ます。 クエリの結果はダッシュボードの形式で表示され、ログはアクセスと攻撃の様子をリアルタイムで分析し、セキュリティ部門が保護の方針を策定するのに役立ちます。

1. Anti-DDoS Pro コンソールにログインし、左側のナビゲーションペインで **Log** > **Full Log** を選択します。 **Full Log** ページを開きます。
2. 初めて DDoS ログ収集を設定する場合は、ページの指示に従います。

   DDoS は、権限付与されると DDoS ログをログストアに配信する権限を持ちます。

3. DDoS ログ収集機能を有効にする Web サイトを選択し、ステータス がオンになっていることを確認します。

図 4-5：機能の有効化



これで、現行の Web サイトで DDoS ログ収集を有効にしました。 Log Service はご使用するアカウントの下に自動的にログストアを作成します。 DDoS は、この機能が有効になっている Web サイトのすべてのログをこのログストアにインポートします。 ログストアのデフォルト構成について、デフォルト構成をご参照ください。

表 4-1：デフォルト構成

デフォルト構成項目	構成内容
Project	デフォルトでは、ddos-pro-logstore プロジェクトが作成されています。
Logstore	デフォルトでは、ログストアが作成されています。 ログストア名は、購入した DDoS のドメインによって決まります。 • 中国本土の DDoS インスタンス：ddos-pro-project-Alibaba Cloud Account ID-cn-hangzhou • その他の DDoS インスタンス： ddos-pro-project-Alibaba Cloud Account ID-ap-southeast-1 DDoS ログ収集機能により生成されたすべてのログは、このログストアに保存されます。
リージョン	• DDoS リージョンが中国本土にある場合、デフォルトのプロジェクトは China East 1 に保存されます。 • DDoS リージョンが中国本土外にある場合、デフォルトのプロジェクトは Asia Pacific SE 1 に保存されます。
Shard	デフォルトでは、2 つのシャードが作成され、自動分割シャード 機能がオンになっています。
ログの保存期間	デフォルトの保管期間は、無料クォータ内で 3 日間です。 3 日後にログは自動的に削除されます。 保管期間を長くするには、構成をカスタマイズします。 詳細については、#unique_101セクション内の**Web** サイトログの保存期間を変更する方法を参照してください。

デフォルト構成項目	構成内容
ダッシュボード	デフォルトでは、2 つのダッシュボードが作成されます。 • **ddos-pro-logstore**_ddos_operation_center: オペレーションセンター • **ddos-pro-logstore**_ddos_access_center:アクセスセンター ダッシュボードの詳細は、ログレポート をご参照ください。

現行の **Full Log** ページで、収集したログをリアルタイムでクエリおよび分析できます。 ログフィールドの説明については、次の図を参照してください。 さらに、Log Service は DDoS オペレーションセンターとアクセスセンターの 2 つのダッシュボードを作成しています。 ダッシュボードの構成情報をカスタマイズできます。

フィールド	説明	例
__topic__	ログのトピックは ddos_access_log に限定されています。	-
body_bytes_sent	Body のサイズを送信するようリクエストします。 単位：バイト	2
content_type	コンテンツのタイプ。	application/x-www-form-urlencoded
host	ソース Web サイト。	api.zhihu.com
http_cookie	リクエスト cookie。	k1=v1;k2=v2
http_referer	リクエストリファラー ヌルの場合は、- と表示されます。	http://xyz.com
http_user_agent	ユーザーエージェントのリクエスト。	Dalvik/2.1.0 (Linux; U; Android 7.0; EDI-AL10 Build /HUAWEIEDISON-AL10)
http_x_forwarded_for	プロキシにリダイレクトされたアップストリームユーザー IP。	-
https	HTTPS リクエストであるかどうかを判断します。そのうち： • true: HTTPS リクエスト。 • false: HTTP リクエスト。	true

フィールド	説明	例
matched_host	構成がマッチングしたソース Web サイトは、汎ドメイン名である可能性があります。マッチングしていない場合は、 - と表示されます。	*.zhihu.com
real_client_ip	カスタマーの実 IP にアクセスします。 利用できない場合は、 - と表示されます。	1.2.3.4
isp_line	BGP、テレコミュニケーション、Unicom などの回線情報。	テレコミュニケーション
remote_addr	リクエストの発信元クライアント IP 。	1.2.3.4
remote_port	リクエストの発信元クライアントポート。	23713
request_length	リクエストの長さ。 単位：バイト。	123
request_method	HTTP リクエストの方式。	GET
request_time_msec	リクエスト時刻。 単位：マイクロ秒。	44
request_uri	リクエストパス。	/answers/377971214/banner
server_name	マッチングした host 名。マッチングしていない場合、default と表示されます。	api.abc.com
status	HTTP ステータスコード。	200
time	時刻。	2018-05-02T16:03:59+08:00
cc_action	none、challenge、pass、close、captcha、wait、login、n などの CC 保護ポリシー。	close

フィールド	説明	例
cc_blocks	CC 保護がブロックされているかどうかを示します。 • 1: ブロックされています。 • その他コード: Passed.	1
cc_phase	seccookie、server_ip_blacklist、static_whitelist、server_header_blacklist、server_cookie_blacklist、server_args_blacklist、qps_overmax などの CC 保護ポリシー。	server_ip_blacklist
ua_browser	ブラウザー。	ie9
ua_browser_family	ブラウザーシリーズ。	internet explorer
ua_browser_type	ブラウザータイプ。	web_browser
ua_browser_version	ブラウザーバージョン。	9.0
ua_device_type	クライアントデバイスタイプ。	computer
ua_os	クライアント OS。	windows_7
ua_os_family	クライアント OS シリーズ。	windows
upstream_addr	送信元アドレスリストを返します。形式： IP:Port。 複数のアドレスはカンマで区切ります。	1.2.3.4:443
upstream_ip	実際の返信元アドレス IP。	1.2.3.4
upstream_response_time	ソースの応答時間。 単位：秒。	0.044
upstream_status	ソースリクエストの HTTP ステータスを返します。	200
user_id	Alibaba Cloud ユーザー ID。	12345678

- 収集されたログデータでログ分析、クエリ分析の順にクリックします。
- ログレポートをクリックして組み込みダッシュボードを表示します。
- 詳細管理をクリックして Log Service コンソールに移動し、統計情報のクエリと収集、ストリームの消費、収集したログデータのアラームの設定を行います。

4.4.3 ログ分析

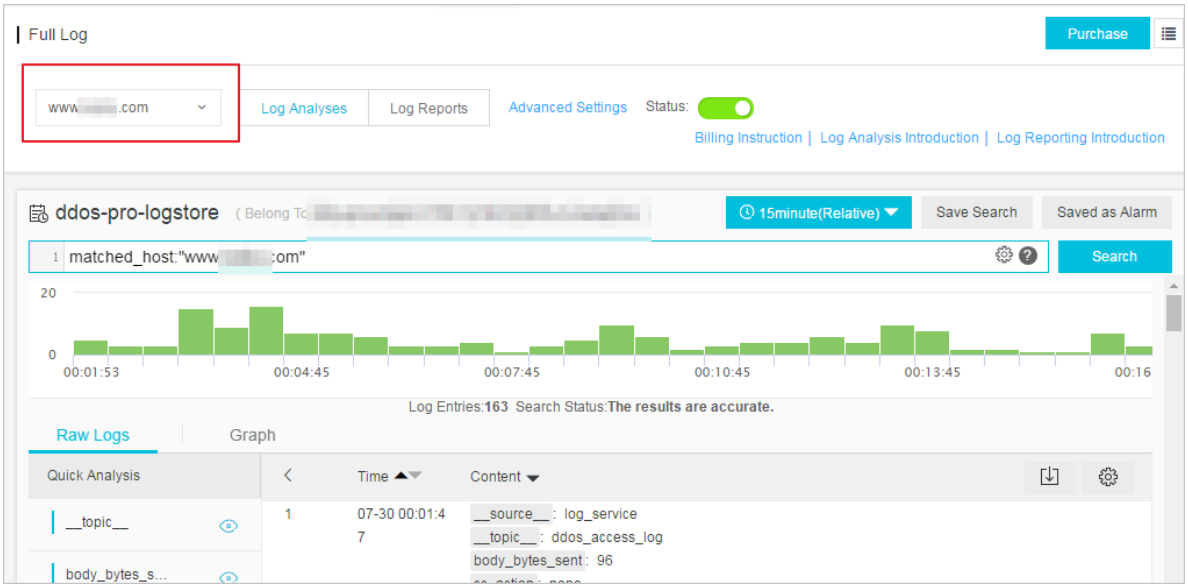
Anti-DDoS Pro は、Log Service のフルログページをログ分析とログレポートに埋め込んでいます。 特定の Web サイトに対して DDoS ログ保護機能を有効にすると、現行ページでリアルタイムに収集されたログデータを照会および分析し、ダッシュボードを表示または編集し、モニタリングアラームを設定できます。

手順

1. Anti-DDoS Pro コンソールにログインし、左側のナビゲーションペインで **Log** > **Full log**を選択します。
2. DDoS ログ収集保護を有効にする Web サイトを選択して、ステータスがオンになっていることを確認します。
3. ログ分析をクリックします。

   現行のページには Log Service のクエリ分析ページが埋め込まれており、システムは自動的に matched_host:www.aliyun.com などのクエリステートメントを入力して、選択した Web サイトに基づいてログデータを表示します。

   図 4-6：ログ分析

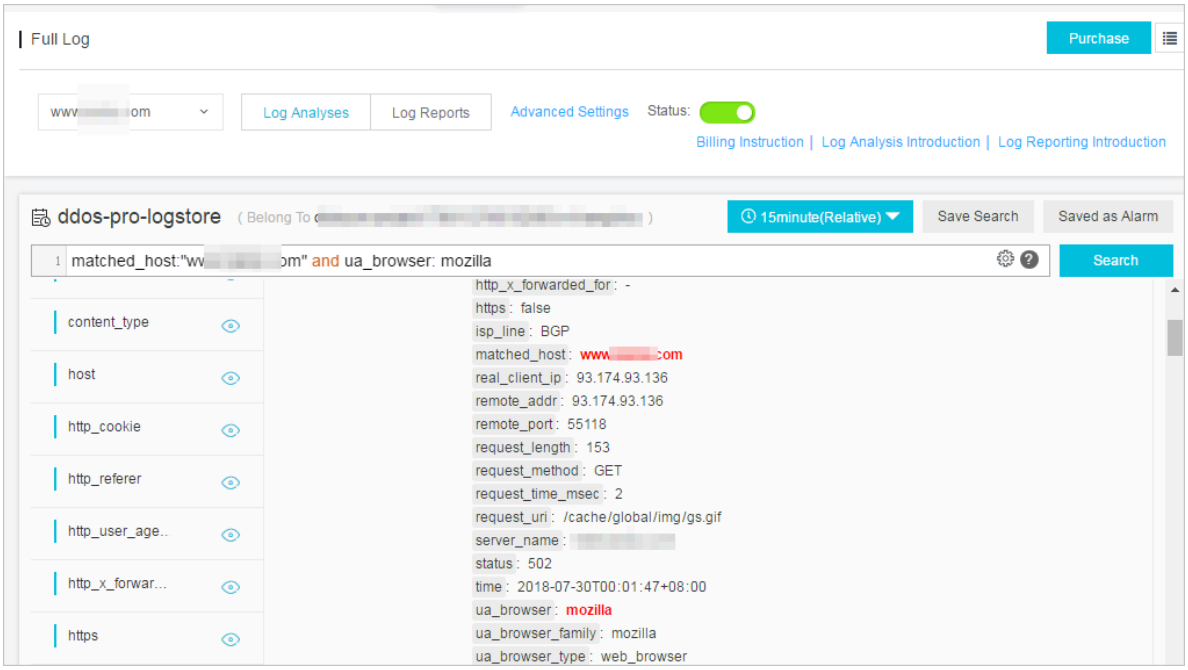


4. クエリ分析ステートメントを入力し、ログ時間範囲を選択してクエリをクリックします。

   注：

DDoS ログのデフォルトの保存期間は3日です。 3 日後、ログデータは削除されます。 デフォルトでは、過去 3 日間のログデータのみをクエリできます。 ログ保存時間を変更するには、ログ保存時間の変更をご参照ください。

図 4-7 : ログクエリ



[クエリと分析]ページでは、以下の操作も実行できます。

- カスタムクエリと分析

  Log Service は、多様な複雑なシナリオでログクエリをサポートするために、さまざまなクエリおよび分析構文を提供します。 詳細は、カスタムクエリと分析をご参照ください。

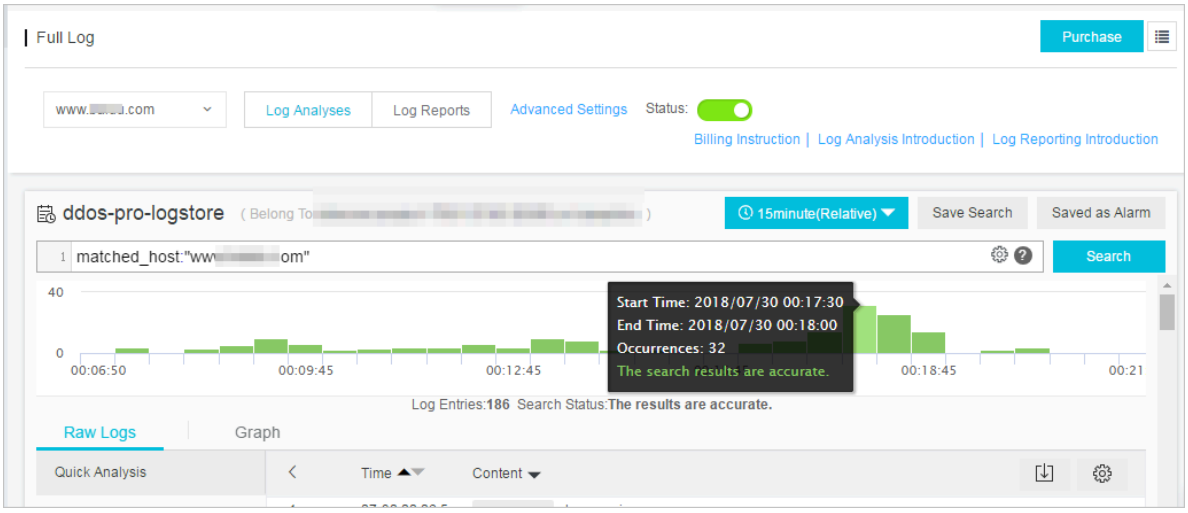
- ログ時間分布を表示する

  検索ボックスの下には、クエリ時刻とクエリステートメントに一致するログの時間分布が表示されます。 時間分布は、横軸と縦軸のヒストグラム形式で表示されます。 クエリされたログの総数が表示されます。

注 :

ヒストグラムをスライドさせて、より絞り込んだ範囲のタイムゾーンを選択すると、タイムピッカーが選択した時間範囲を自動的に更新して結果を更新します。

図 4-8：ログ時間分布を表示する



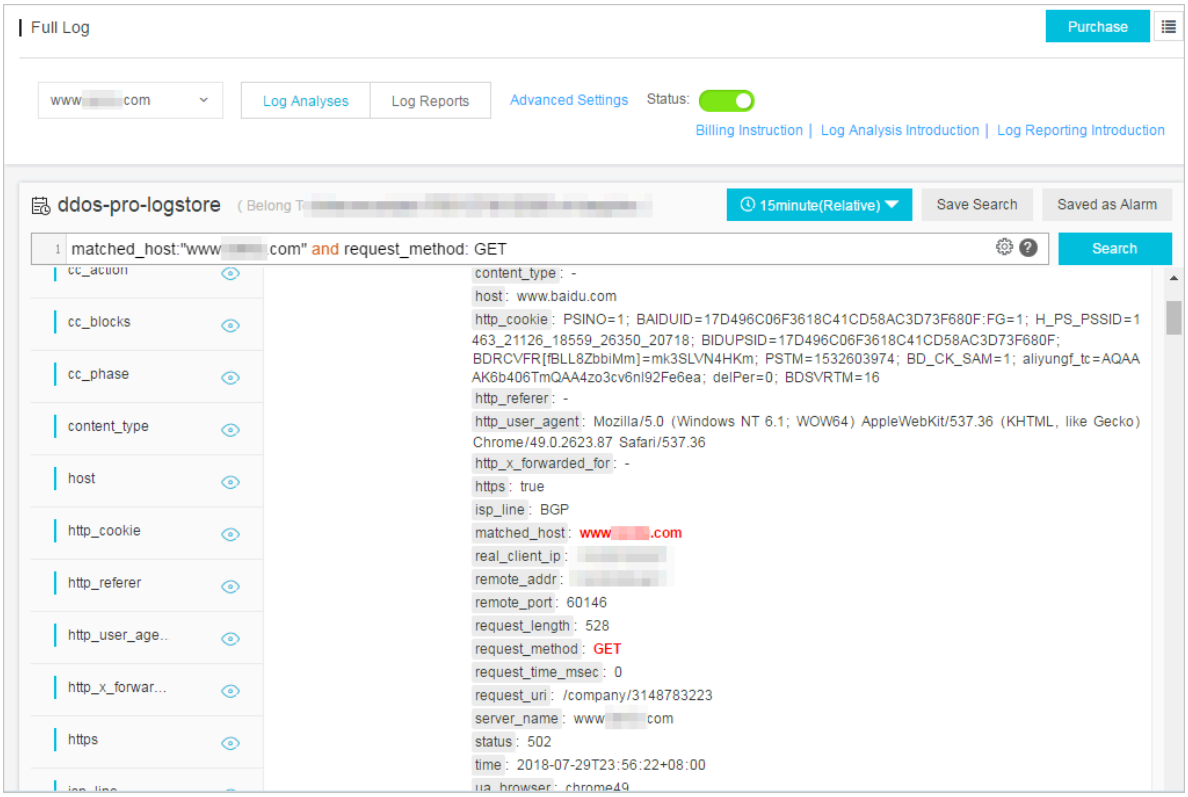
- **Raw** ログを表示する

  Raw ログでは、各ログの詳細がページ区切りで表示されます。時間、内容、及びそのうちの各フィールドも含まれます。 列の並べ替え、現行のクエリ結果のダウンロード、歯車アイコンをクリックして特定のフィールドを選択して表示することなどができます。

  ページ内の対応するフィールドの値または一部をクリックすると、検索ボックスに適切な検索条件が自動的に入力されます。 たとえば、request_method：GET で値 GET をクリックすると、次のステートメントが自動的に検索ボックスに追加されます。

```
Raw search statement and request_method: GET
```

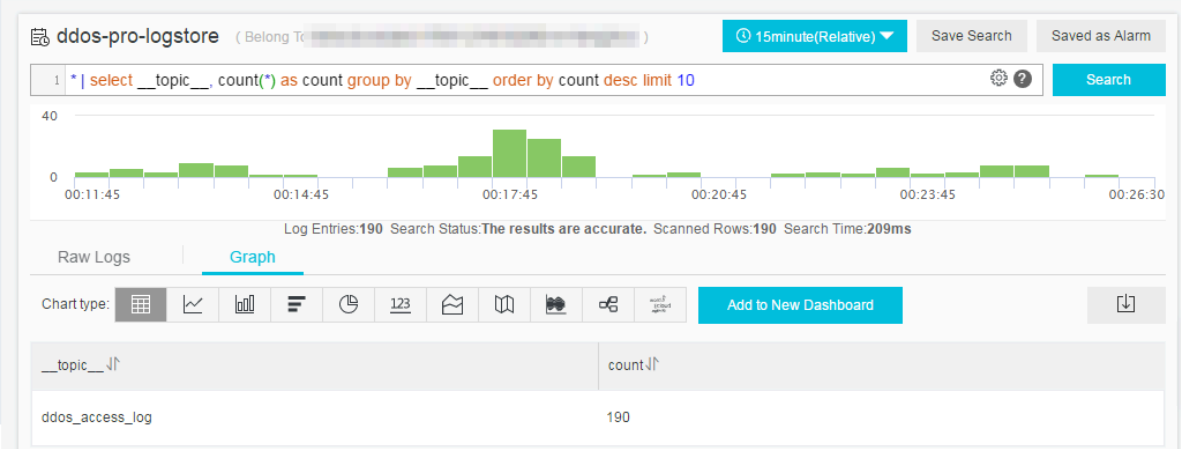
図 4-9：Raw ログ



- 分析グラフの表示

  Log Service は分析結果のグラフィック表示をサポートしています。統計グラフページでさまざまなグラフタイプを選択できます。 詳細は、分析グラフを参照してください。

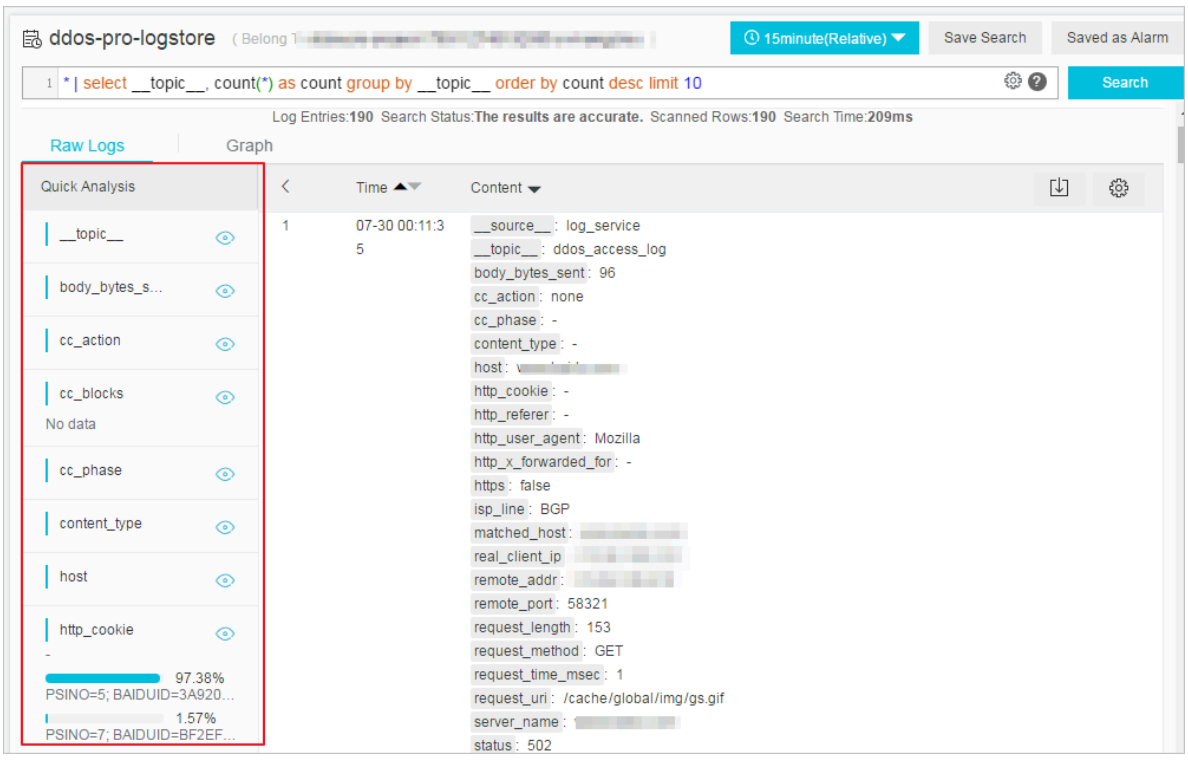
図 4-10：統計グラフ



- クイック分析

  クイック分析機能は、ワンクリックの対話式クエリを提供します。これにより、一定期間にわたるフィールドの分布を迅速に分析し、重要なデータのインデックス付けにかかる時間コストを削減できます。 詳細は、クイック分析をご参照ください。

図 4-11：クイック分析



カスタムクエリ分析

ログクエリステートメントは、クエリ構文（Search）と分析構文（Analytics）の 2 つの部分で構成され、 | で区切られています。

```
$Search | $Analytics
```

タイプ	説明
クエリ(Search)	クエリ条件は、キーワード、ファジー、数値、間隔範囲、および組み合わせ条件によって生成できます。 空白のまま、または * にすると、すべてのデータが表示されます。
分析 (Analytics)	クエリ結果または全データ量を計算し、統計します。

 注：

検索と分析は両方オプションです。 Search が空の場合、指定した期間内のすべてのデータはフィルター処理されず、結果は直接統計されます。 Analytics が空の場合、クエリ結果が返され、統計は収集されません。

クエリ構文

Log Service のクエリ構文は、フルテキストクエリとフィールドクエリをサポートしています。クエリボックスは、改行表示、構文の強調表示、及びその他の機能をサポートしています。

- フルテキストクエリ

  フィールドを指定することなく、キーワードクエリを直接入力することができます。 キーワードを二重引用符（ ""）で囲み、スペースで区切るか、複数のキーワードの間に and を挿入します。

  例

  - 複数キーワードクエリ

    www.aliyun.com と error を含むログを検索します。 例：

    ```
    www.aliyun.com error
    ```

    または

    ```
    www.aliyun.com and error
    ```

  - 条件付きクエリ

    www.aliyun.com を含み、 error または 404 を含むログを検索します。 例：

    ```
    www.aliyun.com and (error or 404)
    ```

  - プレフィックスクエリ

    www.aliyun.com を含み、failed_ で始まるすべてのキーワードを検索します。 例：

    ```
    www.aliyun.com and failed_*
    ```

     注：
    クエリはプレフィックス + * をサポートしますが、*_error のようなプレフィックスが* になる形式をサポートしません。

- フィールドクエリ

  Log Service は、フィールドに基づくより正確なクエリをサポートしています。

  数値型フィールドの比較は、 field:value または field>=value のような形式で実装でき、and または or を使用して組み合わせることができます。 また、and と or の組み合わせを使用して、フルテキスト検索と組み合わせることもできます。

  DDoS Web サイトのアクセスログと攻撃ログもフィールドクエリに基づくことが可能です。各フィールドの意味、種類、形式、およびその他の情報については、DDoS ログフィールドをご参照ください。

  例

  - 複数フィールドクエリ

    CC に攻撃された www.aliyun.com を含むログを検索します。

    ```
    matched_host: www.aliyun.com and cc_blocks: 1
    ```

    Web サイト www.aliyun.com でクライアント 1.2.3.4 のエラー 404 を含むアクセスログを検索します。

    ```
    real_client_ip: 1.2.3.4 and matched_host: www.aliyun.com and status: 404
    ```

     注 :

例 matched_host、cc_blocks、real_client_ip、および status で使用されているフィールドは、DDoS アクセスおよび攻撃ログのフィールドです。 フィールドに関する詳細情報は、 DDoS ログフィールドを参照してください。

- 数値フィールドクエリ

  応答時間が 5 秒を超えるすべてのスローリクエストログを検索します。

  ```
  request_time_msec > 5000
  ```

  間隔クエリをサポートしています。応答時間が 5 秒を超え 10 秒以下のログを検索します：

  ```
  request_time_msec in (5000 10000]
  ```

  クエリは、次のステートメントでも実行できます。

  ```
  request_time_msec > 5000 and request_time_msec <= 10000
  ```

- 日本語が使用されているかどうかを確認します。

  特定のフィールドの存在をクエリします。

  ■ ua_browser フィールドに存在するログをクエリします： ua_browser: *

  ■ ua_browser フィールドに存在しないログをクエリします： not ua_browser: *

クエリ構文の詳細は、インデックスとクエリを参照してください。

分析構文

ログデータの分析と統計には SQL / 92 構文を使用できます。 Log Service でサポートされている構文と機能の詳細は、#unique_110を参照してください。

注：

- 解析ステートメントでは、標準 SQL 構文の from table name ステートメント、つまり from logを省略できます。
- ログデータはデフォルトで最初の 100 のエントリを返します。戻り値の範囲は#unique_111を参照して変更できます。

時間ベースのログクエリ分析

各 DDoS ログには、年-月-日-T時：分：秒 + タイムゾーンの形式の time フィールドがあります。 たとえば、 2018-05-31T20:11:58+08:00 の場合、タイムゾーンは UTC+8、つまり北京の時間です。 同時に、各ログには組み込みフィールド：__time__ があります。これは、このログの時刻も示しているため、時間ベースの計算を統計で実行できます。 形式：Unix タイムスタンプ。 その

本質は、1970 年 1 月 1 日 0 時 0 分 0 秒からの累積秒数です。 そのため、実際の使用では、時刻を表示する前に、まず計算とフォーマットが必要となります。

- 時間の選択及び表示

  特定の期間にわたって、CC に攻撃されたウェブサイト www.aliyun.com の最新の 10 のログを選択し、直接 time フィールドを使用して、時間、送信元 IP およびアクセスクライアントを表示します。

```
matched_host: www.aliyun.com and cc_blocks: 1
| select time, real_client_ip, http_user_agent
    order by time desc
    limit 10
```

- 時間の計算

  CC攻撃後の日数をクエリするには、__time__ を使用して計算します：

```
matched_host: www.aliyun.com and cc_blocks: 1
| select time,
        round((to_unixtime(now()) - __time__)/86400, 1) as "days_passed", real_client_ip
, http_user_agent
     order by time desc
     limit 10
```

   注：

  round((to_unixtime(now()) - __time__)/86400, 1) を使用しています。まず to_unixtime を使用して now() より取得した時刻を Unix タイムスタンプに変換します、そして組込みの時刻フィールド __time__と減算して経過した秒数を取得します。 最後に、1 日の合計秒数である 86400 で割り、それから関数 round(data, 1) を使用して 10 進数に丸めます。 1桁の値は、各攻撃ログが何日かを過ぎたことを示します。

- 特定の時間に基づくグループ統計

  Web サイトが CC に攻撃されている毎日の様子を知りたい場合は、次の SQL を使用します。

```
matched_host: www.aliyun.com and cc_blocks: 1
| select date_trunc('day', __time__) as dt,
       count(1) as PV
    group by dt
  order by dt
```

   注：

  この例では、時間調整のために組み込みの時間フィールド __time__ を関数 date_trunc('day', ..) に渡します。 各ログは、統計の合計数（count（1））ごとに所属する日付のパーティションにまとめられ、パーティションの時間ブロックでソートされます。 関数 date_trunc の

最初の引数は、 second、miniute、hour、week、 month、year を含むその他の単位の配置を提供します。 関数の詳細は、#unique_112をご参照ください。

- 時間ベースのグループ統計

  より柔軟なグループ化時間ルールを使用するには、たとえば、5 分ごとにCCに攻撃されている Web サイトの傾向を把握するには、数学計算が必要です。 次の SQL を実行します：

```
matched_host: www.aliyun.com and cc_blocks: 1
| select from_unixtime(__time__ - __time__% 300) as dt,
       count(1) as PV
    group by dt
  order by dt
  limit 1000
```

 注：

組み込みの時間フィールドを使用して __time__ - __time __%300 を計算し、from_unixtime 関数を使用してフォーマットします。 各ログは、5分（300秒）のパーティションにまとめられて合計数を統計します（count（1））、そしてパーティションの時間ブロックでソートされ、最初の 1000 のログを取得します（選択した時間内の最初の 83 時間のデータに相当します）。

時間形式の変換など、その他の時間分析関数では、date_parse と date_format を使用する必要があります。 詳細は、#unique_112をご参照ください。

クライアント IP ベースのクエリ分析

DDoS ログにはクライアントの実 IP を取得するための real_client_ip というフィールドがあります。 ただし、ユーザーがプロキシによって実際の IP を取得できず、ヘッダー内の IP アドレスが正しくない場合は、remote_addr フィールドを使用してクライアント IP に直接接続できます。

- アタッカーの国分布

  Web 上での CC 攻撃の発信国の分布

```
matched_host: www.aliyun.com and cc_blocks: 1
| SELECT ip_to_country(if(real_client_ip='-', remote_addr, real_client_ip)) as country,
       count(1) as "number of attacks"
  group by country
```

 注：

real_client_ip フィールドまたは real_client_ip フィールド（ real_client_ip が - の場合）を選択するには、関数 if(condition, option1, option2) を使用します。 取得した IP を関数 ip_to_country に渡して、この IP に対応する国情報を取得します。

- アクセス分布

  より詳細な省ベースの分布情報を取得するには、ip_to_province 関数を使用します。例えば：

  ```
  matched_host: www.aliyun.com and cc_blocks: 1
  | SELECT ip_to_province(if(real_client_ip='-', remote_addr, real_client_ip)) as province,
        count (1) as "number of attacks"
    group by province
  ```

  
  注：

  もう一つの IP 関数 ip_to_province が IP の所属する省の情報を取得できます。 IP アドレスが中国国外の場合でも、システムは州へ変換します。

- 攻撃者の熱分布

  攻撃者のヒートマップを取得するには、ip_to_geo 関数を使用します。例えば：

  ```
  matched_host: www.aliyun.com and cc_blocks: 1
  | SELECT ip_to_geo(if(real_client_ip='-', remote_addr, real_client_ip)) as geo,
        count (1) as "number of attacks"
    group by geo
    limit 10000
  ```

  
  注：

  IP の緯度と経度を取得して最初の 10,000 を取得するには、もう一つの IP 関数 ip_to_geo を使用します。

IP演算子 ip_to_provider の取得、IP がインターネットかイントラネットの ip_to_domain かの判断など、その他の IP ベースの構文解析機能については、#unique_113をご参照ください。

4.4.4 ログレポート

ログレポートページは、ログサービスのダッシュボードに埋め込まれています。 このページには、デフォルトのダッシュボードが表示されます。 時間範囲を変更してフィルタを追加することにより、さまざまなフィルタ条件でダッシュボードデータを表示できます。

レポートの表示

1. Anti-DDoS Pro コンソールにログインし、左側のナビゲーションペインで **Log** > **Full Log** を選択します。 **Full Log** ページに入ります。
2. DDoS ログ収集機能を有効にするWebサイトを選択し、**Status** がオンになっていることを確認します。

3. ログレポートをクリックします。

   Log Service ダッシュボードページが現在のページに埋め込まれ、フィルタ条件が自動的に追加されます。 たとえば、選択された Web サイトに基づいてログレポートを表示するには、matched_host : www.aliyun.com を使用します。

図 4-12：レポートの表示

Web サイトで DDoS ログ収集機能を有効にした後、 Log Service はオペレーションセンターとアクセスセンターの 2 つの既定のレポート作成ツールを自動的に作成します。 デフォルトダッシュボードの詳細については、デフォルトダッシュボードを参照してください。

ダッシュボード	ダッシュボード名	説明
ddos-pro-logstore_ ddos_operation_center	DDoS 運用センター	有効なリクエストステータス、トラフィック、傾向、攻撃元の分布および CC によって攻撃されたトラフィック量とピークを含む、DDoS で保護された Web サイトの現在の全体的な動作ステータスを表示します。
ddos-pro-logstore_ ddos_access_center	DDoS アクセスセンター	PV / UV トレンド、帯域幅のピーク、訪問者、トラフィック、クライアントの種類、 リクエスト 、訪問した Web サイトの分布など、DDoS で保護された Web サイトの現在の全体的な運用ステータスを表示します。

図 4-13：デフォルトのダッシュボード

レポートの表示に加えて、次の操作を実行できます。

- 時間範囲を選択
- フィルタ条件の追加または編集
- グラフを見る

タイムピッカー

ダッシュボードページのすべてのグラフは、異なる期間の統計結果に基づいています。 たとえば、訪問のデフォルトの時間範囲は 1 日で、アクセス傾向は 30 日です。 現行のページのすべてのグラフを同じ時間範囲に表示するように設定するには、時間ピッカーを設定します。

1. 選択をクリックします。
2. ダイアログボックスで設定を行います。 相対時間、全体のポイント時間、またはカスタム時間を選択することができます。

> 注：
>
> - 時間範囲が変更されると、すべてのチャートの時間がこの時間範囲に変更されます。
> - 時間ピッカーは、現在のページのチャートの一時的な表示のみを提供し、システムは設定を保存しません。 次にレポートを表示すると、システムはデフォルトの時間範囲を表示します。

図 4-14：時間範囲を設定する

フィルター条件

Web サイトを選択し、ログレポートをクリックしてダッシュボードページに入ります。 選択された Web サイトに基づいてログレポートを表示するには、matched_host：www.aliyun.com などのフィルタ条件が自動的に追加されます。

フィルタ条件を設定することによって、レポートのデータ表示範囲を変更できます。

- すべてのウェブサイトの全体的なレポートを表示する

  フィルタ条件をクリアして、全体レポートライブラリ ddos-pro-logstore を表示します。

- より多くのフィルタ条件を追加する

  キーと値を設定することにより、レポートデータをフィルタリングできます。 複数のフィルター間の AND 関係がサポートされています。

  たとえば、電気通信回線によるアクセス要求の全体的な状況を表示します。

  図 4-15：フィルタ条件を追加する

  > 注：
  > isp_line は DDoS ログのフィールドで、オペレータネットワークがポートに接続していることを示します。 フィールドの詳細については、DDoS ログフィールドを参照してください。

グラフの種類

レポート表示領域には、以下のタイプを含む定義済みのレイアウトに従って複数のレポートが表示されます。 グラフの種類の詳細については、#unique_114を参照してください。

グラフの種類	説明
数字	効果的なリクエスト率や攻撃のピークなどの重要なインジケータを表示します。
ライン/エリアマップ	着信帯域幅の傾向や攻撃の遮断率など、特定の期間内の特定の重要なインジケータの傾向グラフを表示します。
地図	CC 攻撃国、アクセスホットスポットなどの訪問者と攻撃者の地理的分布を表示します。
円グラフ	攻撃を受けた Web サイトのトップ 10、クライアントタイプの分布など、情報の分布を表示します。
表	通常複数の列に分割された攻撃者のリストなどの情報を表示します。
地図	データの地理的分布を表示します。

デフォルトのダッシュボード

- オペレーションセンター

  オペレーションセンターは、有効なリクエストステータス、トラフィック、トレンド、攻撃者の分布、および CC によって攻撃されたトラフィック量とピークを含む、DDoS で保護された Web サイトの現在の全体的な運用ステータスを表示します。

グラフ	タイプ	デフォルトの時間範囲	説明	例
有効リクエストパッケージレート	単一値	1 時間（相対）	有効なリクエスト、つまり、すべてのリクエストの総数に対する非 CC 攻撃または 400 エラーリクエストの数。	95%
有効リクエストの流量	単一値	1 時間（相対）	すべてのリクエストの合計流量に対する有効リクエストの割合。	95%
受信トラフィック	単一値	1 時間（相対）	受信した有効リクエストの合計。 単位：MB。	300 MB
攻撃トラフィック	単一値	1 時間（相対）	CC 攻撃の着信トラフィックの合計。 単位：MB。	30 MB
発信トラフィック	単一値	1 時間（相対）	送信した有効トラフィックの合計。 単位：MB。	300 MB
ネットワーク in の帯域幅ピーク。	単一値	1 時間（相対）	Web サイトによってリクエスト済みの着信トラフィックレートの最高ピーク。 単位：bytes/s。	100 Bytes/s

グラフ	タイプ	デフォルトの時間範囲	説明	例
ネットワーク out の帯域幅ピーク。	単一値	1 時間（相対）	Web サイトによってリクエスト済みの送信トラフィックレートの最高ピーク。 単位：bytes/s。	100 Bytes/s
受信データパケット	単一値	1 時間（相対）	有効なリクエスト（非 CC 攻撃）に対する着信リクエストの数。 単位：個。	30, 000
攻撃データパケット	単一値	1 時間（相対）	CC 攻撃のリクエスト数の合計。 単位：個。	100
攻撃ピーク	単一値	1 時間（相対）	CC 攻撃のピーク。 単位は分あたりの数です。	100/ 分
受信帯域幅と攻撃のトレンド	二線図	1 時間（絶対）	1 分あたりの有効リクエスト数と攻撃リクエストのトラフィック帯域幅のトレンドグラフ。 単位：KB/s。	-
リクエストと傍受のトレンド	二線図	1 時間（絶対）	1 分あたりのリクエストおよび傍受された CC 攻撃リクエストの傾向図。 単位は分あたりの数です。	-

グラフ	タイプ	デフォルトの時間範囲	説明	例
有効リクエスト率のトレンド	二線図	1 時間（絶対）	1 分あたりの有効リクエスト数（非 CC 攻撃または 400 エラーリクエスト）の、全リクエストの総数に対する傾向図。	-
アクセス状況の分布トレンド	流れ図	1 時間（絶対）	1 分あたりのさまざまなリクエスト処理状況（400、304、20）の傾向図。 単位は分あたりの数です。	-
CC 攻撃分布	世界地図	1 時間（相対）	発信国における CC 攻撃数の合計。	-
CC 攻撃分布	中国地図	1 時間（相対）	発信元の省（中国）における CC 攻撃数の合計。	-
攻撃の一覧表	表	1 時間（相対）	IP、都市、ネットワーク、攻撃数、および総トラフィックを含む、最初の 100 の攻撃の攻撃者情報。	-
攻撃アクセス回線の分布	円グラフ	1 時間（相対）	telecommunications、Unicom、BGP などの DDoS 回線にアクセスした CC 攻撃の分布。	-
攻撃されたウェブサイトトップ 10	ドーナツグラフ	1 時間（相対）	攻撃されたウェブサイトトップ 10	-

- アクセスセンター

  アクセスセンターは、PV / UV のトレンドと帯域幅のピーク、訪問者、トラフィック、クライアントの種類、リクエスト、訪問した Web サイトの分布など、DDoS で保護された Web サイトの現在の全体的な運用状況を表示します。

グラフ	タイプ	デフォルト時間範囲	説明	例
PV	単一値	1 時間（相対）	リクエストの合計数。	100,000
UV	単一値	1 時間（相対）	個別のアクセスクライアントの総数	100,000
受信トラフィック	単一値	1 時間（相対）	Web サイトの受信トラフィックの合計。単位：MB。	300 MB
ネットワーク in の帯域幅ピーク。	単一値	1 時間（相対）	Web サイトによってリクエスト済みの受信トラフィックレートの最高ピーク。 単位：bytes/s。	100 Bytes/s
ネットワーク out の帯域幅ピーク。	単一値	1 時間（相対）	Web サイトによってリクエスト済みの送信トラフィックレートの最高ピーク。 単位：bytes/s。	100 Bytes/s
トラフィック帯域幅トレンド	二線図	1 時間（絶対）	1 分あたりの Web サイトの受信トラフィックと送信トラフィックの傾向図。 単位：KB/s。	-

グラフ	タイプ	デフォルト時間範囲	説明	例
リクエストと傍受のトレンド	二線図	1 時間（絶対）	1 分あたりのリクエストおよび傍受された CC 攻撃リクエストの傾向図。 単位は分あたりの数です。	-
PV/UV アクセストレンド	二線図	1 時間（絶対）	1 分あたりの PV と UV の傾向図。 単位：個。	-
訪問者分布	世界地図	1 時間（相対）	送信元国における訪問者の分布（PV）。	-
訪問者ヒートマップ	Amap	1 時間（相対）	訪問者の地理的アクセスヒートマップ。	-
受信トラフィック分布	世界地図	1 時間（相対）	送信元国での受信トラフィック分布の合計。 単位：MB。	-
受信トラフィック分布	中国地図	1 時間（相対）	発信元の省（中国）における受信トラフィックの合計。 単位：MB。	-
アクセス回線分布	ドーナツグラフ	1 時間（相対）	telecommunications、Unicom、BGP などの DDoS 回線にアクセスした訪問者の分布。	-

グラフ	タイプ	デフォルト時間範囲	説明	例
受信トラフィックネットワーク事業者の分布。	ドーナツグラフ	1 時間（相対）	訪問者がネットワーク事業者によってアクセスする受信トラフィックの分布。 例：telecommunications 、 Unicom 、 mobile connections 、 education network。 単位：MB。	-
訪問回数の最も多いクライアント	表	1 時間（相対）	IP、都市、ネットワーク、リクエスト方式の分布、着信トラフィック、不正アクセス数、傍受された CC 攻撃数など、最も訪問回数が多いクライアントのトップ 100。	-
アクセスドメイン名	ドーナツグラフ	1 時間（相対）	最も訪問回数の多いドメイン名のトップ 20。	-
Referer	表	1 時間（相対）	最もリダイレクトされた参照元 URL、ホスト、および頻度のトップ 100。	-

グラフ	タイプ	デフォルト時間範囲	説明	例
クライアントタイプの分布	ドーナツグラフ	1 時間（相対）	iPhone、iPad、Windows IE、Chrome など、最も訪問回数の多いユーザーエージェントのトップ 20。	-
リクエストコンテンツタイプの分布	ドーナツグラフ	1 時間（相対）	HTML、フォーム、JSON、ストリーミングデータなど、最もリクエストの多いコンテンツタイプのトップ 20。	-

4.5 TDSログ

4.6 WAF ログ

4.7 Anti-Bot ログ

4.8 ActionTrail のアクセスログ

4.8.1 概要

Alibaba Cloud の ActionTrail を Log Service と連動させることにより、ActionTrail のログを Log Service でリアルタイムに収集/分析することができます。ActionTrail に収集された操作ログは、リアルタイムに Log Service に送信されます。Log Service には、さまざまな機能が用意されており、ログをリアルタイムにクエリ、分析、ダッシュボード表示することができます。

効率とサービス品質の向上に、情報技術とクラウドコンピューティング技術を採用する企業は増えていますが、企業や組織のネットワーク、デバイス、データへの攻撃は止まることがありません。一般的に、攻撃者の目的は、損害を与えることにはなく、利益を得ることにあり、身を隠すことに長けています。 その結果、攻撃の発見と特定がますます困難になっています。

また、監査およびセキュリティにおける原因特定には、企業の IT およびデータリソースの操作ログは非常に重要な位置を占めています。 ネットワーク情報技術の発展に伴い、中国においては「サイバーセキュリティー法」の徹底的な実施により、各企業や組織は操作ログの保管と分析の徹底が図られています。クラウドコンピューティングでの、リソースに対して行った操作のログは非常に重要です。

ActionTrail には、クラウドアカウントのリソースに対して行った操作が記録され、そのログを照会することができます。なお、ログファイルは指定の OSS (Object Storage Service) または Log Service に保存されます。 ActionTrail に保存された操作ログをもとに、セキュリティ分析、リソース変更トラッキング、およびコンプライアンス監査を実施できます。

ActionTrail には、クラウドサービスの API 呼び出しログが収集されます (コンソール操作によってトリガーされた API 呼び出しログを含む)。正規化処理後、操作ログは JSON 形式に保存され、送信できる形になります。通常、コンソール操作または SDK 呼び出しを行うと、ActionTrail はその操作ログを 10 分以内に収集します。

ActionTrail は Log Service と連動しているため、Log Service でリアルタイムにログを収集/分析することができます。 ActionTrail の収集する操作ログは、リアルタイムに Log Service に送信されます。 Log Service には、さまざまな機能が用意されており、ログをリアルタイムにクエリ、分析、およびダッシュボード表示することができます。

利点

- 設定が容易: リアルタイムログの収集を難なく設定できます。設定手順とログフィールドの詳細については、手順をご参照ください。
- リアルタイム分析: Log Service を使用することにより、ログをリアルタイムに分析し、通知センターをすぐに使用し始めることができます。また、重要なクラウドリソースに対する操作ログの詳細データをリアルタイムに探し出すことができます。
- リアルタイムアラーム: 擬似リアルタイムモニタリングおよびアラームの指標を設定することができるため、重大なエラーに迅速に対応できるようになります。
- エコシステム: ストリーム処理、クラウドストレージ、可視化ソリューションといった他のエコシステムと連携させることで、データ価値を高めることができます。
- 無料枠: 毎月 500 MB のデータのインポートおよびストレージを無料で利用できます。なお、コンプライアンス、追跡、およびファイリングするために保管期間を延長することもできます。永久保管のサービスを月額 0.0875 GB/USD の低価格でご利用いただけます。 課金の詳細については、「課金方法」をご参照ください。

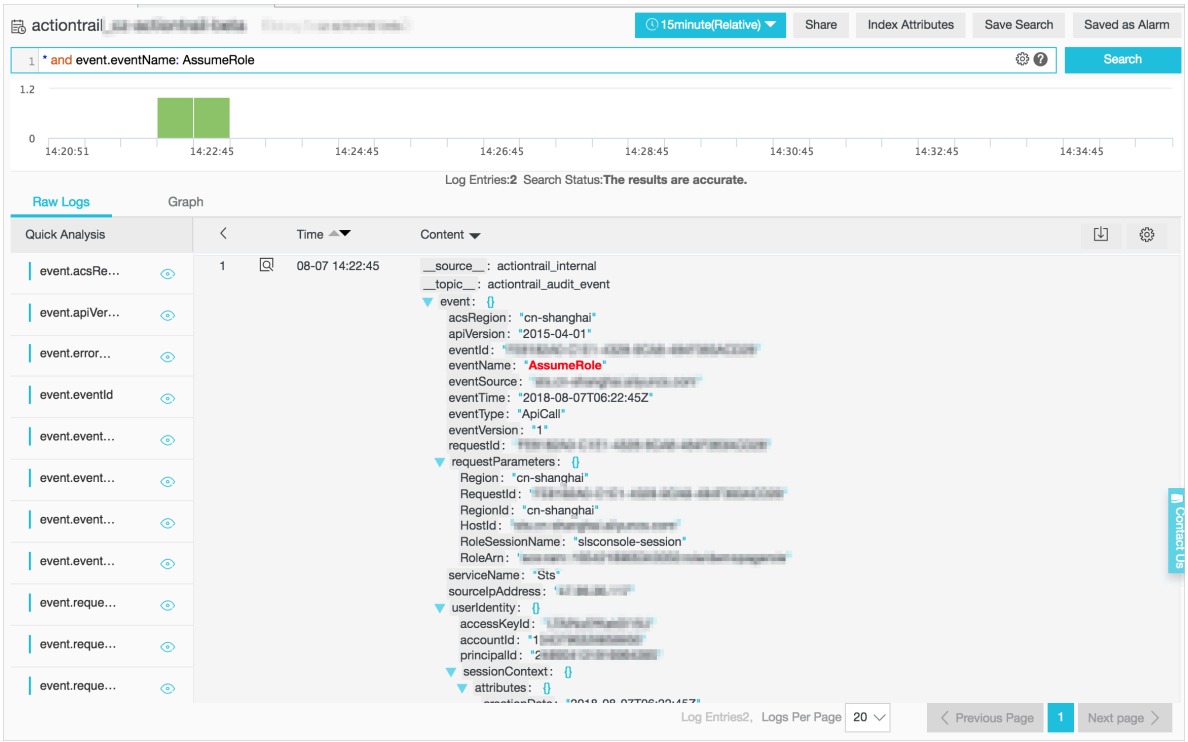
アプリケーションシナリオ

- 異常な操作のトラブルシューティングと分析

  アカウント下の全 Alibaba Cloud リソースに対する操作をモニタリングし、異常な操作を迅速にトラブルシューティングおよび分析できます。不注意による削除、リスクを伴う操作といった操作のログを残すことで、追跡できるようになります。

  例: Elastic Compute Service (ECS) リリース操作のログを表示

図 4-16 : ECS リリース操作のログを表示

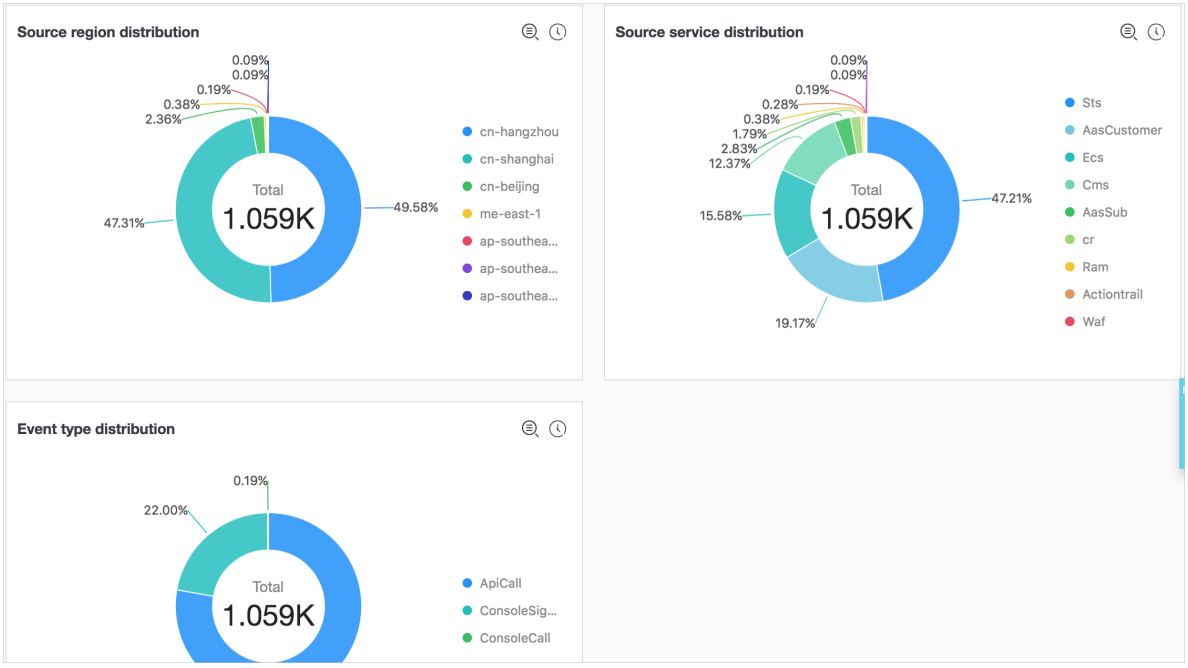


- 重要度の高いリソース操作の分布および発生源の追跡

  ログコンテンツを分析し、重要度の高いリソース操作の分布と発生源を追跡することができます。また、分析結果から解決方法を特定し、最適化することができます。

  例: リレーショナルデータベースサービス (RDS) を削除した事業者の国分布を表示

  図 4-17 : RDS 削除の分布を表示



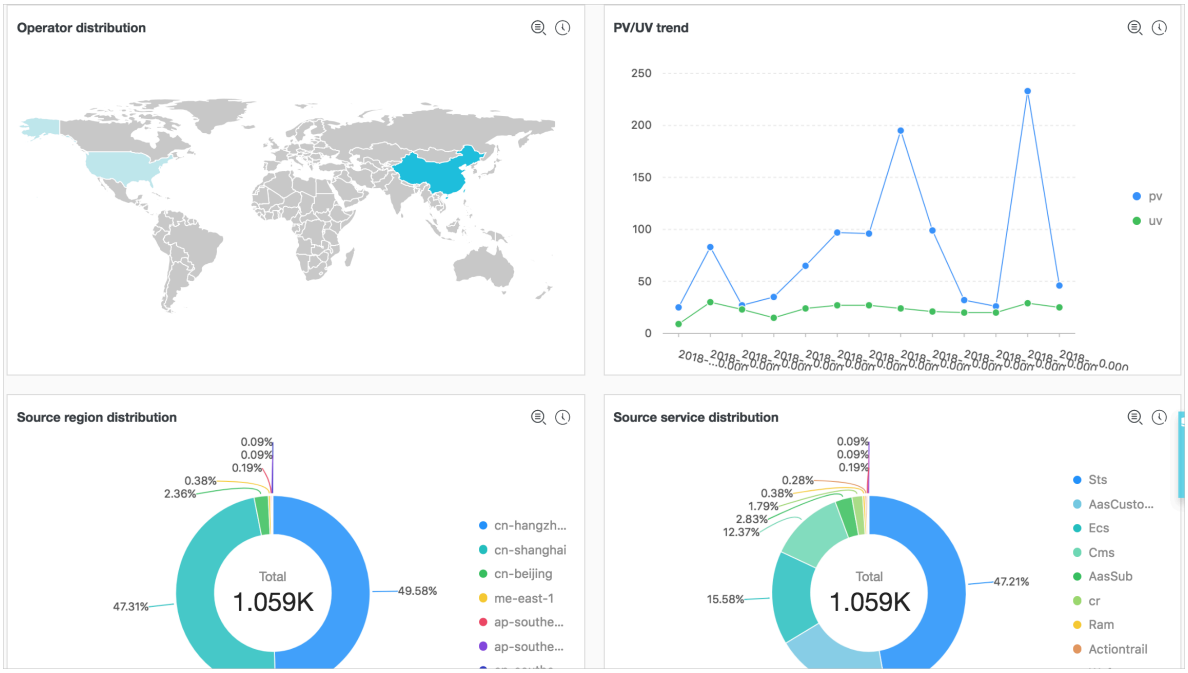
- リソース操作の分布を表示

  収集した ActionTrail 操作ログを SQL クエリステートメントでリアルタイムに照会/分析し、リソースに対するすべての操作、運用保守操作の分布と傾向を時系列に表示できます。管理者

はリソースの稼働状況をリアルタイムにモニタリングできるようになります。運用保守の信頼性がひと目でわかるようになります。

例: 失敗操作の分布を表示

図 4-18 : 失敗操作の分布



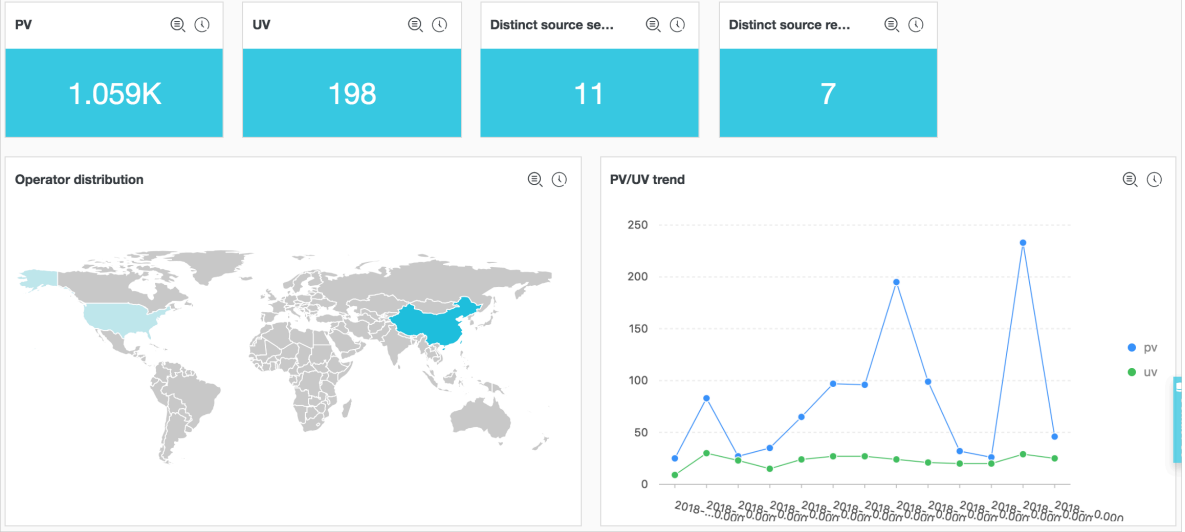
- 操作ログのリアルタイム分析

  操作ごとにクエリステートメントをさまざまにカスタマイズ、データごとにクイック検索/分析ダッシュボードを作成することができます。また、リソース使用状況やユーザーのログイン

ステータスといったデータをリアルタイムに表示するダッシュボードを作成することもできます。

例: アクセスのインターネットキャリア分布を表示

図 4-19 : アクセスのインターネットキャリア分布



4.8.2 手順

ActionTrail は Log Service と連動させることができます。 ActionTrail で収集された操作ログデータは、リアルタイムに Log Service に送信されます。 本ドキュメントは ActionTrail ログのログフィールドと収集手順を紹介します。

前提要件

1. Log Service を有効化していること。
2. ActionTrail を有効化していること。

手順

1. ActionTrail コンソールにログインします。
2. 左側のナビゲーションペインでトレイルリストをクリックし、トレイルリストページに移動します。
3. 右上隅のトレイルの作成をクリックしてトレイルの作成ページに移動します。

4. 各パラメータを設定します。

   a. トレイル名を入力します。

   b. 監査イベントを OSS バケットに送信します (オプション)。

      詳細は、トレイルの作成をご参照ください。

   c. Log Service のリージョンを選択します。

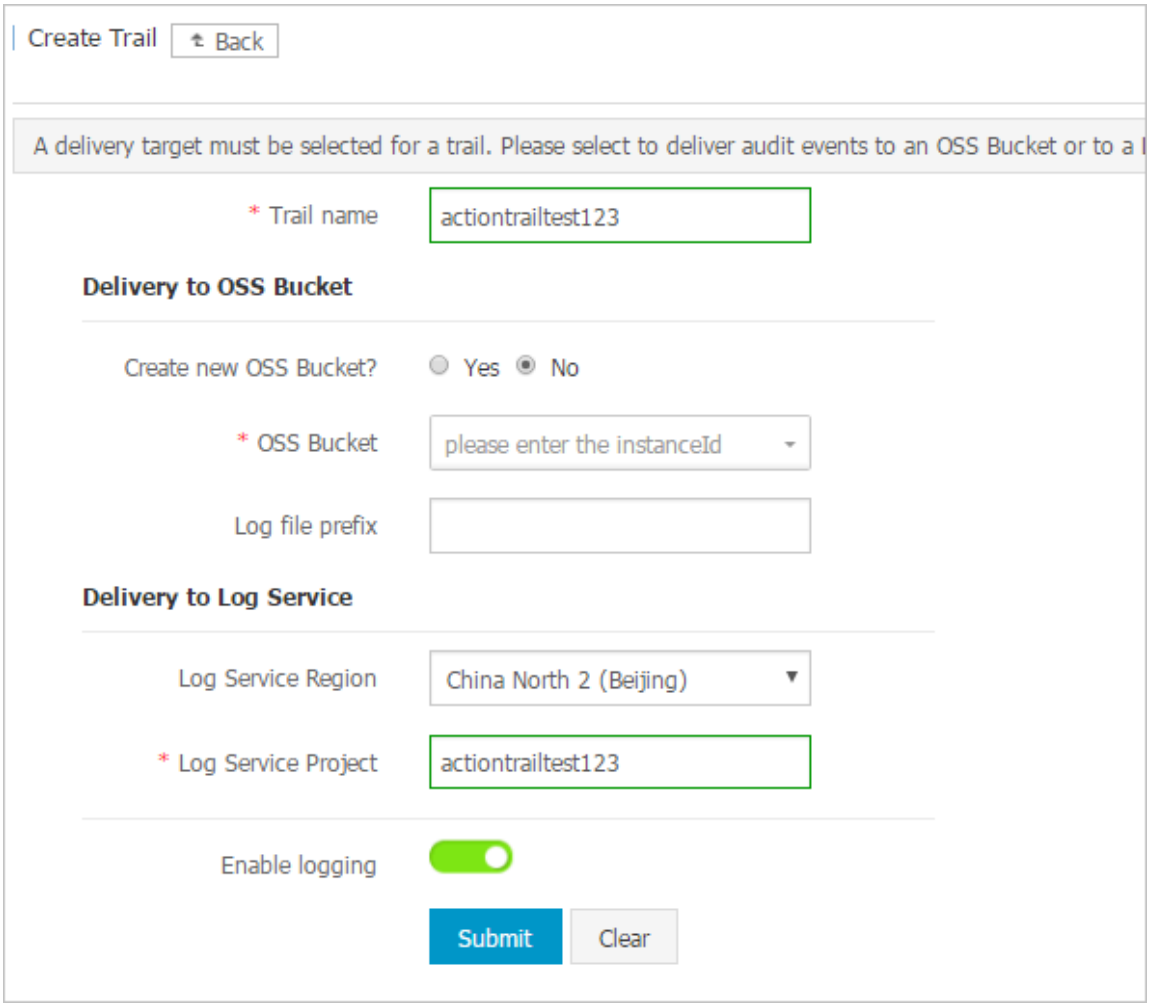
   d. **Log Service** プロジェクトを入力します。

      プロジェクトは ActionTrail ログの保存にに使用されます。 選択したリージョンに既存のプロジェクト名、または新しいプロジェクト名を入力してログを新しいプロジェクトに送信します。

   e. ログを有効にする。

      ログを有効にするをクリックします。 本機能を有効にすると、ActionTrail で記録したクラウドリソースの操作ログが Log Service に送信されます。

      図 4-20：トレイルパラメータを設定します。

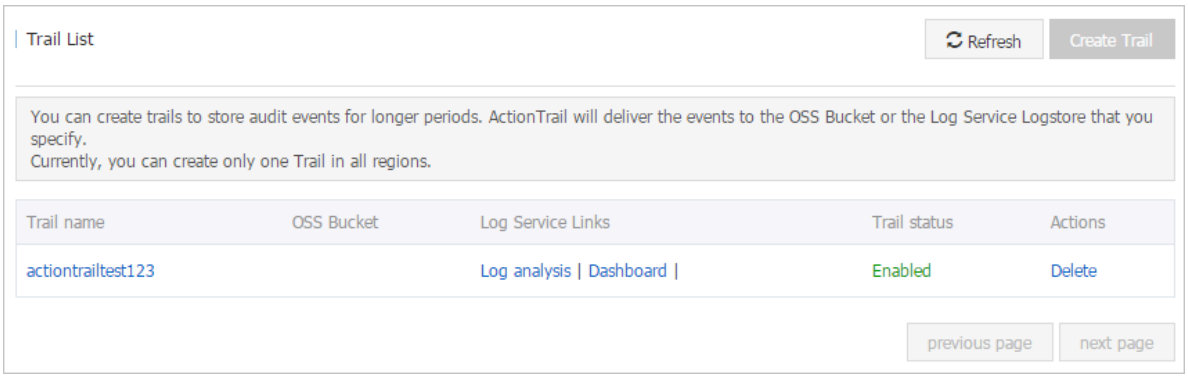


5. 送信をクリックして設定を完了します。

以上でトレイルの作成は完了です。作成したトレイルをトレイルリストに表示できます。

> 注：
> 初めて ActionTrail ログ収集を設定する場合は、ページの指示に従って ActionTrail に権限付与します。 権限付与により、ActionTrail は ActionTrail ログを Logstore に送信できるようになります。 権限付与を完了したら、もう一度送信をクリックして設定を終了します。

図 4-21：トレイルリスト



制限

- **1** つのアカウントに作成できるトレイルは **1** つのみです。

  トレイルを使用すると、指定した OSS バケットまたは Log Service Logstore に監査イベントを送信できます。 現在、すべてのリージョンのアカウントに作成できるトレイルは 1 つのみです。 このトレイルは、OSS バケットと Logstore の両方またはいずれかに、すべてのリージョンにわたって監査イベントを送信します。
- トレイルを作成した場合は、トレイルが作成されたリージョンでのみトレイルを処理できます。

  トレイルを作成した場合は、そのトレイルが作成されたリージョンでのみトレイルを表示、変更、または削除できます。 例：OSS のトレイルを作成したときに Log Service のトレイルを構成する必要がある場合は、作成した OSS のトレイルに Log Service の構成を追加します。
- 専用 **Logstore** は追加データの書き込みをサポートしません。

  専用 Logstore は、Action Trail の操作ログだけを保存するために使用されます。そのため、専用 Logstore はその他のデータの書き込みをサポートしません。 クエリ、統計、アラーム、ストリーミングの消費など、その他の機能には制限がありません。

- 従量課金。

  ActionTrail のログ収集機能は、Log Service の請求方法を使用します。 Log Service には、従量課金をサポートし、一定量の無料クォータも提供しています。 詳細は、#unique_30をご参照ください。

クエリと分析

トレイルの構成が完了後、収集されたログデータをクエリおよび分析するには、トレイルリストページの **Log Service** リストでログ分析およびログレポートをクリックします。

- ログ分析：ログクエリおよび分析ページに入ります。

  Log Service はログのクエリと分析を提供します。 このページでは、収集した ActionTrail ログをリアルタイムでクエリおよび分析できます。

  クエリ構文と分析構文を定義することで、Log Service はさまざまな複雑なシナリオでログクエリを提供します。 クエリおよび分析の構文については、クエリ構文および分析構文を参照してください。

  重要なログデータを定期的にモニタリングし、異常な状態のアラーム通知を設定するには、現在のクエリ条件をクイック検索とアラームとして検索ページに保存します。詳細手順は、#unique_120をご参照ください。
- ログレポート：ダッシュボードページに入ります。

  Log Service は、ActionTrail 専用の組み込みダッシュボードによって、イベントタイプやイベントソースなどのリアルタイムのダイナミクスの全体像を示します。

  専用ダッシュボードを変更したり、カスタムダッシュボードを作成したり、さまざまなシナリオのカスタム分析グラフをダッシュボードに追加したりできます。 ダッシュボードの詳細は、#unique_121をご参照ください。

デフォルト構成

構成が完了すると、Log Service は専用のプロジェクトと専用の Logstore を作成します。 ActionTrail で収集したクラウドリソースの操作ログは、リアルタイムで Logstore に送信されます。 さらに、Log Service は、クラウドリソースの運用状況をリアルタイムで表示するためのダッシュボードも作成します。 プロジェクトや Logstore などのデフォルト構成については、次の表をご参照ください。

表 4-2：デフォルト構成

デフォルト構成項目	構成内容
Project	トレイルを作成するときに選択、またはカスタマイズするプロジェクト。
Logstore	Logstore はデフォルトで作成されます。 Logstore 名は actiontrail_#####となります。 ActionTrail のすべてのログはこの Logstore に保存されます。
リージョン	トレイルを作成するときに選択したリージョン。
シャード	デフォルトでは、2 つのシャードが作成され、シャード自動分割機能が有効になります。
ログの保存期間	デフォルトでは、ログは永続的に保存されます。 ログ保存期間は 1~3000 日の範囲の値にカスタマイズできます。詳細手順は、#unique_122をご参照ください。
ダッシュボード	ダッシュボードはデフォルトで作成されます： • 中国語環境：actiontrail_#####_audit_center_cn • 英語環境：actiontrail_#####_audit_center_en

ログフィールド

フィールド名	名前	例
__topic__	ログトピック。	このフィールドは actiontrail_audit_eventに限定されています
event	イベントの本体。JSON形式です。 イベント本体の内容はイベントによって異なります。	イベントの例
event.eventId	イベントのID、イベントを一意に示します。	07F1234-3E1D-4BFF-AC6C-12345678
event.eventName	イベント名。	CreateVSwitch
event.eventSource	イベントの発生源。	http://account.aliyun.com:443/login/login_aliyun.htm
event.eventType	イベントの種類。	ApiCallApicall

フィールド名	名前	例
event.eventVersionEvent.eventversion	ActionTrail のデータ形式のバージョン。現在は 1 に限定されています。	1
event.acsRegion	イベントの存在するリージョン。	cn-hangzhou
event.requestId	クラウドサービス操作のリクエストID。	07F1234-3E1D-4BFF-AC6C-12345678
event.apiVersion	関連APIのバージョン。	2017-12-04
event.errorMessage	イベント失敗のエラーメッセージ。	unknown confidential
event.serviceName	イベント関連のサービス名。	Ecs
event.sourceIpAddress	イベントに関連付けられている送信元IP。	1.2.3.4
event.userAgent	イベント関連のクライアントエージェント。	Mozilla/5.0 (...)
event.requestParameters.HostId	リクエスト関連パラメータ内のホスト ID。	ecs.cn-hangzhou.aliyuncs.com
event.requestParameters.Name	リクエスト関連パラメータの名前。	ecs-test
event.requestParameters.Region	リクエスト関連パラメータ内のドメイン。	cn-hangzhou
event.userIdentity.accessKeyId	リクエストで使用された AccessKey ID。	25 **********
event.userIdentity.accountId	リクエストされたアカウントの ID。	123456
event.userIdentity.principalId	リクエストされたアカウントのバウチャー ID。	123456
event.userIdentity.type	リクエストされたアカウントのタイプ。	root-account
event.userIdentity.userName	リクエストされたアカウントの名前。	root

イベントの例

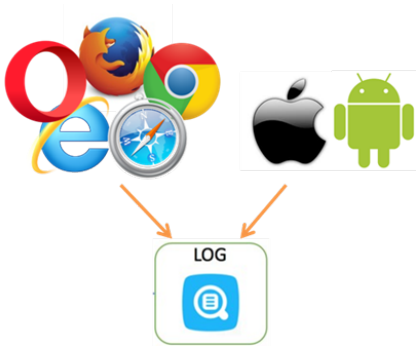
```
{
  "acsRegion": "cn-hangzhou",
  "additionalEventData": {
```

```
    "isMFAChecked": "false",
    "loginAccount": "test1234@aliyun.com"
  },
  "eventId": "7be1e173-1234-44a1-b135-1234",
  "eventName": "ConsoleSignin",
  "eventSource": "http://account.aliyun.com:443/login/login_aliyun.htm",
  "eventTime": "2018-07-12T06:14:50Z",
  "eventType": "ConsoleSignin",
  "eventVersion": "1",
  "requestId": "7be1e173-1234-44a1-b135-1234",
  "serviceName": "AasCustomer",
  "sourceIpAddress": "42.120.75.137",
  "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (
KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",
  "userIdentity": {
    "accessKeyId": "25**********",
    "accountId": "1234",
    "principalId": "1234",
    "type": "root-account",
    "userName": "root"
  }
}
```

5 その他の収集方法

5.1 Web トラッキング

Web トラッキングを使用することにより、HTML、H5、iOS、または Android プラットフォームからデータを収集し、Log Service でディメンションおよび指標をカスタマイズできます。



上図に示されているように、Web トラッキング機能を使用することにより、あらゆるブラウザ、iOS アプリ、および Android アプリよりユーザー情報を収集できます (iOS / Android SDK を除く)。 たとえば、

- ユーザーの使用しているブラウザ、オペレーティングシステム、および解像度
- ユーザーの閲覧行動パターン (ユーザーのクリック行動や Web サイトの購入行動など)
- ユーザーのアプリケーション使用時間、アクティブ/非アクティブ

> 注：
> Web トラッキングを使用すると、インターネットの匿名による Logstore への書き込みが可能になります。不正なデータが生成される危険性があります。

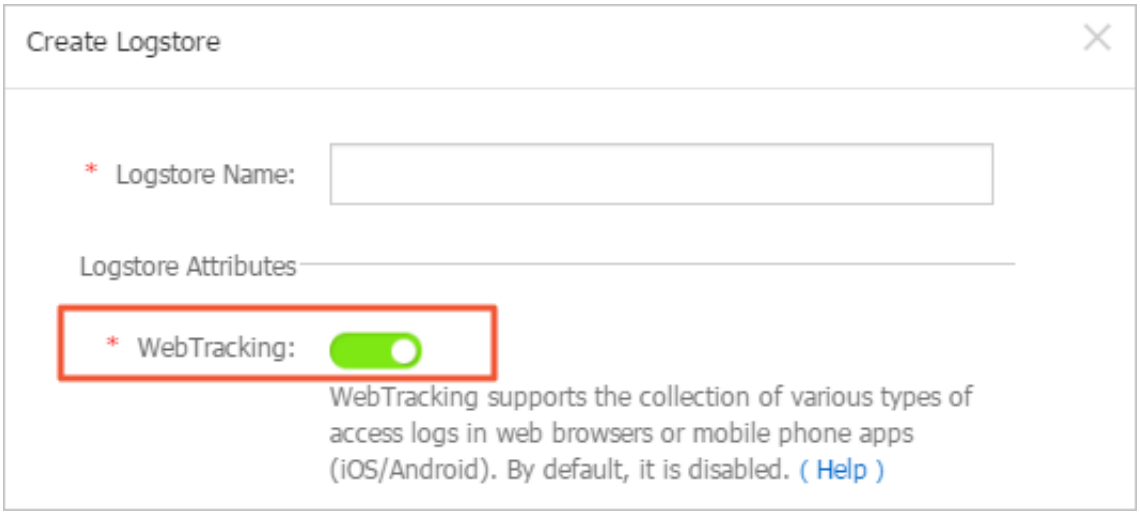
手順

ステップ1 Web トラッキングを有効化

Web トラッキングは、コンソールまたは Java SDK より有効にします。

- コンソールより **Web** トラッキングを有効化

  1. **Logstore** リストページで、 Web トラッキング機能を有効にする Logstore 名の右側の変更をクリックします。
  2. Webトラッキングをオンにします。



- **Java SDK** より **Web** トラッキングを有効化

  Java SDK

```
import com.aliyun.openservices.log.Client;
import com.aliyun.openservices.log.common.LogStore;
import com.aliyun.openservices.log.exception.LogException;
public class WebTracking {
  static private String accessId = "your accesskey id";
  static private String accessKey = "your accesskey";
  static private String project = "your project";
  static private String host = "log service data address";
  static private String logStore = "your logstore";
  static private Client client = new Client(host, accessId, accessKey);
  public static void main(String[] args) {
     try {
        //作成した Logstore の Web Tracking 機能を有効化
        LogStore logSt = client.GetLogStore(project, logStore). GetLogStore();
        client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.
GetShardCount(), true));
        //Web Tracking 機能を無効化
        //client.UpdateLogStore(project, new LogStore(logStore, logSt.GetTtl(), logSt.
GetShardCount(), false));
        //Web Tracking 機能を有効にした Logstore を生成
        //client.UpdateLogStore(project, new LogStore(logStore, 1, 1, true));
     }
     catch (LogException e){
        e.printStackTrace();
     }
  }
```

```
}
```

ステップ2 ログを収集

Logstore の Web トラッキング機能を有効にしたら、次の 3 つのいずれかの方法で、データを Logstore にアップロードします。

- **HTTP GET** リクエスト

```
curl --request GET 'http://${project}.${host}/logstores/${logstore}/track? APIVersion=0.6.0&key1=val1&key2=val2'
```

パラメータの定義

フィールド	定義
$ {project}	Log Service に作成したプロジェクトの名前
${host}	Log Service のデプロイされているリージョンのドメイン名
${logstore}	${project}下で Web トラッキング機能が有効になっている Logstore の名前
APIVersion=0.6.0	予約フィールド (必須)
__topic__=yourtopic	ログトピックを指定します (オプションの予約フィールド)
key1=val1, key2=val2	Logstore にアップロードされるキーと値のペア (複数ペア指定可、URL は 16 KB 未満)

- **HTML** の **img** タグ

```
<img src='http://${project}.${host}/logstores/${logstore}/track.gif? APIVersion=0.6.0&key1=val1&key2=val2'/>
<img src='http://${project}.${host}/logstores/${logstore}/track_ua.gif? APIVersion=0.6.0&key1=val1&key2=val2'/>
```

パラメータの定義は、 HTTP GET リクエストと同様です。track_ua.gif は、カスタムパラメータをアップロードするだけでなく、HTTP ヘッダーの UserAgent および referer をサーバーのログフィールドとして送信します。

 注 :

HTTPS ページのリファラーを収集するには、前述の Webトラッキングのリンクが HTTPS タイプでなければなりません。

- **JS SDK**

  1. loghub-tracking.jsを webディレクトリにコピーし、次のスクリプトをページに追加します。

     こちらをクリックしてダウンロード

     ```
     <script type="text/javascript" src="loghub-tracking.js" async></script>
     ```

     注：
     ページ読み込みを中断させることのないよう、スクリプトは HTTP リクエストを非同期に送信します。ページの読み込み中に何度もデータを送信する必要がある場合、前の HTTP リクエストは後続のリクエストに上書きされ、ブラウザには追跡リクエストを終了した旨が示されます。この問題は、同期リクエストを送信することにより、防ぐことができます。同期リクエストを送信するには、スクリプトの次の文を変更します。

     元のスクリプト

     ```
     this.httpRequest_.open("GET", url, true)
     ```

     最後のパラメータを変更

     ```
     this.httpRequest_.open("GET", url, false)
     ```

  2. Tracker オブジェクトを作成します。

     ```
     var logger = new window.Tracker('${host}','${project}','${logstore}');
     logger.push('customer', 'zhangsan');
     logger.push('product', 'iphone 6s');
     logger.push('price', 5500);
     logger.logger();
     logger.push('customer', 'lisi');
     logger.push('product', 'ipod');
     logger.push('price', 3000);
     logger.logger();
     ```

     パラメータの意味は次のとおりです。

フィールド	定義
${host}	Log Service のデプロイされているリージョンのドメイン名
${project}	Log Service に作成したプロジェクトの名前

フィールド	定義
${logstore}	${project}下で Web トラッキング機能が有効になっている Logstore の名前

上記ステートメントが実行されると、Log Service に、以下の 2 つのログが表示されます。

```
customer:zhangsan
product:iphone 6s
price:5500
```

```
customer:lisi
product:ipod
price:3000
```

Log Service にデータがアップロードされたら、Log Service のLogSearch/Analyticsでリアルタイムにログデータを検索/分析し、さまざまな可視化ソリューションによって分析結果をリアルタイムに表示させることができます。 また、Log Service に用意されている LogHub クライアントライブラリ を使用して Log Service のデータを読み込むこともできます。

5.2 DataWorks を使用して MaxCompute データを Log Service に収集

シナリオ

DataWorks は Alibaba Cloud のデータ中継サービスです。DataWorks は、Log Service の収集したログファイルを MaxCompute に送信します。MaxCompute に送信されたログファイルは保存され、分析できるようになります。MaxCompute では、オフライン処理を行うことができます。オンライン分析処理 (OLAP) の必要がある場合、DataWorks より MaxCompute に送信されたログファイルをエクスポート、また、その処理結果を Log Service にエクスポートします。Log Service はエクスポートされたデータをリアルタイムに検索/分析します。

実装

LogHub Writer は、Reader の生成するデータを DataWorks を介して取得し、DataWorks で使用されたデータ型を文字列型に変換します。データ量が指定の batchSize に達すると、LogHub Writer は Log Service Java SDK を使用してすべてのデータを Log Service に一括送信します。 デフォルトでは、LogHub Writer は 1,024 件を一括送信します (最大batchSize: 4096)。

前提条件

1. Log Service を有効化し、プロジェクトおよび Logstore を作成していること。
2. MaxCompute を有効化し、テーブルを作成していること。

3. DataWorks を有効化していること。

手順

1. DataWorks コンソールにログインして LogHub データソースを作成します。

   データソースの作成方法については、「Dataworks を介した MaxCompute へのデータ転送」をご参照ください。

2. スクリプトモードで同期タスクを作成します。

   a. 左側のナビゲーションメニューより、同期タスク、スクリプトモード と順にクリックし、同期タスクを設定します。

   図 5-1 : スクリプトモード

   b. インポートテンプレートでパラメータを指定します。

   図 5-2 : インポートテンプレート

パラメータ	説明
ソースタイプ	データソースの種類は、ODPS にします。
データソース	データソースの名前。 データソース追加 をクリックしてデータソースを作成することもできます。
オブジェクトタイプ	送信するオブジェクトタイプは LogHub にします。
データソース	送信オブジェクトの名前。手順 1 で作成した LogHub データの送信先を選択するか、データソース追加 をクリックしてデータの送信先を作成します。

   確認 をクリックし、同期タスクの設定に進みます。

   c. 設定情報を入力します。

   例

```
{
  "type": "job",
  "version": "1.0",
  "configuration": {
    "setting": {
      "errorLimit": {
```

```
      "record": "0"
     },
     "speed": {
      "mbps": "1",
      "concurrent": 1,
      "dmu": 1,
      "throttle": false
     }
    },
    "reader": {
     "plugin": "odps",
     "parameter": {
           "accessKey":"*****",
        "accessId":"*****",
        "column":["*"],
        "isCompress":"false",
        "partition":["pt=20161226"],
        "project":"aliyun_account",
        "table":"ak_biz_log_detail"
     }
    },
    "writer": {
     "plugin": "loghub",
     "parameter": {
      "endpoint": "",
      "accessId": "",
      "accessKey": "",
      "project": "",
      "logstore": "",
      "batchSize": "1024",
      "topic": "",
      "time" :"time_str",
      "timeFormat":"%Y_%m_%d %H:%i:%S",
      "column": [
       "col0",
       "col1",
       "col2",
       "col3",
       "col4",
       "col5"
      ],
      "datasource": "sls"
     }
    }
   }
}
```

パラメータ	必須項目	説明
endpoint	はい	Log Service のエンドポイント。 詳細は、サービスエンドポイントをご参照ください。
accessKeyId	はい	Alibaba Cloud アカウントまたは RAM ユーザーの AccessKeyId

パラメータ	必須項目	説明
accessKeySecret	はい	Alibaba Cloud アカウントまたは RAM ユーザーの AccessKeySecret
project	はい	Log Service 内の対象プロジェクト名
logstore	はい	Log Service 内の対象 Logstore 名
topic	いいえ	Log Service のトピックフィールドとして指定する MaxCompute のフィールド (デフォルトは空の文字列)
batchSize	いいえ	LogHub Writer が一括送信するエントリ数 (デフォルトは 1024)
column	はい	各エントリのカラム名 注 : **column** パラメータにカラムを指定しない場合、カラムはダーティーデータ (不要データ) と見なされます。
time	いいえ	time フィールドの名前 注 : time フィールドを指定しない場合は、システム時間になります。

パラメータ	必須項目	説明
timeFormat	time フィールドを指定した場合、timeFormat の指定は必須です。	timeFormat の書式: • bigint: unix のタイムスタンプ • timestamp: 文字列で取得された時間 (例：%Y_%m_%d %H:%M:%S) bigint 型の time フィールドの値が「1529382552」の場合、timeFormat フィールドは bigint になります。 time フィールドが文字列「2018_06_19 12:30:25」の場合、timeFormat フィールドは %Y_%m_%d %H:%M:%Sになります。
datasource	はい	DataWorks で定義されているデータ型

3. タスクを保存し、実行します。

   保存 をクリックして、この同期タスクの保存先のパスを指定します。 このタスクをそのまま実行することも、スケジュールシステムに送信することもできます。

   図 5-3：同期タスクの実行

   - タスクの実行

     実行をクリックして、すべてのデータの同期を開始します。
   - タスクのスケジュール

     送信をクリックして、タスクをスケジューリングシステムに送信します。 タスクを受信したスケジューリングシステムは、設定に従って自動的に実行します。

     注：

スケジューリングサイクルはパーティション生成サイクルと同じ設定にすることを推奨します。 たとえば、1 時間ごとにデータ収集してパーティション生成する場合、スケジューリングサイクルも 1 時間にします。

タスクのスケジュール方法については、「DataWorks を介した MaxCompute へのデータ転送」をご参照ください。

データ型

DataWorks を使用して MaxCompute データを Log Service にインポートすると、下表のとおり、すべてのデータ型は文字列型に変換されます。

MaxCompute データ型	**LogHub** にインポート後のデータ型
Long	String
Double	String
String	String
Data	String
Boolean	String
Bytes	String

5.3 Logstash

5.3.1 カスタムインストール

Logstash は、クイックインストールまたはカスタムインストールによりインストールします。

logstroudsburg のインストールで詳細設定が必要な場合は、カスタムインストールを選択し、デフォルトのインストール構成を変更します。

1. Java をインストール

   a. インストールパッケージをダウンロード

      Java 公式 Web サイトより JDK をダウンロードします。

   b. 環境変数を設定

      システムの詳細設定で環境変数を追加または変更します。

      - **PATH:**C:\Program Files\Java\jdk1.8.0_73\bin
      - **CLASSPATH:**C:\Program Files\Java\jdk1.8.0_73\lib;C:\Program Files\Java\jdk1.8.0_73\lib\tools.jar
      - **JAVA_HOME:**C:\Program Files\Java\jdk1.8.0_73

   c. 確認

      PowerShellまたはcmd.exeを実行して確認します。

      ```
      PS C:\Users\Administrator> java -version
      java version "1.8.0_73"
      Java(TM) SE Runtime Environment (build 1.8.0_73-b02)
      Java HotSpot(TM) 64-Bit Server VM (build 25.73-b02, mixed mode)
      PS C:\Users\Administrator> javac -version
      javac 1.8.0_73
      ```

2. Logstash をインストール

   a. 公式の Web サイトからインストールパッケージをダウンロードします。

      Logstashのホームページで Version 2.2 以降を選択します。

   b. Logstash をインストールします。

      logstash-2.2.2.zip を C:\logstash-2.2.2 ディレクトリに展開します。

      Logstash 起動プログラムが C:\logstash-2.2.2\bin\logstash.batであることを確認します。

3. Logstash が Log Service にログを書き込むためプラグインをインストール

   マシンのネットワーク環境に合わせて、オンラインまたはオフラインでプラグインをインストールします。

   - オンラインインストール

     RubyGems がホストのプラグインです。 詳細は、こちらをご参照ください。

     PowerShellまたは cmd.exeを実行して Logstash インストールディレクトリに移動します。

     ```
     PS C:\logstash-2.2.2> .\bin\plugin install logstash-output-logservice
     ```

   - オフラインインストール

     Logstash の公式 Web サイトからダウンロードします。 logstash-output-logservice ページに移動し、右下隅のダウンロードをクリックします。

     ログが収集されたマシンがインターネットにアクセスできない場合は、ダウンロードした gem パッケージをマシンの C:\logstash-2.2.2ディレクトリにコピーします。PowerShell またはcmd.exeを実行して Logstash インストールディレクトリに移動します。次のコマンドを実行して iLogstash をインストールします。

     ```
     PS C:\logstash-2.2.2> .\bin\plugin install C:\logstash-2.2.2\logstash-output-logservice-0.2.0.gem
     ```

   - 確認

     ```
     PS C:\logstash-2.2.2> .\bin\plugin list
     ```

     マシンのプラグインリストに「logstash-output-logservice」があることを確認します。

4. NSSM のインストール

   Logstash の公式 Web サイトよりダウンロードします。NSSM のインストールパッケージをダウンロードするには、NSSM の公式 Web サイトにアクセスします。

   インストールパッケージをローカルマシンにダウンロード後、C:\logstash-2.2.2\nssm-2.24 ディレクトリに展開します。

5.3.2 Logstash 収集の構成ファイル作成

プラグインパラメータ

- **logstash-input-file**

  本プラグインは、tail モードでログファイルを収集するときに使用します。 詳細は、logstash-input-fileをご参照ください。

   注：
  path にはファイルパスを Unix のファイル区切り文字を使用して指定します (例: C:/test/multiline/*.log)。Unix のファイル区切り文字でない場合、ファジーマッチに対応しません。

- **logstash-output-logservice**

  本プラグインは、logstash-input-file プラグインによって収集されたログを Log Service に出力するために使用します。

パラメータ	説明
endpoint	Log Service のエンドポイント (例: http://regionid.example.com、詳細は「Log Service エンドポイント」を参照)
project	Log Service プロジェクトの名前
logstore	Logstore 名
topic	ログトピック名 (デフォルト値: null)
source	ログソース (本パラメータの設定が空の場合、ローカルマシンの IP アドレスがログソースになります)
access_key_id	Alibaba Cloud アカウントの AccessKeyID
access_key_secret	Alibaba Cloud アカウントのキーシークレット
max_send_retry	例外のためにデータパケットを Log Service に送信できない場合に実行される最大再試行回数 (200 ミリ秒ごとに再試行し、再試行に失敗したデータパケットは破棄されます)

1. 収集の構成ファイルを作成

   C:\logstash-2.2.2-win\conf\ディレクトリに構成ファイルを作成し、Logstash を再起動してファイルを適用します。

   ログタイプごとに構成ファイルを作成することができます。ファイルの拡張子は*.confにします。構成ファイルは C:\logstash-2.2.2-win\conf\ディレクトリに作成し、管理しやすくすることをお勧めします。

    注：

構成ファイルは、BOM なしの UTF-8 でエンコードする必要があります。Notepad++ をダウンロードして、ファイルのエンコード形式を変更します。

- IIS ログ

  詳細は「Logstash を使用した IIS ログの収集」をご参照ください。

- CSV ログ

  ログ収集のシステム時間が、ログのアップロード時間になります。詳細は、CSV ログ設定をご確認ください。

- デフォルトのログ時間

  CSV ログの場合、 ログコンテンツ内の時間が、ログのアップロード時間になります。詳細については、「Logstash を使用した CSV ログの収集」をご参照ください。

- 一般的なログ

  デフォルトでは、ログ収集のシステム時間が、ログのアップロード時間になります。ログのフィールドは解析されません。一行および複数行のログの両方がサポートされています。詳細については「Logstash を使用したその他のログの収集」をご参照ください。

2. 構成の構文確認

   a. PowerShellまたは cmd.exeを実行して Logstash インストールディレクトリに移動します。

   ```
   PS C:\logstash-2.2.2-win\bin> .\logstash.bat agent --configtest --config C:\
   logstash-2.2.2-win\conf\iis_log.conf
   ```

   b. 収集の構成ファイルを変更します。コンソールに収集結果を出力するには、出力フェーズに rubydebug を一時的に追加します。必要に応じてタイプフィールドを設定します。

   ```
   output {
   If [type] = "***"{
     stdout { codec => rubydebug }
     logservice {

     }
   ```

   c. PowerShellまたはcmd.exeを実行して Logstash インストールディレクトリに移動し、プロセスを起動します。

   ```
   PS C:\logstash-2.2.2-win\bin> .\logstash.bat agent -f C:\logstash-2.2.2-win\conf
   ```

   確認後、logstash.batプロセスを終了し、一時的な rubydebug 設定を削除します。

PowerShell で logstash.bat を起動すると、Logstash プロセスはフォアグラウンドで動作します。一般的に、Logstash は設定のテストやログ収集のデバッグに使用します。したがって、Logstash を電源投入時に自動的に起動し、バックグラウンドで実行させるには、デバッグ後に Logstash を Windows サービスに登録することをお勧めします。Logstash を Windows サービスに登録する方法については、「Logstash を Windows サービスに登録」をご参照ください。

5.3.3 Logstash を Windows サービスに登録

PowerShell で Logstash.bat が起動すると、Logstash プロセスはフロントエンドで実行されます。Logstash は通常、構成の確認や収集のデバッグに使用します。したがって、電源投入時に自動的に Logstash が起動し、バックエンドで実行させるよう、デバッグ後に Logstash を Windows サービスに設定されることをお勧めします。

Logstash を Windows サービスとして設定する以外にも、コマンドラインよりサービスを開始、停止、変更、および削除することもできます。NSSM の使用方法については、NSSM 公式文書をご参照ください。

Logstash を Windows サービスに追加

通常は、Logstash を初めてデプロイする際に追加します。Logstash を既に追加している場合は、この手順をスキップします。

Logstash を Windows サービスに追加するには、次のコマンドを実行します。

- 32 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe install logstash "C:\logstash-2.2.2-win\bin\logstash.bat" "agent -f C:\logstash-2.2.2-win\conf"
```

- 64 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe install logstash "C:\logstash-2.2.2-win\bin\logstash.bat" "agent -f C:\logstash-2.2.2-win\conf"
```

サービスの起動

Logstash のconfディレクトリにある設定ファイルが更新されている場合は、Logstash サービスを一旦停止し、再起動します。

サービスを開始するには、次のコマンドを実行します。

- 32 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe start logstash
```

- 64 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe start logstash
```

サービスの停止

サービスを停止するには、次のコマンドを実行します。

- 32 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe stop logstash
```

- 64 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe stop logstash
```

サービスの修正

サービスを変更するには、次のコマンドを実行します。

- 32 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe edit logstash
```

- 64 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe edit logstash
```

サービスの削除

サービスを削除するには、次のコマンドを実行します。

- 32 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win32\nssm.exe remove logstash
```

- 64 ビットシステム

```
C:\logstash-2.2.2-win\nssm-2.24\win64\nssm.exe remove logstash
```

5.3.4 高度な関数

Logstash には、あらゆる要件に対応できるようさまざまなプラグインが用意されています。たとえば、

- grok: 正規表現を使用してログを複数のフィールドに構造解析
- json_lines、json: JSON ログの構造解析

- date: ログの日付フィールドおよび時間フィールドを解析して変換
- multiline: さまざまな複数行のログをカスタマイズ
- kv: キーと値がペアになっているログの構造解析

5.3.5 Logstash エラー処理

Logstash によるログ収集に、以下の収集エラーが発生した場合は、そのガイドラインに従って、エラーを処理します。

Logstash によるログ収集に、以下の収集エラーが発生した場合は、そのガイドラインに従って、エラーを処理します。

- Log Service の文字化けデータ

  Logstash は、デフォルトでファイルを UTF-8 エンコードします。入力ファイルが正しくエンコードされているかどうかを確認します。
- コンソールのエラーメッセージ

  コンソールに、エラーメッセージ `io/console not supported; tty will not be manipulated` が表示されます。ただし、本エラーは機能自体には影響を及ぼさないため、無視して構いません。

  その他のエラーの発生時には、Google または Logstash フォーラムでの検索をお勧めします。

5.4 SDK 収集

5.4.1 Producer Library

LogHub Producer Library は、並行性の高い Java アプリケーション向けに作成された LogHub のクラスライブラリです。Producer Library およびConsumer Libraryは、LogHub がデータの収集および読み込みのしきい値を下げるための読み書きパッケージです。

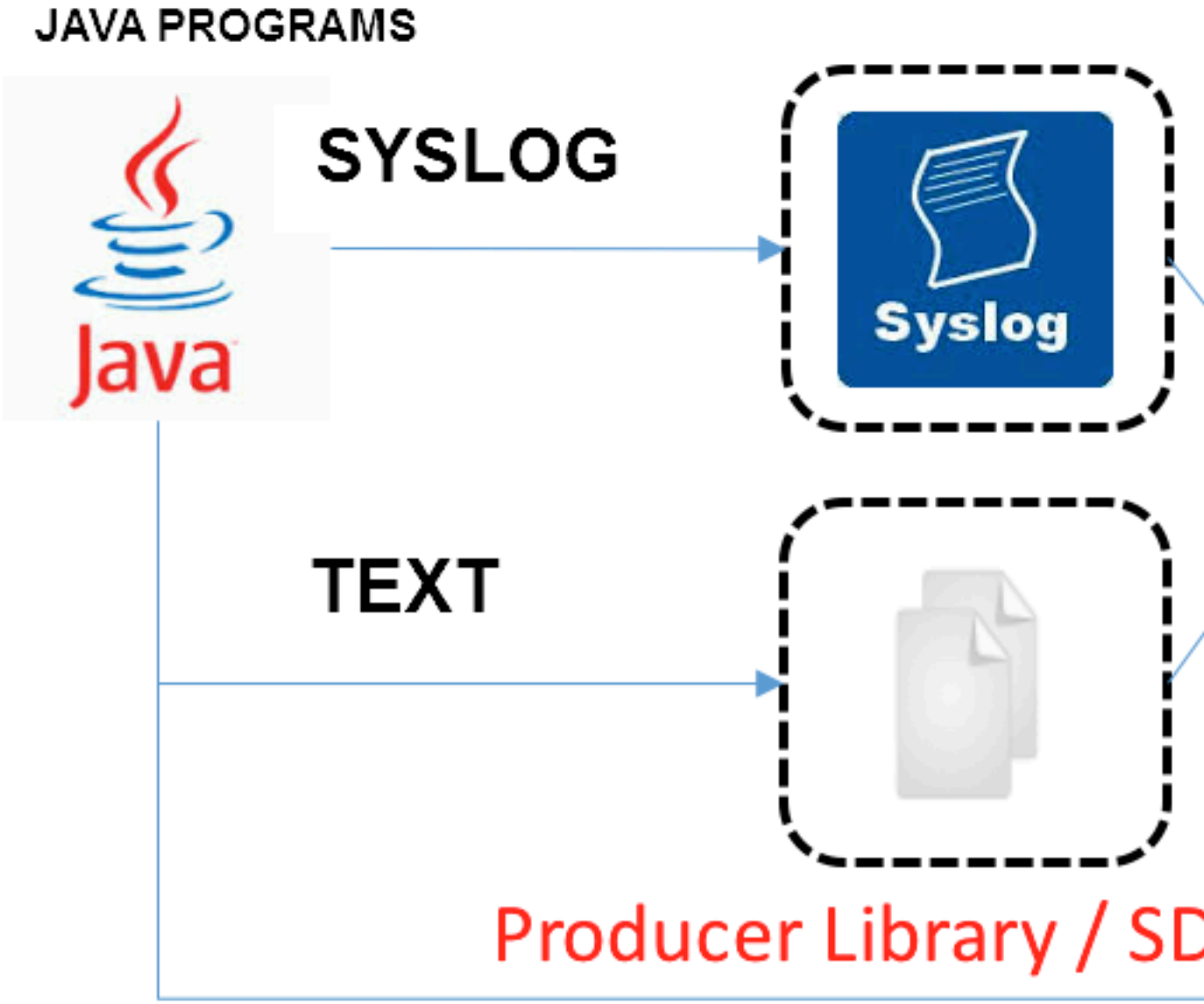
機能の特徴

- 非同期送信インターフェイスにより、セキュアなスレッドを保証します。
- プロジェクト構成を複数追加することができます。
- ログ送信のネットワーク I/O スレッド数を設定することができます。
- 統合パッケージのログの件数とサイズを設定することができます。
- メモリ使用量を調節できます。メモリ使用量が指定のしきい値に達すると、アイドルメモリが使用可能になるまで Producer の送信インターフェイスはブロックされます。

機能の利点

- クライアントからの収集ログは、ディスクには書き込まれません。ログデータが生成されると、ネットワークを介してそのまま Log Service に送信されます。
- 高い並行処理でクライアントに書き込まれます (100 件/秒を超える書き込み処理)。
- クライアントコンピューティングの I/O は論理的に分離されます。ログの書き込みは、処理時間に影響しません。

Producer Library を使用することにより、プログラム開発が簡素になります。書き込みリクエストは集約され、非同期に LogHub サーバーに送信されます。プログラムに集約パラメーター、また、サーバーにエラーが発生したときの処理を指定することができます。



上記のアクセス方法の比較一覧は下表のとおりです。

アクセス方法	利点/欠点	シナリオ
ログのディスクへの書き込み + Logtail	ログ収集とログは切り離されているため、コードを書き換える必要がありません。	一般的なシナリオ
Syslog + Logtail	高パフォーマンス (80 MB/s)。ログはディスクに書き込まれません。syslog プロトコルに対応している必要があります。	Syslog シナリオ

アクセス方法	利点/欠点	シナリオ
SDK による直接送信	ディスクに書き込まれず、直接サーバーに送信されます。ネットワーク I/O とプログラム I/O の切り替えは、適切に処理する必要があります。	ログはディスクに書き込まれません。
Producer Library	ディスクに書き込まれず、非同期にマージされ、サーバーに送信されるため、高スループットです。	ログはディスクに書き込まれず、クライアントのQPSは高くなります。

手順

- Java Producer
- Log4J1. Log4J1.XAppender (based on Java Producer)
- Log4J2. XAppender (based on Java Producer)
- LogBack Appender (based on Java Producer)
- C Producer
- C Producer Lite

5.4.2 LogHub Log4j Appender

Log4j は Apache のオープンソースプロジェクトであり、ログの出力先をコンソール、ファイル、GUI コンポーネント、ソケットサーバー、NT イベントレコーダー、または UNIX Syslog デーモンに設定できます。各ログの出力形式およびレベルを設定して、細かい粒度でログの生成を制御することもできます。構成ファイルに設定するため、アプリケーションコードを書き換える必要がなく、柔軟に対応できます。

Alibaba Cloud の Log4j Appender では、ログの出力先を Log Service に指定することができます。ダウンロードおよびユーザーガイドは、GitHubをご参照ください。

5.4.3 C Producer Library

LogHub は、Java Producer Library だけでなく、C Producer Library および Producer Lite Library にも対応しており、プラットフォーム間をシンプルかつ高性能に、最小限のリソースでワンストップのログ収集を実現できます。

GitHub プロジェクトの URL は、以下をご参照ください。

- C Producer Library (サーバー向け)、C Producer Library (サーバー向け)
- C Producer Lite Library (IOT およびスマートデバイス向け)

5.4.4 Go Producer Library

Aliyun LOG Go Producer Library は、使いやすく、高度に設定可能な Go ライブラリです。 並行性の高いビッグデータシナリオで実行される Go アプリケーションに合わせて調整されています。 Aliyun LOG Go Producer Library を使用することにより、Go アプリケーションは失敗したログを自動的に再送信し、送信するデータを圧縮してデータ書き込み効率を向上させます。

詳細は、GitHub の Aliyun Log Go Producer をご参照ください。

5.5 一般的なログフォーマット

5.5.1 Log4j ログ

アクセスモード

Log Service では、次のプロダクトを使用して Log4j を収集します。

- LogHub Log4j Appender
- Logtail

LogHub Log4j Appender を使用した Log4j ログの収集

詳細については、「LogHub Log4j Appender」をご参照ください。

Logtail を使用した Log4j ログの収集

Log4j ログには第 1 世代および第 2 世代がありますが、本ドキュメントでは、第 1 世代のデフォルト設定を例に、Log4j の一般的な設定方法を説明します。第 2 世代の Log4j を使用している場合、完全な日付を出力するために、デフォルトの設定を修正する必要があります。

```
<Configuration status="WARN">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{yyyy-MM-dd HH:mm:ss:SSS zzz} [%t] %-5level %logger{
36} - %msg%n"/>
    </Console>
  </Appenders>
  <Loggers>
    <Logger name="com.foo.Bar" level="trace">
      <AppenderRef ref="Console"/>
    </Logger>
    <Root level="error">
      <AppenderRef ref="Console"/>
    </Root>
  </Loggers>
</Configuration>
```

Log4j のログを収集するための設定については、「Python ログ」をご参照ください。ネットワーク構成およびネットワーク設定に適した設定を選択します。

自動生成される正規表現は、ログサンプルにのみ基づいており、すべてのログフォーマットに対応しているわけではありません。したがって、正規表現が自動生成されてから、微修正する必要があります。

Log4j のログファイルに出力される Log4j のデフォルトのログフォーマットのサンプルは次のとおりです。

```
2013-12-25 19:57:06,954 [10.207.37.161] WARN impl.PermanentTairDaoImpl - Fail
to Read Permanent Tair,key:e:470217319319741_1,result:com.example.tair.Result@
172e3ebc[rc=code=-1, msg=connection error or timeout,value=,flag=0]
```

複数行ログの行の先頭を一致させる (IP アドレスで行の先頭を示す)

```
\d+-\d+-\d+\s.
```

ログデータの抽出に使用される正規表現

```
(\d+-\d+-\d+\s\d+:\d+:\d+,\d+)\s\[([^\]]*)\]\s(\S+)\s+(\S+)\s-\s(.
```

時間の変換書式

```
%Y-%m-%d %H:%M:%S
```

ログサンプルの抽出結果

キー	値
time	2013-12-25 19:57:06,954
ip	10.207.37.161
level	WARN
class	impl.PermanentTairDaoImpl
message	Fail to Read Permanent Tair,key:e:470217319319741_1,result:com.example.tair.Result@172e3ebc[rc=code=-1, msg=connection error or timeout,value=,flag=0]

5.5.2 Python ログ

Python のロギングモジュールは、サードパーティーのモジュールやアプリケーションで使用できる汎用のロギングシステムです。ロギングモジュールは、ファイル、HTTP GET/POST、SMTP、Socket といったさまざまな方法で、さまざまなログレベルおよびログを記録します。ロギング方法をカスタマイズすることもできます。ロギングモジュールは基本的には Log4j と同じですが、細部が異なります。ロギングモジュールには、Logger、Handler、Filter、および Formatter 機能があります。

Python ログを収集するには、logging handler をそのまま使用されることをお勧めします。

- logging handler で Python ログを自動アップロード
- logging handler で KV 形式のログを自動解析
- logging handler で JSON 形式のログを自動解析

Python のログフォーマット

formatter のログ出力フォーマットがログフォーマットです。Formatter には、メッセージのフォーマット (String 型) および日付 (String 型) の 2 つのパラメーターを定義します。両パラメーターともオプションです。

Python のログフォーマット

```
import logging
import logging.handlers
LOG_FILE = 'tst.log'
handler = logging.handlers.RotatingFileHandler(LOG_FILE, maxBytes = 1024*1024,
backupCount = 5) # Instantiate the handler
fmt = '%(asctime)s - %(filename)s:%(lineno)s - %(name)s - %(message)s'
formatter = logging.Formatter(fmt)   # Formatter を初期化
handler.setFormatter(formatter)      # Handler に Formatter を追加
logger = logging.getLogger('tst')    # 「tst」Logger を取得
logger.addHandler(handler)           # Logger に Handler を追加
logger.setLevel(logging.DEBUG)
logger.info('first info message')
logger.debug('first debug message')
```

フィールドの説明

Formatter の設定書式は、%(key)sです。属性辞書のキーワードを置き換えます。キーワードは、以下のとおりです。

フォーマット	意味
%(name)s	生成されたログの Logger 名
%(levelno)s	メッセージのログレベル (DEBUG、INFO、WARNING、ERROR、および CRITICAL) の番号
%(levelname)s	メッセージのログレベル (DEBUG、INFO、WARNING、ERROR、および CRITICAL) の文字列
%(pathname)s	ロギングの呼び出しソースファイルの完全パス (取得可能な場合)
%(filename)s	ファイル名
%(module)s	ロギングの呼び出しモジュール名
%(funcName)s	ロギングの呼び出し関数名

フォーマット	意味
%(lineno)d	ロギングの呼び出しコードの行 (取得可能な場合)
%(created)f	ログの生成時間 (UNIX タイムスタンプ)。1970-1-100 00:00:00 UTC からの秒数。
%(relativeCreated)d	ログの生成時間およびロギングモジュールのロード時間との差 (単位: ミリ秒)
%(asctime)s	ログの生成時間。書式: デフォルトで「2003-07-08 16：49：45,896」(コンマ (,) の後の数字はミリ秒数)
%(msecs)d	ログ生成時間 (単位: ミリ秒)
%(thread)d	スレッドID (取得可能な場合)
%(threadName)s	スレッド名 (取得可能な場合)
%(process)d	プロセス ID (オプション)
%(message)s	ログメッセージ

ログサンプル

ログサンプル

```
2015-03-04 23:21:59,682 - log_test.py:16 - tst - first info message
2015-03-04 23:21:59,682 - log_test.py:17 - tst - first debug message
```

一般的な Python ログおよび正規表現

- ログフォーマット

  ```
  2016-02-19 11:03:13,410 - test.py:19 - tst - first debug message
  ```

  正規表現

  ```
  (\d+-\d+-\d+\s\S+)\s+-\s+([^:]+):(\d+)\s+-\s+(\w+)\s+-\s+(. *)
  ```

- ログフォーマット

  ```
  %(asctime)s - %(filename)s:%(lineno)s - %(levelno)s %(levelname)s %(pathname)s
   %(module)s %(funcName)s %(created)f %(thread)d %(threadName)s %(process)d %(
  name)s - %(message)s
  ```

  ログサンプル

  ```
  2016-02-19 11:06:52,514 - test.py:19 - 10 DEBUG test.py test <module> 1455851212.
  514271 139865996687072 MainThread 20193 tst - first debug message
  ```

  正規表現

  ```
  (\d+-\d+-\d+\s\S+)\s-\s([^:]+):(\d+)\s+-\s+(\d+)\s+(\w+)\s+(\S+)\s+(\w+)\s+(\S+)\s
  +(\S+)\s+(\d+)\s+(\w+)\s+(\d+)\s+(\w+)\s+-\s+(. *)
  ```

Logtail を使用した Python ログの収集

Logtail を使って Python ログを収集する詳しい手順については、「5 分でクイックスタート」をご参照ください。ネットワーク構成とネットワーク設定をもとに、対応する構成を選択します。

1. プロジェクトおよび Logstore を作成します。手順の詳細については、「準備」をご参照ください。
2. **Logstore**リストページで、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. データソースを選択します。

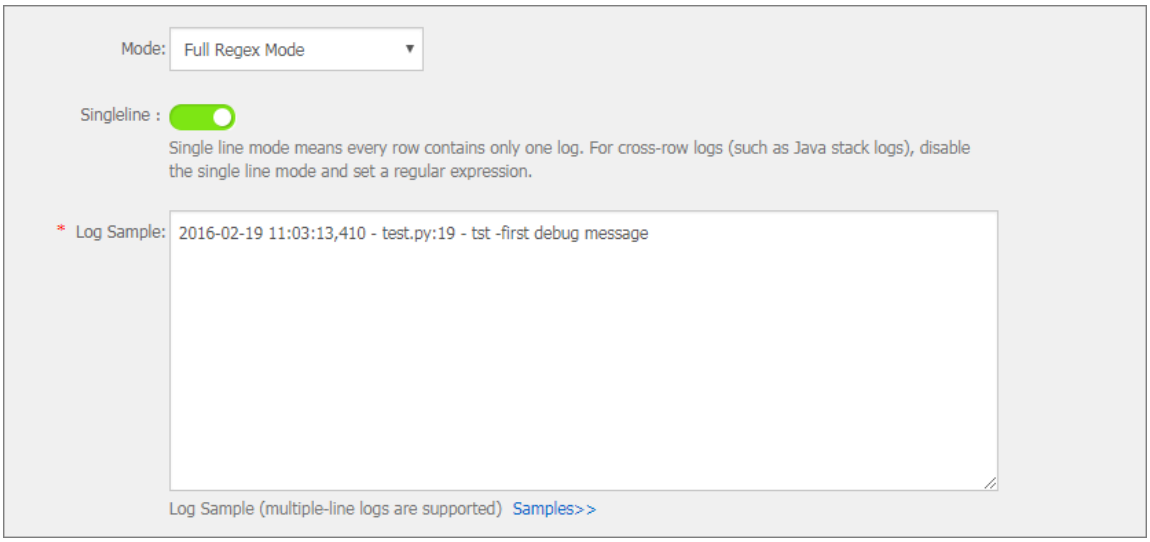
   テキストを選択します。

4. データソースを設定します。

   a. 構成名およびログパスを入力し、モードドロップダウンリストで完全正規表現モードを選択します。

   b. 単一行スイッチをオンにします。

   c. ログサンプルを入力します。



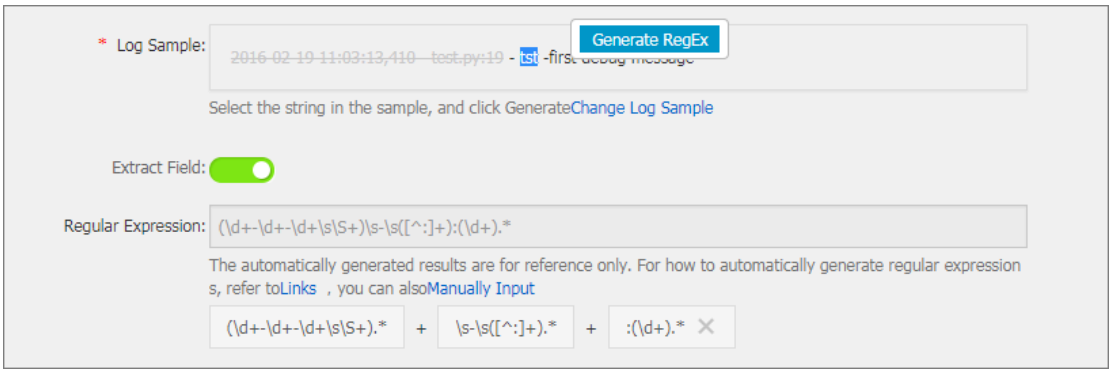
   d. フィールド抽出スイッチをオンにします。

   e. 正規表現を設定します。

      A. ログサンプルの文字列を選択して正規表現を生成します。

         自動生成された正規表現がログサンプルと異なる場合、ログサンプルの文字列を選択して正規表現を生成することができます。ログサンプルの文字列を選択すると、Log Service は選択した文字列からフィールドを自動解析し、正規表現を自動生成します。ログサンプルで、ログフィールドを選択し、**Generate RegEx**をクリックします。選択

したフィールドの正規表現が正規表現列に表示されます。複数の選択することにより、ログサンプルの完全な正規表現が生成されます。



B. 正規表現を修正します。

ログフォーマットを修正する必要がある場合、手動入力をクリックして、自動生成された正規表現を修正して、収集処理に必要なログフォーマットに一貫性を持たせます。

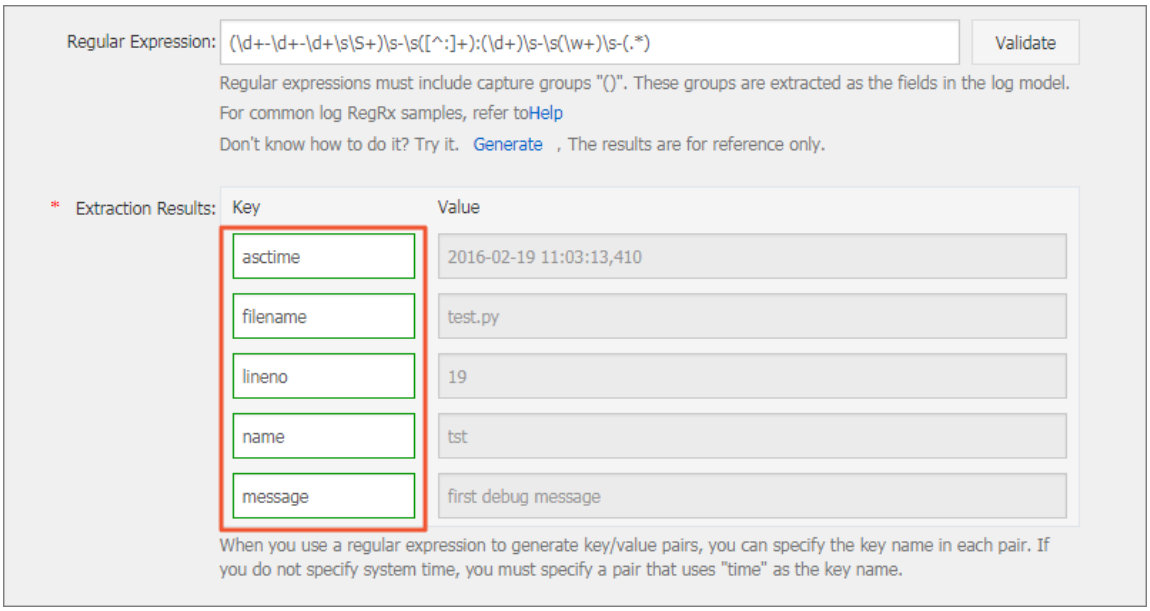
C. 正規表現を検証します。

正規表現を修正したら、検証をクリックします。正規表現が正しい場合、抽出結果が表示されます。エラーがあった場合には、正規表現を修正します。

f. 抽出結果を確認します。

ログフィールドの解析結果を表示し、ログ抽出結果に対応するキーを入力します。

各ログフィールドの抽出結果は、容易に判別できるフィールド名を割り当てます。たとえば、time に時間フィールドを割り当てます。システム時間を使用しない場合は、time キーに値が時間であるフィールドを割り当てます。



g. システム時間スイッチをオンにします。

システム時間を採用すると、Logtail クライアントがログを解析した時間がログ時間になります。

h. (オプション) 詳細オプションを設定します。

i. 次へをクリックします。

Logtail 設定が完了したら、設定をマシングループに適用して Python ログを収集します。

5.5.3 Node.js ログ

Node.js のログは、直接コンソールに出力されます。ログデータの収集およびトラブルシューティングには不便です。Log4js を使用すると、ログをファイルに出力し、ログフォーマットをカスタマイズすることができます。データの収集にも運用にも便利です。

```
var log4js = require('log4js');
log4js.configure({
  appenders: [
    {
      type: 'file', //file output
      filename: 'logs/access.log',
      maxLogSize: 1024,
      backups:3,
      category: 'normal'
    }
  ]
});
var logger = log4js.getLogger('normal');
logger.setLevel('INFO');
logger.info("this is a info msg");
logger.error("this is a err msg");
```

ログフォーマット

Log4js を使用してログデータをテキストファイルに保存する場合、ファイルに出力されるログフォーマットは、次のとおりです。

```
[2016-02-24 17:42:38.946] [INFO] normal - this is a info msg
[2016-02-24 17:42:38.951] [ERROR] normal - this is a err msg
```

Log4js には、trace、debug、info、warn、error、および fatal の 6 つの出力ログレベルがあります (後者ほどより深刻)。

Logtail を使って Node.js ログを収集

Logtail を使用して Node.js ログを収集する方法については、「Python ログ」をご参照ください。ネットワーク構成およびネットワーク設定に適した設定を選択します。

自動生成される正規表現は、ログサンプルのみを基にしており、すべてのログフォーマットに対応しているわけではありません。したがって、正規表現が自動生成されたら、修正を加える必要があります。以下の Node.js ログサンプルを参考に、ログに必要な正規表現を記述します。

一般的な **Node.js** ログとその正規表現は、以下のとおりです。

- ログサンプル 1
  - ログサンプル

    ```
    [2016-02-24 17:42:38.946] [INFO] normal - this is a info msg
    ```

  - 正規表現

    ```
    \[([^]]+)]\s\[([^\]]+)]\s(\w+)\s-(. *)
    ```

  - 抽出フィールド

    time、level、loggerName、および message

- ログサンプル 2
  - ログサンプル

    ```
    [2016-01-31 12:02:25.844] [INFO] access - 42.120.73.203 - - "GET /user/projects/
    ali_sls_log? ignoreError=true HTTP/1.1" 304 - "http://
    aliyun.com/" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36
     (KHTML, like Gecko) Chrome/48.0.2564.97 Safari/537.36"
    ```

  - 正規表現

    ```
    \[([^]]+)]\s\[(\w+)]\s(\w+)\s-\s(\S+)\s-\s-\s"([^"]+)"\s(\d+)[^"]+("[^"]+)"\s"([^"]+). *
    ```

  - 抽出フィールド

    time、level、loggerName、ip、request、status、referer および user_agent

5.5.4 Wordpress ログ

WordPress のログフォーマット (デフォルト)

ログ (未加工) のサンプル

```
172.64.0.2 - - [07/Jan/2016:21:06:39 +0800] "GET /wp-admin/js/password-strength
-meter.min.js? ver=4.4 HTTP/1.0" 200 776 "http://wordpress.c4a1a0aecdb194316955
5231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php" "Mozilla/5.0 (
```

```
Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.
0.2526.106 Safari/537.36"
```

複数行ログの行の先頭の正規表現 (行の始まりは IP アドレス):

```
\d+\.\d+\.\d+\.\d+\s-\s. *
```

ログ情報を抽出する正規表現:

```
(\S+) - - \[([^\]]*)] "(\S+) ([^"]+)" (\S+) (\S+) "([^"]+)" "([^"]+)"
```

時間の変換書式:

```
%d/%b/%Y:%H:%M:%S
```

ログサンプルの抽出結果

キー	値
ip	10.10.10.1
time	07/Jan/2016:21:06:39 +0800
method	GET
url	/wp-admin/js/password-strength-meter.min.js? ver=4.4 HTTP/1.0
status	200
length	776
ref	http://wordpress.c4a1a0aecdb1943169555231dcc4adfb7.cn-hangzhou.alicontainer.com/wp-admin/install.php
user-agent	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.106 Safari/537.36

5.5.5 ThinkPHP ログ

ThinkPhP は PHP 言語に基づいた、Web アプリケーションの開発フレームワークです。

ログフォーマット

ThinkPhP では、以下のロギング方法が使用されています。

```
<? php
Think\Log::record('D method instantiation does not find the model class' );
```

ログサンプル

```
[ 2016-05-11T21:03:05+08:00 ] 10.10.10.1 /index.php
```

```
INFO: [ app_init ] --START--
INFO: Run Behavior\BuildLiteBehavior [ RunTime:0.000014s ]
INFO: [ app_init ] --END-- [ RunTime:0.000091s ]
Info: [app_begin] -- start --
INFO: Run Behavior\ReadHtmlCacheBehavior [ RunTime:0.000038s ]
INFO: [ app_begin ] --END-- [ RunTime:0.000076s ]
INFO: [ view_parse ] --START--
INFO: Run Behavior\ParseTemplateBehavior [ RunTime:0.000068s ]
INFO: [ view_parse ] --END-- [ RunTime:0.000104s ]
INFO: [ view_filter ] --START--
INFO: Run Behavior\WriteHtmlCacheBehavior [ RunTime:0.000032s ]
INFO: [ view_filter ] --END-- [ RunTime:0.000062s ]
INFO: [ app_end ] --START--
INFO: Run Behavior\ShowPageTraceBehavior [ RunTime:0.000032s ]
INFO: [ app_end ] --END-- [ RunTime:0.000070s ]
ERR: D method instantiation does not find the model class
```

Logtail を使用して ThinkPhP ログを収集

Logtail を使用して ThinkPhP ログを収集する手順については、「Python ログ」をご参照ください。ネットワーク構成およびネットワーク設定に合わせて構成を選択します。

自動生成される正規表現は、ログサンプルのみを基にしており、すべてのログフォーマットに対応しているわけではありません。したがって、正規表現が自動生成されたら、修正を加える必要があります。

ThinkPhP ログは、複数行に渡り、定まった形式がありません。ThinkPhP ログより、時間、ユーザーの IP アドレス、アクセス URL、表示メッセージといったフィールドを抽出できます。メッセージフィールドの情報は複数行に渡り、定型化されていないため、1 つのフィールドとして扱われます。

Logtail より **ThinkPhP** ログの設定パラメータを取得

行の先頭の正規表現:

```
\[\s\d+-\d+-\w+:\d+:\d+\+\d+:\d+\s.
```

正規表現:

```
\[\s(\d+-\d+-\w+:\d+:\d+)[^:]+:\d+\s]\s+(\S+)\s(\S+)\s+(.
```

時間書式:

```
%Y-%m-%dT%H:%M:%S
```

5.5.6 Unity3D

Unity3D は、あらゆるプラットフォーム向けの統合されたゲーム開発ツールです。Unity Technologies の開発した Unity3D では、3D ビデオゲーム、建築的視覚化、リアルタイム 3D ア

ニメーションといったさまざまなインタラクティブコンテンツを簡単に作成できます。Unity3D は完全に統合されたプロフェッショナルなゲームエンジンです。

Log Service の Web トラッキング を使用することにより、Unity3D のログを簡単に収集できます。本ドキュメントでは、Webトラッキング機能を使用して Unity3D のログ Unity Debug.Log を Log Service に収集する方法を紹介します。

1. Web トラッキング機能の有効化

   詳細については、「Web トラッキング」をご参照ください。

2. Unity3D LogHandler の登録

   Unity エディタで C# ファイル LogOutputHandler.csを作成します。以下のコードを入力し、コード内の次の 3 つのメンバー変数を変更します。

   - プロジェクト: ログプロジェクトの名前
   - Logstore: Logstore の名前
   - serviceAddr: ログプロジェクトのアドレス

     詳細については、「サービスエンドポイント」をご参照ください。

```
using UnityEngine;
using System.Collections;
public class LogOutputHandler : MonoBehaviour
{
    //Register the HandleLog function on scene start to fire on debug.log events
    public void OnEnable()
    {
        Application.logMessageReceived += HandleLog;
    }
    //Remove callback when object goes out of scope
    public void OnDisable()
    {
        Application.logMessageReceived -= HandleLog;
    }
    string project = "your project name";
    string logstore = "your logstore name";
    string serviceAddr = "http address of your log service project";
    //Capture debug.log output, send logs to Loggly
    public void HandleLog(string logString, string stackTrace, LogType type)
    {
        string parameters = "";
        parameters += "Level=" + WWW.EscapeURL(type.ToString());
        parameters += "&";
        parameters += "Message=" + WWW.EscapeURL(logString);
        parameters += "&";
        parameters += "Stack_Trace=" + WWW.EscapeURL(stackTrace);
        parameters += "&";
        //Add any User, Game, or Device MetaData that would be useful to finding issues
later
        parameters += "Device_Model=" + WWW.EscapeURL(SystemInfo.deviceModel);
        string url = "http://" + project + "." + serviceAddr + "/logstores/" + logstore + "/track
? APIVersion=0.6.0&" + parameters;
        StartCoroutine(SendData(url));
    }
```

```
    public IEnumerator SendData(string url)
    {
        WWW sendLog = new WWW(url);
        yield return sendLog;
    }
}
```

上記のコードにより、非同期に Log Service にログを送信することができます。収集するフィールドを上記コードに追加することもできます。

3. Unity ログの生成

プロジェクトで、LogglyTest.csファイルを作成し、 次のコードを追加してください。

```
using UnityEngine;
using System.Collections;
public class LogglyTest : MonoBehaviour {
    void Start () {
        Debug.Log ("Hello world");
    }
}
```

4. コンソールでログのプレビュー

上記の手順を完了したら、Unity プログラムを実行します。送信されたログを Log Service コンソールでプレビューできます。

上記の例では、Debug.Log、Debug.LogError、Debug.LogExceptionといったログを収集する方法です。Unity のコンポーネントオブジェクトモデル、Unity プログラムのクラッシュ API、およびその他のログ API を使用することで、クライアント上のデバイス情報を容易に収集できます。

5.5.7 Logstash を使用した IIS ログの収集

Logstash を使用して IIS ログを収集する前に、IIS ログフィールドが解析されるよう、構成ファイルを変更する必要があります。

ログサンプル

IIS ログ設定を表示し、W3C 形式 (デフォルトのフィールド設定) を選択し、フォーマットを保存して有効にします。

```
2016-02-25 01:27:04 112.74.74.124 GET /goods/list/0/1.html - 80 - 66.249.65.102
Mozilla/5.0+(compatible;+Googlebot/2.1;++http://www.google.com/bot.html) 404 0 2
703
```

ログ収集の設定

```
input {
  file {
    type => "iis_log_1"
    path => ["C:/inetpub/logs/LogFiles/W3SVC1/*.log"]
    start_position => "beginning"
```

```
  }
}
filter {
  if [type] == "iis_log_1" {
  #ignore log comments
  if [message] =~ "^#" {
    drop {}
  }
  grok {
    # check that fields match your IIS log settings
    match => ["message", "%{TIMESTAMP_ISO8601:log_timestamp} %{IPORHOST:site}
 %{WORD:method} %{URIPATH:page} %{NOTSPACE:querystring} %{NUMBER:port} %{
NOTSPACE:username} %{IPORHOST:clienthost} %{NOTSPACE:useragent} %{NUMBER:
response} %{NUMBER:subresponse} %{NUMBER:scstatus} %{NUMBER:time_taken}"]
  }
    date {
    match => [ "log_timestamp", "YYYY-MM-dd HH:mm:ss" ]
      timezone => "Etc/UTC"
  }
  useragent {
    source=> "useragent"
    prefix=> "browser"
  }
  mutate {
    remove_field => [ "log_timestamp"]
  }
  }
}
output {
  if [type] == "iis_log_1" {
  logservice {
        codec => "json"
        endpoint => "***"
        project => "***"
        logstore => "***"
        topic => ""
        source => ""
        access_key_id => "***"
        access_key_secret => "***"
        max_send_retry => 10
    }
    }
}
```

注：

- 構成ファイルは、BOM なしの UTF-8 でエンコードする必要があります。Notepad++ をダウンロードして、ファイルのエンコード形式を変更します。
- pathにはファイルパスを指定します。C:/test/multiline/*.logのように Unix 形式のファイル区切り文字を使用します。Unix 形式以外のファイル区切り文字を使用すると、ファジー一致を使用できません。
- typeフィールドは統一させ、ファイル全体で一貫性が保たれている必要があります。また、マシンに Logstash 構成ファイルが複数ある場合、各構成ファイルの type フィールドは統一させます。統一されていない場合、データは正しく処理されません。

関連プラグイン: file および grok

Logstash を再起動して設定を適用

confディレクトリに構成ファイルを作成します。Logstash を再起動すると構成ファイルが適用されます。詳細は Logstash を Windows サービスに登録 をご参照ください。

5.5.8 Logstash を使用した CSV ログの収集

Logstash を使用して CSV ログを取得する前に、CSV ログフィールドが解析されるよう、構成ファイルを修正する必要があります。 CSV ログの収集は、ログを収集した時点のシステム時間とログ内に記載される時間をログをアップロードする時間として使用できます。ログ時間の各定義方法においては、CSV ログを収集するための Logstash の構成方法が 2 種類あります。

ログのアップロード時間をシステム時間に

- ログサンプル

```
10.116.14.201,-,2/25/2016,11:53:17,W3SVC7,2132,200,0,GET,project/shenzhen-test/logstore/logstash/detail,C:\test\csv\test_csv.log
```

- ログ収集の設定

```
input {
  file {
    type => "csv_log_1"
    path => ["C:/test/csv/*.log"]
    start_position => "beginning"
  }
}
filter {
  if [type] == "csv_log_1" {
  csv {
    separator => ","
    columns => ["ip", "a", "date", "time", "b", "latency", "status", "size", "method", "url", "file"]
  }
  }
}
output {
  if [type] == "csv_log_1" {
  logservice {
        codec => "json"
        endpoint => "***"
        project => "***"
        logstore => "***"
        topic => ""
        source => ""
        access_key_id => "***"
        access_key_secret => "***"
        max_send_retry => 10
    }
    }
```

```
}
```

注：

- 構成ファイルは、BOM なしの UTF-8 でエンコードする必要があります。Notepad++ をダウンロードして、ファイルのエンコード形式を変更します。
- pathにはファイルパスを指定します。C:/test/multiline/*.logのように Unix 形式のファイル区切り文字を使用します。Unix 形式以外のファイル区切り文字を使用すると、ファジー一致を使用できません。
- typeフィールドは統一させ、ファイル全体で一貫性が保たれている必要があります。また、マシンに Logstash 構成ファイルが複数ある場合、各構成ファイルの type フィールドは統一させます。統一されていない場合、データは正しく処理されません。

関連するプラグイン：fileおよびcsv。

- **Logstash** を再起動して設定を適用

  confディレクトリに構成ファイルを作成し、Logstashを再起動してファイルを適用します。詳細については、「Logstash を Windows サービスに登録」をご参照ください。

ログフィールドの内容をアップロードされたログ時刻として使用する

- ログサンプル

```
10.116.14.201,-,Feb 25 2016 14:03:44,W3SVC7,1332,200,0,GET,project/shenzhen-test/logstore/logstash/detail,C:\test\csv\test_csv_withtime.log
```

- 収集の設定

```
input {
  file {
    type => "csv_log_2"
    path => ["C:/test/csv_withtime/*.log"]
    start_position => "beginning"
  }
}
filter {
  if [type] == "csv_log_2" {
  csv {
    separator => ","
    columns => ["ip", "a", "datetime", "b", "latency", "status", "size", "method", "url", "file
"]
  }
  date {
    match => [ "datetime" , "MMM dd YYYY HH:mm:ss" ]
  }
  }
}
output {
  if [type] == "csv_log_2" {
  logservice {
       codec => "json"
```

```
        endpoint => "***"
        project => "***"
        logstore => "***"
        topic => ""
        source => ""
        access_key_id => "***"
        access_key_secret => "***"
        max_send_retry => 10
    }
    }
}
```

注：

- 構成ファイルは、BOM なしの UTF-8 でエンコードする必要があります。Notepad++ をダウンロードして、ファイルのエンコード形式を変更します。
- pathにはファイルパスを指定します。C:/test/multiline/*.logのように Unix 形式のファイル区切り文字を使用します。Unix 形式以外のファイル区切り文字を使用すると、ファジー一致を使用できません。
- typeフィールドは統一させ、ファイル全体で一貫性が保たれている必要があります。また、マシンに Logstash 構成ファイルが複数ある場合、各構成ファイルの type フィールドは統一させます。統一されていない場合、データは正しく処理されません。

関連プラグイン: fileおよびcsv

- **Logstash** の再起動で設定を適用

  confディレクトリに構成ファイルを作成し、Logstash を再起動すると、構成ファイルが適用されます。 詳細については、「Logstash を Windows サービスに登録をご参照ください。

5.5.9 Logstash を使用して他のログを収集

Logstash を使用してログを収集する前に、ログフィールドが解析されるよう、構成ファイルを修正する必要があります。

ログのアップロード時間をシステム時間に

- ログサンプル

```
2016-02-25 15:37:01 [main] INFO com.aliyun.sls.test_log4j - single line log
2016-02-25 15:37:11 [main] ERROR com.aliyun.sls.test_log4j - catch exception !
 java.lang.ArithmeticException: / by zero
    at com.aliyun.sls.test_log4j.divide(test_log4j.java:23) ~[bin/:?]
    at com.aliyun.sls.test_log4j.main(test_log4j.java:13) [bin/:?]
2016-02-25 15:38:02 [main] INFO com.aliyun.sls.test_log4j - normal log
```

- ログ収集の設定

```
input {
  file {
    type => "common_log_1"
```

```
    path => ["C:/test/multiline/*.log"]
    start_position => "beginning"
    codec => multiline {
      pattern => "^\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}"
      negate => true
      auto_flush_interval => 3
      what => previous
    }
  }
}
output {
  if [type] == "common_log_1" {
  logservice {
       codec => "json"
       endpoint => "***"
       project => "***"
       logstore => "***"
       topic => ""
       source => ""
       access_key_id => "***"
       access_key_secret => "***"
       max_send_retry => 10
    }
    }
}
```

注：

- 構成ファイルは、BOM なしの UTF-8 でエンコードする必要があります。Notepad++ をダウンロードして、ファイルのエンコード形式を変更します。
- pathにはファイルパスを指定します。C:/test/multiline/*.logのように Unix 形式のファイル区切り文字を使用します。Unix 形式以外のファイル区切り文字を使用すると、ファジー一致を使用できません。
- typeフィールドは統一させ、ファイル全体で一貫性が保たれている必要があります。また、マシンに Logstash 構成ファイルが複数ある場合、各構成ファイルの type フィールドは統一させます。統一されていない場合、データは正しく処理されません。

関連プラグイン: fileおよびmultiline（ログファイルが 1 行のみの場合、codec => multiline 設定を削除してください)

- **Logstash** を再起動して設定を適用

  confディレクトリに構成ファイルを作成し、Logstash を再起動してファイルを適用します。詳細については、「Logstash を Windows サービスに登録」をご参照ください。