Senju Service Manager 2024.0.0

Elasticsearch 連携機能ガイド

-ElasticsearchCollaborationGuide-

株式会社野村総合研究所

Nomura Research Institute, Ltd.

- 本書は、Senju Service Manager システムバージョン 2024.0.0 の Elasticsearch 連携機能について説明します。なお、万一ご不明な点や記載誤り・漏れなど、お気づきの点がございましたら弊社までお知らせ下さい。
- Senju Service Manager システムバージョン 2024.0.0 の Elasticsearch 連携機能に対応する Elasticsearch のシステムバージョンについては、Senju Service Manager のリリースノートの稼働環境を参照してください。
- 本書は、Senju Service Manager システムをインストールまたは利用する前に一読して下さい。なお、万一ご不明な点や記載誤り・漏れなど、お気づきの点がございましたら弊社までお知らせ下さい。
- 本書に記載した内容は予告無く変更することがあります。
- 本書の内容の一部または全部を無断でコピーすることは法律で禁止されています。
- Senju Service Manager は、株式会社野村総合研究所の登録商標です。
  Adobe 及び Acrobat は、Adobe Systems Incorporated(アドビ システムズ社)の商標です。
  Microsoft Windows は、米国 Microsoft Corporation の米国及び他の国における登録商標です。
  ORACLE は，米国 Oracle Corporation の登録商標です。
  Oracle Developer/2000 は，米 Oracle Corporation の登録商標です。
  Oracle Applications は，オラクル社の商標です。
  UNIX は、The Open Group の米国ならびに他の国における登録商標です。
  Intel および Pentium は、Intel Corporation の登録商標です。
  iPhone は Apple Inc.の登録商標です。
  PostgreSQL は、PostgreSQL の米国およびその他の国における商標または登録商標です。
  Elasticsearch 及び logstash は Elasticsearch 社の登録商標です。
  その他のすべての会社名や製品名は、それぞれの会社の商標、登録商標または、サービスマークです。
- 本書では、便宜上 Microsoft 社の Windows ファミリーを Windows と表記しています。
  また、特に断りが無い場合、Windows NT とは”Windows NT Workstation””Windows NT Server”を、Windows 2000 とは”Windows 2000 Professional””Windows 2000 Server””Windows 2000 Advanced Server”を、Windows XP とは”Windows XP Professional”を、Windows Server 2003 とは”Windows Server 2003 Standard Edition””Windows Server 2003 Enterprise Edition”を、Windows Vista とは ”Windows Vista Ultimate””Windows Vista Business” を 、 Windows 7 と は ”Windows 7 Ultimate””Windows 7 Professional”を、Windows 8.1 とは”Windows 8.1 Pro”を、Windows 10 とは ”Windows 10 Pro” を 、 Windows Server 2008 と は ”Windows Server 2008 Standard Edition””Windows Server 2008 Enterprise Edition” 、 Windows Server 2012 と は ”Microsoft Windows Server 2012 Standard Edition”を、Windows Server 2016 とは”Microsoft Windows Server 2016Standard Edition”を、Windows Server 2019 とは”Microsoft Windows Server 2019 Standard Edition”, ”Microsoft Windows Server 2019 Datacenter Edition”を、Windows Server 2022 とは ”Microsoft Windows Server 2022 Standard Edition” , ”Microsoft Windows Server 2022 Datacenter Edition” , ” Microsoft Windows Server 2022 Datacenter: Azure Edition ”を指します。
- 本書では、便宜上、Senju Service Manager を SSM、Senju Operation Conductor を SOC、Senju Enterprise Navigator を SEN と表記している箇所があります。また、Senju DevOperation Conductor と Senju Operation Conductor を合わせて Senju Operation Conductor と表記しています。

発行日　2024年　6月　1日
第1.0.0版

著作、編集、発行　株式会社野村総合研究所
お問合せ先　マルチクラウドインテグレーション事業本部
クラウド運用ソリューション事業部
〒220-0012　神奈川県横浜市西区みなとみらい4-4-1 横浜野村ビル

TEL： 0120-736-580
E-mail： senjuinfo@nri.co.jp
URL： http://senjufamily.nri.co.jp/

1 Elasticsearch 連携機能ガイド

1.1 Elasticsearch連携機能の概要

Elasticsearch は、多様なタイプのデータを蓄積し、検索および分析を行うための分散型の検索/分析エンジンです。
商用利用可能なオープンソースライセンスである Apache 2.0 license で提供されています。

Senju Service Manager では OSS の高速検索ツールである Elasticsearch およびその周辺ツールと連携して、Senju Service Manager に登録されているトランザクションデータを収集解析します。
サービスマネジメントプロセスの実現において、各プロセスを横断した高度な全文検索や、データの類似性に基づく類似インシデントの検索機能を提供します。

Elasticsearch 連携機能は、分析を行うための Elasticsearch サーバーと分析する日本語のテキストを単語に分割することに特化した形態素解析ソフトウェアプラグイン Kuromoji、Elasticsearch へのデータ連携を行う Logstash で構成されます。

1.2 稼働環境

Senju Service Manager で Elasticsearch 連携機能を利用するにあたり、Elasticsearch をインストールするサーバーの構成と、インストールする製品、バージョンを確認してください。

仕様補足 本機能は 2018.0.0.2 以降でご利用いただけます。

1.2.1 Elasticsearch サーバー環境

1.2.1.1 Linux 版 Elasticsearch の導入

導入に必要となる構成は以下の通りです。

構成種別	構成詳細
OS	CentOS/Redhat EnterpriseLinux 7.x/ Redhat EnterpriseLinux 8.x
Java ランタイム	OracleJDK/OpenJDK 1.8.0
連携ソフトウェア（高速検索）	Elasticsearch
連携ソフトウェア（形態素解析プラグイン）	Kuromoji
連携ソフトウェア（データ収集）	Logstash

1.2.1.2 Windows 版 Elasticsearch の導入

導入に必要となる構成は以下の通りです。

構成種別	構成詳細
OS	Microsoft Windows Server 2016/Microsoft Windows Server 2019/Microsoft Windows Server 2022
連携ソフトウェア（高速検索）	Elasticsearch
連携ソフトウェア（形態素解析プラグイン）	Kuromoji
連携ソフトウェア（データ収集）	Logstash

仕様補足 CPU プロセッサ数 4 以上、ディスク容量 300GB 以上、メモリ 8GB 以上（16GB 以上推奨）の環境で稼働させてください。
Senju Service Manager と Elasticsearch サーバーを同一筐体で稼働させる場合、CPU プロセッサ数 6 以上、ディスク容量 400GB 以上、メモリ 12GB 以上（20GB 以上推奨）の環境で稼働させてください。

仕様補足 詳しいサポートバージョンについては 「リリースノート」 を参照してください。

Microsoft Windows Server 2016 に導入する場合、事前に curl コマンドをインストールしてください。

1.2.2 その他の環境

クライアント、サーバー環境については、リリースノートを参照してください。

1.2.3 通信ポート

1.2.3.1 Linux 環境設定

ファイアウォールサービスについて、停止するか、「リリースノート」のポート情報を参考に該当するポートを開放して下さい。

Elasticsearch サーバーへのデフォルトアクセスポート：9200、9300 のポートを開放する場合の手順(“△” は半角スペースを示します。)

コマンド
firewall-cmd△--zone=public△--add-port=9200/tcp△--permanent
firewall-cmd△--zone=public△--add-port=9300/tcp△--permanent
firewall-cmd△--reload

1.2.3.2 Windows 版環境設定

1. 9200 ポートを開放

1. Windows スタート>コントロール パネル>システムとセキュリティ>Windows Defender ファイアウォールを選択し、「Windows Defender ファイアウォール」画面を表示します。

2. 左側の「詳細設定」をクリックし、「セキュリティが強化された　Windows Defender ファイアウォール」画面を表示します
3. 左側の「受信の規則」をクリックし、「受信の規則」画面を表示します。
4. 操作の「新しい規則...」をクリックし、「新規の受信の規則ウィザード」画面を表示します。
5. 規則の種類に「ポート（O）」を選択し、「次へ（N）>」をクリックします。
6. 「プロトコルおよびポート」画面を表示します。「TCP と UDP のどちらにこの規則を適用しますか？」に「TCP（T）」を選択し、「次へ（N）>」をクリックします。
7. 「特定のローカル　ポート（S）：」に「9200」を入力し、「次へ（N）>」をクリックします。
8. 「操作」画面を表示します。「接続を許可する（A）」を選択し、「次へ（N）>」をクリックします。
9. 「プロファイル」画面を表示します。「この規則はいつ適用しますか？」に全部を選択し、

「次へ（N）>」をクリックします。

10. 「名前」を画面表示します。「名前（N）」に任意値を入力し、「完了（F）」をクリックします。

2. 9300 ポートを開放

「9200 ポートを開放」の手順を参考に、9300 ポートを開放してください。

1.3 システム構成図

WEB ブラウザ（クライアント）、SSM WEB サーバー、SSM DB サーバー、Elasticsearch サーバー間の簡単なシステム構成図を図で示します。

- 通信（A）

Elasticsearch サーバー内の logstash が Senju Service Manager の Oracle DB または PostgreSQL DB からテキスト情報を取得します。

- 通信（B）

ユーザーは Senju Service Manager 利用時に全文検索または類似検索機能を使用します。

- 通信（C）

SSM WEB サーバーから送信されたリクエストを元に、Elasticsearch サーバーが検索結果を返却します。

- 通信（D）

Senju Service Manager 上で全文検索または類似検索の結果を確認できます。

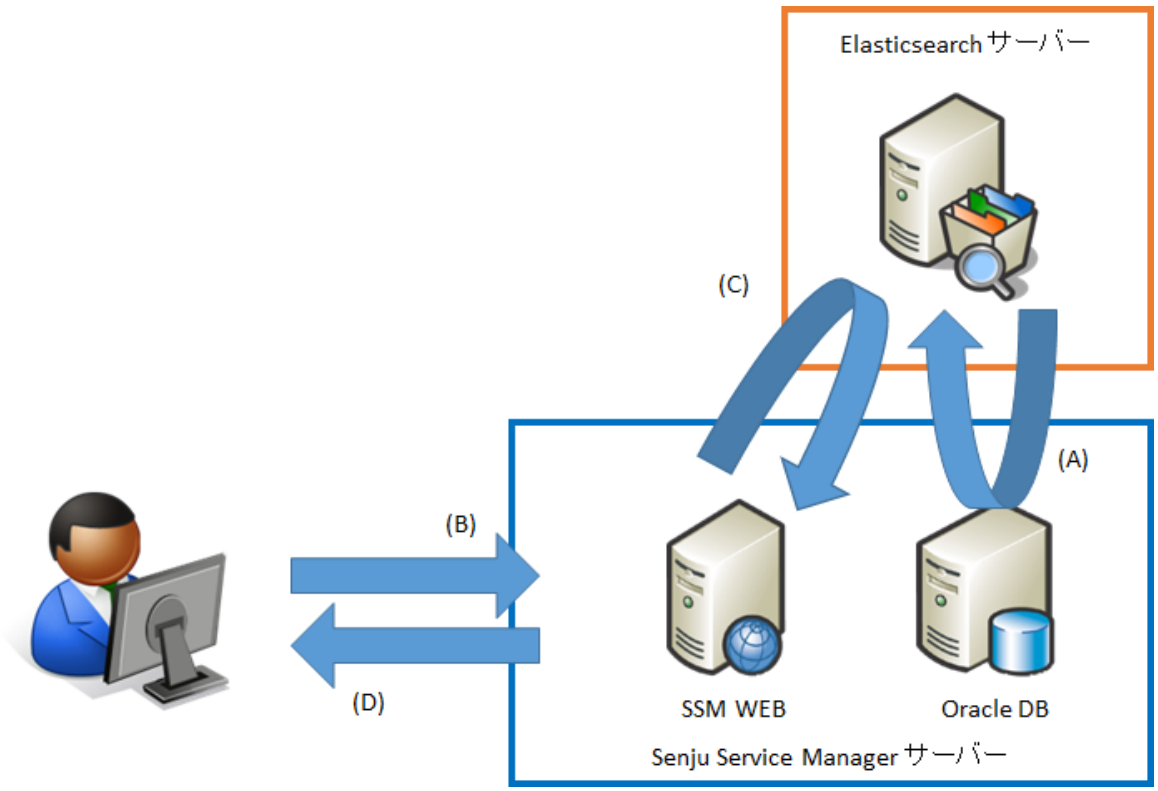

図 1-1 システム構成図

1.4 Linux版 Elasticsearchの導入

Linux サーバーに対して Elasticsearch の導入を行う場合の手順を説明します。

1.4.1 Elasticsearch 連携機能の導入

1.4.1.1 Elasticsearch の設定

Senju Service Manager と連携する Elasticsearch サーバーの稼働に必要なソフトウェアである、Elasticsearch をインストールするための手順について説明します。

1. 対象バージョン

Senju Service Manager と連携する Elasticsearch サーバーでは ElasticSearch 8.11 を使用します。

ここでは、ElasticSearch 8.11.3 を例として導入手順を説明します。

サポート対象となる Elasticsearch のバージョンについては リリースノート を参照してください。

2. Elasticsearch のインストール

1. インストール用の rpm ファイルを入手して、インストールするサーバーの任意のディレクトリに格納します。
   ・ファイル名: elasticsearch-8.11.3-x86_64.rpm

2. インストールするサーバーに管理者権限のアカウントでログインします。

3. rpm コマンドを実行してパッケージをインストールします。
   (“△” は半角スペースを示します。)

コマンド
rpm△-ivh△elasticsearch-8.11.3-x86_64.rpm

4. 自動起動の設定を行います。
   (“△” は半角スペースを示します。)

コマンド
systemctl△daemon-reload
systemctl△enable△elasticsearch.service

3. Elasticsearch の設定

1. Elasticsearch の設定ファイルを編集します。

   ファイルパス： /etc/elasticsearch/elasticsearch.yml

   (“△” は半角スペースを示します。)

コマンド
vi△/etc/elasticsearch/elasticsearch.yml

- 1 cluster.name を ssm-cluster に設定します。

```
#cluster.name: my-application
cluster.name: ssm-cluster
```

- 2 network.host に Elasticsearch サーバーのループバックアドレス、Elasticsearch サーバーの IP アドレスを指定します。

```
#network.host: 0.0.0.0
network.host: 127.0.0.1, eshost
```

デフォルトではループバックアドレス(127.0.0.1)が指定されるため、ループバックアドレス宛の Elasticsearch への接続しかできません。

- 3 discovery.seed_hosts に Elasticsearch サーバーの IP アドレスを指定します。

```
#discovery.seed_hosts: ["host1","host2"]
discovery.seed_hosts: ["127.0.0.1"]
```

- 4 cluster.initial_master_nodes に Elasticsearch サーバーのホスト名を指定します。

```
#cluster.initial_master_nodes: ["node-1","node-2"]
cluster.initial_master_nodes: ["eshostname"]
```

- 5 node.name に Elasticsearch サーバーのホスト名を指定します。

```
#node.name: node-1
node.name: eshostname
```

- 6 (Kibana を利用する場合)以下の記載を追加します。

```
script.max_size_in_bytes: 10000000
```

- 7 Elastic Stack のセキュリティを無効に設定します。

```
#xpack.security.enabled: true
xpack.security.enabled: false

#xpack.security.enrollment.enabled: true
xpack.security.enrollment.enabled: false
```

- 8 Elasticsearch とクライアント間通信の TLS 暗号化を無効に設定します。

```
xpack.security.http.ssl:
#  enabled: true
  enabled: false
#  keystore.path: certs/http.p12
```

- 9 Elasticsearch ノード間の暗号化・相互認証を無効に設定します。

```
xpack.security.transport.ssl:
#  enabled: true
  enabled: false
#  verification_mode: certificate
#  keystore.path: certs/transport.p12
#  truststore.path: certs/transport.p12
```

- 10 cluster.initial_master_nodes は設定済みのためコメントアウトします。

```
# Create a new cluster with the current node only
# Additional nodes can still join the cluster later
#cluster.initial_master_nodes: ["eshost"]
```

- 11 http.host、transport.host はコメントアウトします。

```
#http.host: 0.0.0.0
#transport.host: 0.0.0.0
```

仕様補足

Linux 版 Elasticsearch はインストール時にセキュリティ設定が行われ、Elasticsearch 設定ファイル末尾にセキュリティ設定が記載された状態となります。
セキュリティが有効な状態のため、一度セキュリティを無効に設定します。

2. Elasticsearch のログ設定ファイルを編集します。

ファイルパス： /etc/elasticsearch/log4j2.properties

(“△” は半角スペースを示します。)

コマンド
vi△/etc/elasticsearch/log4j2.properties

- 1 以下の設定を追加します。

```
appender.rolling.strategy.action.condition.type = IfLastModified
appender.rolling.strategy.action.condition.age = 7D
appender.rolling.strategy.action.PathConditions.type = IfFileName
appender.rolling.strategy.action.PathConditions.glob = ${sys:es.logs.cluster_name}-*
#appender.rolling.strategy.action.condition.type = IfFileName
#appender.rolling.strategy.action.condition.glob = ${sys:es.logs.cluster_name}-*
```

修正後のファイルは以下のようになります。

```
... (省略)
appender.rolling.policies.time.modulate = true
```

```
…(省略)
appender.rolling.strategy.type = DefaultRolloverStrategy
…(省略)
appender.rolling.strategy.action.type = Delete
appender.rolling.strategy.action.basepath = ${sys:es.logs.base_path}
appender.rolling.strategy.action.condition.type = IfLastModified
appender.rolling.strategy.action.condition.age = 7D
appender.rolling.strategy.action.PathConditions.type = IfFileName
appender.rolling.strategy.action.PathConditions.glob = ${sys:es.logs.cluster_name}-*
#appender.rolling.strategy.action.condition.type = IfFileName
#appender.rolling.strategy.action.condition.glob = ${sys:es.logs.cluster_name}-*
…(省略)
rootLogger.level = info
…(省略)
```

この設定を追加することで、7 日間変更がない古いログを削除するようになります。ログ削除の設定を追加しない場合、デフォルトでは古いログファイルが残り続けるため、 /var/log/Elasticsearch ディレクトリのディスク容量を圧迫してしまう可能性があります。

3. Elasticsearch の HEAP メモリ拡張を編集します。

   ファイルパス： /etc/elasticsearch/jvm.options

   修正前

```
…(省略)
## -Xms4g
## -Xmx4g
…(省略)
```

   修正後

   例：

```
…(省略)
-Xms2g
-Xmx2g
…(省略)
```

4. インストール確認

1. Elasticsearch を起動します。

   (“△” は半角スペースを示します。)

コマンド
systemctl△start△elasticsearch

2. 以下のコマンドを実行して稼働していることを確認します。

(“△” は半角スペースを示します。)

コマンド
curl△-XGET△'http://localhost:9200/?pretty'

図 1-2 稼働確認

起動に失敗する場合は以下を確認してください。
・インストール環境の残メモリが少ないと起動に失敗します。Linux のシステムログを参照して Elasticsearch が停止されていないか確認してください。
・インストール環境でプログラムが多く起動していると OS のスレッド上限値に抵触して起動できない場合があります。上限値について見直して下さい。

1.4.1.2 Elasticsearch の検索設定

Senju Service Manager と連携する Elasticsearch サーバーで形態素解析に必要なソフトウェアである、Kuromoji をインストールするための手順について説明します。

1. 対象バージョン

Senju Service Manager と連携する Elasticsearch サーバーでは Kuromoji 8.11 を使用します。
ここでは、Kuromoji 8.11.3 を例として導入手順を説明します。

仕様補足
サポート対象となる Kuromoji のバージョンについては リリースノート を参照してください。

2. Kuromoji のインストール

1. インストール用の zip ファイルを入手して、インストールするサーバーの任意のディレクトリに格納します。

   ・ファイル名: analysis-kuromoji-8.11.3.zip

   以降では、上記ファイルを下記ディレクトリに格納した場合を例示いたします。

   ・ディレクトリパス: /home/root/tmp/elasticsearch/

2. インストールするサーバーに管理者権限のアカウントでログインします。

3. install コマンドを実行してパッケージをインストールします。

   (“△” は半角スペースを示します。)

コマンド
/usr/share/elasticsearch/bin/elasticsearch-plugin△install△ file:///home/root/tmp/elasticsearch/analysis-kuromoji-8.11.3.zip

3. サーバーの再起動

1. Elasticsearch サーバーの OS の再起動を行います。

4. インストール確認

2. Kuromoji（形態素解析ソフトウェアプラグイン）がインストールされていることを確認します。

   (“△” は半角スペースを示します。)

コマンド
/usr/share/elasticsearch/bin/elasticsearch-plugin△list△analysis-kuromoji

```
# /usr/share/elasticsearch/bin/elasticsearch-plugin list analysis-kuromoji
analysis-kuromoji
```

図 1-3 インストール確認

5. データクローラの設定

1. Senju Service Manager のインストールメディアから、elasticsearch-definitions フォルダの Elasticsearch 設定ファイルを取得し、インストールするサーバーの任意のディレクトリに格納します。

ファイル名	説明
pipeline_attachment.json	パイプライン設定（添付ファイル）
sm_mappings.json	Index Template（プロセス管理）

ファイル名	説明
sm_mappings_ci.json	Index Template（構成管理）
sm_mappings_faq.json	Index Template（FAQ）
sm_mappings_filelibrary.json	Index Template（ファイルライブラリ）

以降では、上記ファイルを下記ディレクトリに格納した場合を例示いたします。

・ディレクトリパス: /tmp/elasticsearch-definitions/

2. curl コマンドを使用して、パイプライン設定を Elasticsearch サーバーに登録します。
(“△” は半角スペースを示します。)

コマンド
curl△-H△"Content-Type:application/json"△-XPUT△ 'http://localhost:9200/_ingest/pipeline/attachment?pretty'△--data-binary△ @/tmp/elasticsearch-definitions/pipeline_attachment.json

3. 以下のレスポンスが返ってきており、登録が成功したことを確認します。

```
{
"acknowledged" : true
}
```

4. curl コマンドを使用して、IndexTemplate を Elasticsearch サーバーに登録します。
プロセス管理の場合(“△” は半角スペースを示します。)

コマンド
curl△-H△"Content-Type:application/json"△-XPUT△ 'http://localhost:9200/_template/doc?pretty'△--data-binary△@/tmp/elasticsearch- definitions/sm_mappings.json

構成管理の場合(“△” は半角スペースを示します。)

コマンド
curl△-H△"Content-Type:application/json"△-XPUT△ 'http://localhost:9200/_template/doc_ci?pretty'△--data-binary△@/tmp/elasticsearch- definitions/sm_mappings_ci.json

FAQ の場合(“△” は半角スペースを示します。)

コマンド
curl△-H△"Content-Type:application/json"△-XPUT△ 'http://localhost:9200/_template/doc_faq?pretty'△--data-binary△@/tmp/elasticsearch- definitions/sm_mappings_faq.json

ファイルライブラリの場合(“△” は半角スペースを示します。)

コマンド
curl△-H△"Content-Type:application/json"△-XPUT△ 'http://localhost:9200/_template/doc_filelibrary?pretty'△--data-binary△@/tmp/elasticsearch- definitions/sm_mappings_filelibrary.json

5. 以下のレスポンスが返ってきており、登録が成功したことを確認します。

```
{
"acknowledged" : true
}
```

プロセス管理の場合

```
# curl -H "Content-Type:application/json" -XPUT 'http://localhost:9200/_template/doc?p
retty' --data-binary @/tmp/elasticsearch-definitions/sm_mappings.json
{
  "acknowledged" : true
}
```

図 1-4 インデックステンプレート登録

仕様補足

IndexTemplate は Elasticsearch 上にインデックスを作成する際、特定ネーミングルールに沿ったインデックスに対してテンプレートを適応します。Senju Service Manager では様々なインデックス名のインデックスが作成されるためすべてのインデックスに対してテンプレートを適応するようにしています。そのため、Elasticsearch を Senju Service Manager 以外の用途に利用できません。

6. インデックスの作成

1. curl コマンドを使用して、Index を作成します。

プロセス管理の場合(“△” は半角スペースを示します。)

コマンド
curl△-XPUT△'http://localhost:9200/ssm?pretty'

構成管理の場合(“△” は半角スペースを示します。)

コマンド
curl△-XPUT△'http://localhost:9200/ssm_ci?pretty'

FAQ の場合(“△” は半角スペースを示します。)

コマンド
curl△-XPUT△'http://localhost:9200/ssm_faq?pretty'

ファイルライブラリの場合(“△” は半角スペースを示します。)

コマンド
curl△-XPUT△'http://localhost:9200/ssm_filelibrary?pretty'

2. 以下のレスポンスが返ってきており、登録が成功したことを確認します。

プロセス管理の場合

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
```

```
  "index" : "ssm"
}
```

1.4.1.3 Elasticsearch へのデータ連携設定

Senju Service Manager から Elasticsearch へのデータ連携のためのアプリケーションとして Elasticsearch が提供する Logstach を導入します。

1. Logstash のインストール

1. インストール用の rpm ファイルを入手して、インストールするサーバーの任意のディレクトリに格納します。
   ・ファイル名: logstash-8.11.3-x86_64.rpm

2. インストールするサーバーに管理者権限のアカウントでログインします。

3. rpm コマンドを実行してパッケージをインストールします。
   (“△” は半角スペースを示します。)

コマンド
rpm△-ivh△logstash-8.11.3-x86_64.rpm

4. 自動起動の設定を行います。
   (“△” は半角スペースを示します。)

コマンド
systemctl△daemon-reload
systemctl△enable△logstash

2. Logstash の設定

1. Senju Service Manager のインストールメディアから、logstash・logstash-definitions フォルダ一式をインストールするサーバーの任意のディレクトリに格納します。
   ・フォルダ名: logstash
   ・フォルダ名: logstash-definitions
   以降では、上記フォルダを下記ディレクトリに格納した場合を例示いたします。
   ・ディレクトリパス: /tmp/

2. インストールするサーバーに管理者権限のアカウントでログインします。

3. logstash ディレクトリをコピーして/opt 配下に格納し、/opt/ logstash 配下に履歴ファイルの格納フォルダ conf を追加し、所有者およびグループを logstash ユーザーに変更します。以下のコマンドを実行してください。
("△" は半角スペースを示します。)

コマンド
cp△-fr△/tmp/logstash△/opt
mkdir△-p△/opt/logstash/conf
chown△-R△logstash:logstash△/opt/logstash

4. Logstash 設定ファイルを /etc/logstash/conf.d 配下に格納し、所有者およびグループを logstash ユーザーに変更します。以下のコマンドを実行してください。
("△" は半角スペースを示します。)

・データベースが Oracle である場合:

コマンド
cp△/tmp/logstash-definitions/logstash-oracle*.conf△/etc/logstash/conf.d
chown△logstash:logstash△/etc/logstash/conf.d/logstash-oracle*.conf

・ データベースが PostgreSQL である場合:

コマンド
cp△/tmp/logstash-definitions/logstash-postgresql*.conf△/etc/logstash/conf.d
chown△logstash:logstash△/etc/logstash/conf.d/logstash-postgresql*.conf

5. 設定ファイルを以下の通り更新します。
・ データベースが Oracle である場合:

ファイル名	説明
logstash-oracle.conf	設定ファイル(プロセス管理)
logstash-oracle_ci.conf	設定ファイル(構成管理)
logstash-oracle-faq.conf	設定ファイル(FAQ)
logstash-oracle-filelibrary.conf	設定ファイル(ファイルライブラリ)

プロセス管理の場合("△" は半角スペースを示します。)

コマンド
vi△/etc/logstash/conf.d/logstash-oracle.conf

・データベースが PostgreSQL である場合:

ファイル名	説明
logstash-postgresql.conf	設定ファイル(プロセス管理)
logstash-postgresql_ci.conf	設定ファイル(構成管理)
logstash-postgresql-faq.conf	設定ファイル(FAQ)

ファイル名	説明
logstash-postgresql-filelibrary.conf	設定ファイル（ファイルライブラリ）

プロセス管理の場合(“△” は半角スペースを示します。)

コマンド
vi△/etc/logstash/conf.d/logstash-postgresql.conf

1) jdbc_connection_string に記載されている Oracle DB もしくは PostgreSQL DB への接続情報について、[hostname]:[portnumber]/[dbname] から、SSM DB サーバーの ホスト名:ポート番号/ローカル・ネット・サービス名に変更します。

・データベースが Oracle である場合:

（Oracle DB への接続情報をホスト名:ccfsphost、ポート番号:1522、ローカル・ネット・サービス名:ssmdb に変更する場合）

```
# jdbc_connection_string => "jdbc:oracle:thin:@[hostname]:[portnumber]/[dbname]"
jdbc_connection_string => "jdbc:oracle:thin:@ccfsphost:1522/ssmdb"
```

・データベースが PostgreSQL である場合:

（PostgreSQL　DB への接続情報をホスト名:ccfsphost、ポート番号:5432、ローカル・ネット・サービス名:ssmdb に変更する場合）

```
# jdbc_connection_string => "jdbc:postgresql://[hostname]:[portnumber]/[dbname]"
jdbc_connection_string => " jdbc:postgresql://ccfsphost:5432/ssmdb"
```

2) jdbc_user に記載されている DB ユーザー名について、[username]から、正しいに変更します。

（Oracle DB または PostgreSQL DB の DB ユーザー名を DB ユーザー名:ssmuser に変更する場合）

```
# jdbc_user => "[username]"
jdbc_user => "ssmuser"
```

3) jdbc_password に記載されている DB ユーザーのパスワードについて、[password]から、正しいパスワードに変更します。

（Oracle DB または PostgreSQL DB ユーザーのパスワードを DB ユーザーパスワード:ssmpwd に変更する場合）

```
#jdbc_password => "[password]"
jdbc_password => "ssmpwd"
```

4) hosts に記載されている Elasticsearch への接続情報について、[protocol]://[hostname]:[portnumber] から、Elasticsearch サーバーのプロトコル://ホスト名:ポート番号に変更します。

（Elasticsearch への接続情報をプロトコル:http、ホスト名:eshost、ポート番号:9200 に変更する場合）

```
#hosts => [ "[protocol]://[hostname]:[portnumber]"]
hosts => [ "http://eshost:9200"]
```

修正後のファイルは以下のようになります。

・データベースが Oracle である場合:

・logstash-oracle.conf

```
...（省略）
input {
  # for Upsert PROCESS_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:oracle:thin:@ccfsphost:1522/ssmdb"
    jdbc_driver_library    => "/opt/logstash/lib/ojdbc7.jar"
    jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"

    schedule               => "* * * * *"
    last_run_metadata_path => "/opt/logstash/conf/.logstash_oracle_process_fil_last_run"
    record_last_run        => "true"
    use_column_value       => "true"
    tracking_column        => "update_ts"
    statement_filepath     =>
"/opt/logstash/sql/oracle/get_sm_data_from_process_fil.sql"

    type                   => "get_process_fil"
  }

  # for Upsert PROCESS_SUB_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:oracle:thin:@ccfsphost:1522/ssmdb"
    jdbc_driver_library    => "/opt/logstash/lib/ojdbc7.jar"
    jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"

...（省略）
  }

  # for Upsert PROCESS_TABLE_ITEM_1_FIL records
  jdbc {
    jdbc_connection_string => " jdbc:oracle:thin:@ccfsphost:1522/ssmdb "
    jdbc_driver_library    => "/opt/logstash/lib/ojdbc7.jar"
    jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"

...（省略）
  }
```

```
  # for Upsert PROCESS_TABLE_ITEM_2_FIL records
 jdbc {
  jdbc_connection_string => " jdbc:oracle:thin:@ccfsphost:1522/ssmdb "
  jdbc_driver_library    => "/opt/logstash/lib/ojdbc7.jar"
  jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
  jdbc_user              => "ssmuser"
  jdbc_password          => "ssmpwd"

...(省略)
 }

  # for Upsert PROCESS_TABLE_ITEM_3_FIL records
 jdbc {
  jdbc_connection_string => " jdbc:oracle:thin:@ccfsphost:1522/ssmdb "
  jdbc_driver_library    => "/opt/logstash/lib/ojdbc7.jar"
  jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
  jdbc_user              => "ssmuser"
  jdbc_password          => "ssmpwd"

...(省略)
 }

  # for Upsert PROCESS_TABLE_ITEM_4_FIL records
 jdbc {
  jdbc_connection_string => " jdbc:oracle:thin:@ccfsphost:1522/ssmdb "
  jdbc_driver_library    => "/opt/logstash/lib/ojdbc7.jar"
  jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
  jdbc_user              => "ssmuser"
  jdbc_password          => "ssmpwd"

...(省略)
 }

  # for Upsert PROCESS_TABLE_ITEM_5_FIL records
 jdbc {
  jdbc_connection_string => " jdbc:oracle:thin:@ccfsphost:1522/ssmdb "
  jdbc_driver_library    => "/opt/logstash/lib/ojdbc7.jar"
  jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
  jdbc_user              => "ssmuser"
  jdbc_password          => "ssmpwd"
...(省略)

output {
 if [type] == "get_process_fil" {
  elasticsearch{
   hosts             => [ "http://eshost:9200"]
   index             => "ssm"
   document_id       => "%{insert_no}"
   action            => "update"
   retry_on_conflict => 10
   doc_as_upsert     => true
  # user              => "[elasticusername]"
  # password          => "[elasticpassword]"
```

```
    # ssl_enabled       => true
    # ssl_verification_mode => none
    # ssl_keystore_path => "[keystore_path]"
    # ssl_keystore_password => "[keystore_password]"
    }
  } else if [type] == "get_process_sub_fil" {
    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_1_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_2_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_3_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_4_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_5_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
```

・logstash-oracle_ci.conf

```
... (省略)
input {
  # for Update CI_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:oracle:thic:@ccfsphost:1522/ssmdb"
    jdbc_driver_library    => "/opt/logstash/lib/ojdbc7.jar"
    jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"

    schedule               => "* * * * *"
    last_run_metadata_path =>
"/opt/logstash/conf/.logstash_oracle_process_sub_fil_last_run"
    record_last_run        => "true"
    use_column_value       => "true"
    tracking_column        => "update_ts"
```

```
    statement_filepath     =>
"/opt/logstash/sql/oracle/get_sm_data_from_process_sub_fil.sql"

    type                   => "get_ci_fil"
  }
}
...（省略）

output {
  if [type] == "get_ci_fil" {
    elasticsearch{
      hosts             => [ " eshost:9200"]
      index             => "ssm_ci"
      document_id       => "%{ci_id}"
      action            => "update"
      retry_on_conflict => 10
      doc_as_upsert     => true
    # user              => "[elasticusername]"
    # password          => "[elasticpassword]"
...（省略）
```

・データベースが PostgreSQL である場合:

・logstash-postgresql.conf

```
...（省略）
input {
  # for Upsert PROCESS_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:postgresql://ccfsphost:5432/ssmdb"
    jdbc_driver_library    => "/opt/logstash/lib/postgresql-42.2.8.jar"
    jdbc_driver_class      => "org.postgresql.Driver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"

    schedule               => "* * * * *"
    last_run_metadata_path => "/opt/logstash/conf/.logstash_oracle_process_fil_last_run"
    record_last_run        => "true"
    use_column_value       => "true"
    tracking_column        => "update_ts"
    statement_filepath     =>
"/opt/logstash/sql/postgresql/get_sm_data_from_process_fil.sql"

    type                   => "get_process_fil"
  }

  # for Upsert PROCESS_SUB_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:postgresql://ccfsphost:5432/ssmdb"
    jdbc_driver_library    => "/opt/logstash/lib/postgresql-42.2.8.jar"
    jdbc_driver_class      => "org.postgresql.Driver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"
```

```
...(省略)
 }

 # for Upsert PROCESS_TABLE_ITEM_1_FIL records
 jdbc {
  jdbc_connection_string => "jdbc:postgresql://ccfsphost:5432/ssmdb"
  jdbc_driver_library    => "/opt/logstash/lib/postgresql-42.2.8.jar"
  jdbc_driver_class      => "org.postgresql.Driver"
  jdbc_user              => "ssmuser"
  jdbc_password          => "ssmpwd"

...(省略)
 }

 # for Upsert PROCESS_TABLE_ITEM_2_FIL records
 jdbc {
  jdbc_connection_string => "jdbc:postgresql://ccfsphost:5432/ssmdb"
  jdbc_driver_library    => "/opt/logstash/lib/postgresql-42.2.8.jar"
  jdbc_driver_class      => "org.postgresql.Driver"
  jdbc_user              => "ssmuser"
  jdbc_password          => "ssmpwd"

...(省略)
 }

 # for Upsert PROCESS_TABLE_ITEM_3_FIL records
 jdbc {
  jdbc_connection_string => "jdbc:postgresql://ccfsphost:5432/ssmdb"
  jdbc_driver_library    => "/opt/logstash/lib/postgresql-42.2.8.jar"
  jdbc_driver_class      => "org.postgresql.Driver"
  jdbc_user              => "ssmuser"
  jdbc_password          => "ssmpwd"

...(省略)
 }

 # for Upsert PROCESS_TABLE_ITEM_4_FIL records
 jdbc {
  jdbc_connection_string => "jdbc:postgresql://ccfsphost:5432/ssmdb"
  jdbc_driver_library    => "/opt/logstash/lib/postgresql-42.2.8.jar"
  jdbc_driver_class      => "org.postgresql.Driver"
  jdbc_user              => "ssmuser"
  jdbc_password          => "ssmpwd"

...(省略)
 }

 # for Upsert PROCESS_TABLE_ITEM_5_FIL records
 jdbc {
  jdbc_connection_string => "jdbc:postgresql://ccfsphost:5432/ssmdb"
  jdbc_driver_library    => "/opt/logstash/lib/postgresql-42.2.8.jar"
  jdbc_driver_class      => "org.postgresql.Driver"
  jdbc_user              => "ssmuser"
```

```
    jdbc_password          => "ssmpwd"
... (省略)

output {
  if [type] == "get_process_fil" {
    elasticsearch{
      hosts             => [ "http://eshost:9200"]
      index             => "ssm"
      document_id       => "%{insert_no}"
      action            => "update"
      retry_on_conflict => 10
      doc_as_upsert     => true
    # user              => "[elasticusername]"
    # password          => "[elasticpassword]"
    }
  } else if [type] == "get_process_sub_fil" {
    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_1_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_2_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_3_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_4_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
  } else if [type] == "get_process_table_item_5_fil" {

    elasticsearch{
      hosts             => [ "http://eshost:9200"]
... (省略)
```

・logstash-postgresql_ci.conf

```
... (省略)
input {
  # for Update CI_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:postgresql:// ccfsphost:5432/ssmdb"
    jdbc_driver_library    => "/opt/logstash/lib/postgresql-42.2.8.jar"
```

```
    jdbc_driver_class      => "org.postgresql.Driver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"

    schedule               => "* * * * *"
    last_run_metadata_path =>
"/opt/logstash/conf/.logstash_postgresql_process_sub_fil_last_run"
    record_last_run        => "true"
    use_column_value       => "true"
    tracking_column        => "update_ts"
    statement_filepath     =>
"/opt/logstash/sql/postgresql/get_sm_data_from_process_sub_fil.sql"

    type                   => "get_ci_fil"
  }
}
…(省略)

output {
  if [type] == "get_ci_fil" {
    elasticsearch{
      hosts              => [ " eshost:9200"]
      index              => "ssm_ci"
      document_id        => "%{ci_id}"
      action             => "update"
      retry_on_conflict => 10
      doc_as_upsert      => true
    # user               => "[elasticusername]"
    # password           => "[elasticpassword]"
…(省略)
```

6. 設定ファイル pipelines.yml を以下の通り更新します。

   ファイルパス:/etc/logstash/pipelines.yml

   (“△” は半角スペースを示します。)

コマンド
vi△/etc/logstash/pipelines.yml

- 1 pipeline の設定を行います。

項目	設定値
pipeline.id	(インデックス名)
pipeline.workers	1
pipeline.batch.size	1
path.config	(設定ファイルのパス)
queue.type	persisted

```
# - pipeline.id: main
#   path.config: "/etc/logstash/conf.d/*.conf"
- pipeline.id: ssm
```

```
  pipeline.workers: 1
  pipeline.batch.size: 1
  path.config: "/etc/logstash/conf.d/logstash-oracle.conf"
  queue.type: persisted
- pipeline.id: ssm_faq
  pipeline.workers: 1
  pipeline.batch.size: 1
  queue.type: persisted
  path.config: "/etc/logstash/conf.d/logstash-oracle-faq.conf"
- pipeline.id: ssm_filelibrary
  pipeline.workers: 1
  pipeline.batch.size: 1
  queue.type: persisted
  path.config: "/etc/logstash/conf.d/logstash-oracle-filelibrary.conf"
- pipeline.id: ssm_ci
  pipeline.workers: 1
  pipeline.batch.size: 1
  queue.type: persisted
  path.config: "/etc/logstash/conf.d/logstash-oracle_ci.conf"
```

仕様補足

この設定を変更しないと、異常発生時に logstash のデータ登録イベントが抜けてしまう恐れが発生します。

7. Logstash のログ設定ファイル log4j2.properties を編集します。

   ファイルパス:/etc/logstash/log4j2.properties

   (“△” は半角スペースを示します。)

コマンド
vi△/etc/logstash/log4j2.properties

- 1 以下の設定を追加します。

```
appender.rolling.strategy.action.type = Delete
appender.rolling.strategy.action.basepath = ${sys:ls.logs}
appender.rolling.strategy.action.condition.type = IfLastModified
appender.rolling.strategy.action.condition.age = 7D
appender.rolling.strategy.action.PathConditions.type = IfFileName
appender.rolling.strategy.action.PathConditions.glob = logstash-${sys:ls.log.format}-*
```

修正後のファイルは以下のようになります。

```
... (省略)
appender.rolling.strategy.type = DefaultRolloverStrategy
... (省略)
appender.rolling.strategy.action.type = Delete
appender.rolling.strategy.action.basepath = ${sys:ls.logs}
appender.rolling.strategy.action.condition.type = IfLastModified
appender.rolling.strategy.action.condition.age = 7D
appender.rolling.strategy.action.PathConditions.type = IfFileName
appender.rolling.strategy.action.PathConditions.glob = logstash-${sys:ls.log.format}-*
... (省略)
```

仕様補足

この設定を追加することで、7日間変更がない古いログを削除するようになります。ログ削除の設定を追加しない場合、デフォルトでは古いログファイルが残り続けるため/var/log/logstash ディレクトリのディスク容量を圧迫してしまう可能性があります。

3. インストール確認

1. Logstash を起動します。

   (“△” は半角スペースを示します。)

コマンド
systemctl△start△logstash

2. 以下のコマンドを実行して、logstash サービスが正常に動作していることを確認します。

   (“△” は半角スペースを示します。)

コマンド
systemctl△status△logstash

図 1-5 稼働確認

1.4.1.4 Elasticsearch の max_analyzed_offset 設定

Senju Service Manager と連携する Elasticsearch サーバーの、max_analyzed_offset を設定します。

1. curl コマンドを実行して max_analyzed_offset を設定します。

   (“△” は半角スペースを示します。)

コマンド
curl△-XPUT△"http://localhost:9200/_settings?pretty"△-H△'Content-Type:application/json'△-d '{"index":{"highlight.max_analyzed_offset":51000000}}'

   以下のレスポンスが返ってきており、設定が成功したことを確認します。

```
{
"acknowledged" : true
}
```

2. Elasticsearch と Logstash を再起動します。

   (“△” は半角スペースを示します。)

コマンド
systemctl△restart△elasticsearch
systemctl△restart△logstash

1.4.2 ウィルススキャンの除外設定

Elasticsearch が稼働している環境において、アンチウィルスソフトのようなセキュリティ関連ソフトや、バックアップソフト等の予期しない動作により、パフォーマンスの影響や動作不調を起こす場合があります。
そのため、アプリケーションやミドルウェアのフォルダやファイルをリアルタイム検索から除外していただく必要があります。

各アンチウィルスソフトの除外設定手順に従い、設定を行ってください。

対象は 資料集「1.1.1 Senju Service Manager システム」の 9)を参照してください。

1.5 Windows版 Elasticsearchの導入

Windows サーバーに対して Elasticsearch の導入を行う場合の手順を説明します。

1.5.1 Elasticsearch の設定

Senju Service Manager と連携する Elasticsearch サーバーの稼働に必要なソフトウェアである、Elasticsearch をインストールするための手順について説明します。

1. 対象バージョン

Senju Service Manager と連携する Elasticsearch サーバーでは ElasticSearch 8.11 を使用します。

ここでは、ElasticSearch 8.11.3 を例として導入手順を説明します。

仕様補足

サポート対象となる Elasticsearch のバージョンについては リリースノート を参照してください。

2. Elasticsearch のインストール

1. elasticsearch-8.11.3.zip を入手し稼働させたいディレクトリに展開します。
   ※以下、展開したディレクトリを%elasticsearch_home%と記載して説明します。

2. Elasticsearch の設定ファイルを編集します。
   %elasticsearch_home%\config\elasticsearch.yml
3. cluster.name を ssm-cluster に設定します。

```
#cluster.name: my-application
cluster.name: ssm-cluster
```

4. network.host に Elasticsearch サーバーのループバックアドレス、Elasticsearch サーバーのホスト名もしくは IP アドレスを指定します。

```
#network.host: 0.0.0.0
network.host: 127.0.0.1, eshost
```

仕様補足

デフォルトではすべての接続を許可するため、ループバックアドレスおよび Elasticsearch サーバーのホスト名もしくは IP アドレスを指定することで、不正なアクセスを拒否しています。

5. cluster.initial_master_nodes に Elasticsearch サーバーのホスト名を指定します。

```
#cluster.initial_master_nodes: ["node-1","node-2"]
cluster.initial_master_nodes: ["eshostname"]
```

6. node.name に Elasticsearch サーバーのホスト名を指定します。

```
#node.name: node-1
```

```
node.name: eshostname
```

7. (Kibana を利用する場合)以下の記載を追加します。

```
script.max_size_in_bytes: 10000000
```

8. GeoIp の自動更新を false に設定します。

```
ingest.geoip.downloader.enabled: false
```

9. Elastic Stack のセキュリティ機能を無効(false)に設定します。

```
xpack.security.enabled: false
```

10. Elasticsearch のログ設定ファイルを編集します。

    %elasticsearch_home%\config\log4j2.properties

    以下の設定をファイルの appender.rolling.policies.time.modulate の後に追加します。

```
appender.rolling.strategy.type = DefaultRolloverStrategy
appender.rolling.strategy.action.type = Delete
appender.rolling.strategy.action.basepath = ${sys:es.logs.base_path}
appender.rolling.strategy.action.condition.type = IfLastModified
appender.rolling.strategy.action.condition.age = 7D
appender.rolling.strategy.action.PathConditions.type = IfFileName
appender.rolling.strategy.action.PathConditions.glob = ${sys:es.logs.cluster_name}-*
```

    修正後のファイルが以下の通りになっていることを確認します

```
... (省略)
appender.rolling.policies.time.modulate = true
appender.rolling.strategy.type = DefaultRolloverStrategy
appender.rolling.strategy.action.type = Delete
appender.rolling.strategy.action.basepath = ${sys:es.logs.base_path}
appender.rolling.strategy.action.condition.type = IfLastModified
appender.rolling.strategy.action.condition.age = 7D
appender.rolling.strategy.action.PathConditions.type = IfFileName
appender.rolling.strategy.action.PathConditions.glob = ${sys:es.logs.cluster_name}-*
appender.rolling.policies.size.type = SizeBasedTriggeringPolicy
... (省略)
```

11. Elasticsearch の HEAP メモリ拡張を編集します。

    %elasticsearch_home%\config\ jvm.options

    修正前

```
... (省略)
## -Xms4g
## -Xmx4g
... (省略)
```

    修正後

例：

```
…(省略)
-Xms4g
-Xmx4g
…(省略)
```

仕様補足

-Xms および -Xmx は：物理メモリサイズの半分を推奨します。
例：8GB の物理メモリを搭載している場合 4GB (4g) を設定してください。

3. Elasticsearch のインストール確認

1. コマンドプロンプトを開いて、以下のコマンドを実行し、Elasticsearch を起動します。
（“△” は半角スペースを示します。）

コマンド
cd△%elasticsearch_home%¥bin elasticsearch.bat

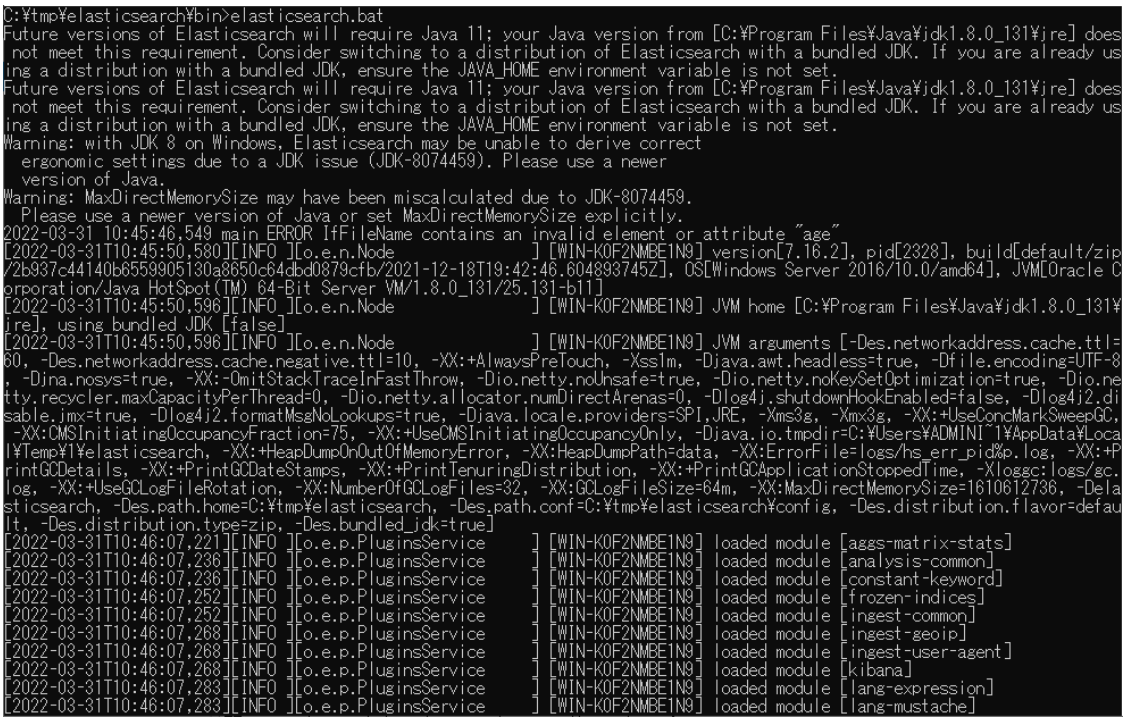

図 1-6 Elasticsearch のインストール

2. "started"という文言が表示されたら、もう一つコマンドプロンプトを開いて、下記コマンドを実行します。
（“△” は半角スペースを示します。）

コマンド
cd△%elasticsearch_home%¥bin curl△-H△"Content-Type:△application/json"△-XGET△"http://localhost:9200/?pretty"

以下のような応答があることを確認します。

```
...（省略）
"cluster_name":"ssm-cluster",
...（省略）
```

図 1-7 インストール の確認

1.5.2 ElasticSearch の検索設定

Senju Service Manager と連携する Elasticsearch サーバーで形態素解析に必要なソフトウェアである、Kuromoji をインストールするための手順について説明します。

1. 対象バージョン

Senju Service Manager と連携する Elasticsearch サーバーでは Kuromoji 8.11 を使用します。
ここでは、Kuromoji 8.11.3 を例として導入手順を説明します。

仕様補足

サポート対象となる Kuromoji のバージョンについては リリースノート を参照してください。

2. ElasticSearch の検索設定

1. analysis-kuromoji-8.11.3.zip を入手し任意のディレクトリに展開します。

2. %elasticsearch_home%\plugins\配下に以下の名称のフォルダを作成してください。
   フォルダ名：analysis-kuromoji

3. analysis-kuromoji-8.11.3 配下のファイルを作成したフォルダにコピー上書きしてください。

4. コマンドプロンプトを開いて、以下のコマンドを実行し、Elasticsearch を再起動します。
   (“△” は半角スペースを示します。)

コマンド
cd△%elasticsearch_home%¥bin elasticsearch.bat

3. インストール確認

1. Kuromoji（形態素解析ソフトウェアプラグイン）がインストールされていることを確認します。(“△” は半角スペースを示します。)

コマンド
cd△%elasticsearch_home%¥bin elasticsearch-plugin△list△analysis-kuromoji

以下のような出力結果があることを確認します。

analysis-kuromoji

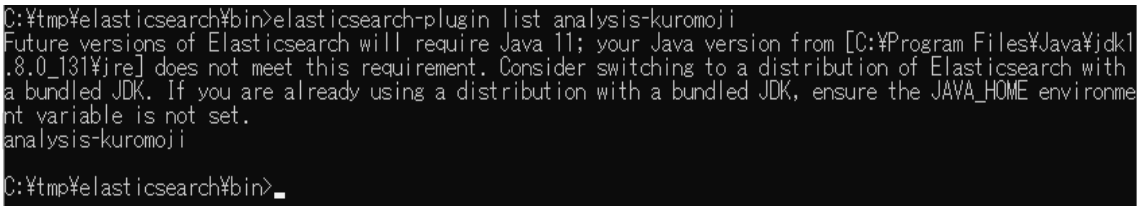

図 1-8 Kuromoji 確認

4. データクローラの設定

1. Senju Service Manager のインストールメディアから、elasticsearch-definitions フォルダの Elasticsearch 設定ファイルを取得し、インストールするサーバーの任意のディレクトリに格納します。

ファイル名	説明
pipeline_attachment.json	パイプライン設定（添付ファイル）
sm_mappings.json	Index Template（プロセス管理）
sm_mappings_ci.json	Index Template（構成管理）
sm_mappings_faq.json	Index Template（FAQ）
sm_mappings_filelibrary.json	Index Template（ファイルライブラリ）

以降では、上記ファイルを下記ディレクトリに格納した場合を例示いたします。

・ディレクトリパス: %elasticsearch_home%\elasticsearch-definitions\

2. curl コマンドを使用して、パイプライン設定を Elasticsearch サーバーに登録します。
("△" は半角スペースを示します。)

コマンド
curl△-H "Content-Type: application/json"△-XPUT△ "http://localhost:9200/_ingest/pipeline/attachment?pretty"△--data-binary△ @%elasticsearch_home%¥elasticsearch-definitions¥pipeline_attachment.json

3. 以下のレスポンスが返ってきており、登録が成功したことを確認します。

```
{
  "acknowledged" : true
}
```

4. curl コマンドを使用して、IndexTemplate を Elasticsearch サーバーに登録します。
プロセス管理の場合(“△” は半角スペースを示します。)

コマンド
curl△-H "Content-Type: application/json"△-XPUT△ "http://localhost:9200/_template/doc?pretty"△--data-binary△ @%elasticsearch_home%¥elasticsearch-definitions¥sm_mappings.json

構成管理の場合(“△” は半角スペースを示します。)

コマンド
curl△-H "Content-Type: application/json"△-XPUT△ "http://localhost:9200/_template/doc_ci?pretty"△--data-binary△ @%elasticsearch_home%¥elasticsearch-definitions¥sm_mappings_ci.json

FAQ の場合(“△” は半角スペースを示します。)

コマンド
curl△-H "Content-Type: application/json"△-XPUT△ "http://localhost:9200/_template/doc_faq?pretty"△--data-binary△ @%elasticsearch_home%¥elasticsearch-definitions¥sm_mappings_faq.json

ファイルライブラリの場合(“△” は半角スペースを示します。)

コマンド
curl△-H "Content-Type: application/json"△-XPUT△ "http://localhost:9200/_template/doc_filelibrary?pretty"△--data-binary△ @%elasticsearch_home%¥elasticsearch-definitions¥sm_mappings_filelibrary.json

5. 以下のレスポンスが返ってきており、登録が成功したことを確認します。
プロセス管理の場合

```
{
  "acknowledged" : true
}
```

図 1-9 データクローラ確認

5. インデックスの作成

1. curl コマンドを使用して、Index を作成します。

   プロセス管理の場合(“△” は半角スペースを示します。)

コマンド
curl△-XPUT△"http://localhost:9200/ssm?pretty"

   構成管理の場合(“△” は半角スペースを示します。)

コマンド
curl△-XPUT△"http://localhost:9200/ssm_ci?pretty"

   FAQ の場合(“△” は半角スペースを示します。)

コマンド
curl△-XPUT△"http://localhost:9200/ssm_faq?pretty"

   ファイルライブラリの場合(“△” は半角スペースを示します。)

コマンド
curl△-XPUT△"http://localhost:9200/ssm_filelibrary?pretty"

3. 以下のレスポンスが返ってきており、登録が成功したことを確認します。

   プロセス管理の場合

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : "ssm"
}
```

1.5.3 Elasticsearch へのデータ連携設定

Senju Service Manager から Elasticsearch へのデータ連携のためのアプリケーションとして Elasticsearch が提供する Logstash を導入します。

1. Logstash のインストール

logstash-8.11.3.zip を入手し稼働させたいディレクトリに展開します。

2. Logstash の設定

1. Senju Service Manager のインストールメディアに格納されている Logstash および logstash-definitions フォルダ一式をインストールするサーバーの任意のディレクトリに格納します。
   - フォルダ名: logstash
   - フォルダ名: logstash-definitions

   以降では、上記フォルダを下記ディレクトリに格納した場合を例示いたします。
   - ディレクトリパス: C:\temp\logstash-8.11.3

   設定ファイル logstash-xxx.confを以下の通り更新します。

   ・ データベースが Oracle である場合:

ファイル名	説明
logstash-oracle.conf	設定ファイル(プロセス管理)
logstash-oracle_ci.conf	設定ファイル(構成管理)
logstash-oracle-faq.conf	設定ファイル(FAQ)
logstash-oracle-filelibrary.conf	設定ファイル(ファイルライブラリ)

   ・データベースが PostgreSQL である場合:

ファイル名	説明
logstash-postgresql.conf	設定ファイル(プロセス管理)
logstash-postgresql_ci.conf	設定ファイル(構成管理)
logstash-postgresql-faq.conf	設定ファイル(FAQ)
logstash-postgresql-filelibrary.conf	設定ファイル(ファイルライブラリ)

   1) jdbc_connection_string に記載されている Oracle DB もしくは PostgreSQL DB への接続情報について、[hostname]:[portnumber]/[dbname] から、SSM DB サーバーの ホスト名:ポート番号/ローカル・ネット・サービス名に変更します。

      例としてホスト名:ccfsphost、ポート番号:5432、ローカル・ネット・サービス名:ssmdb に変更する場合を以下に示します。

```
# jdbc_connection_string =>"jdbc:postgresql://[hostname]: [portnumber]/[dbname]"
jdbc_connection_string => " jdbc:postgresql://ccfsphost:5432/ssmdb"
```

   2) jdbc_user に記載されている DB ユーザー名について、[username]から、正しいユーザー名に変更します。

      例として DB ユーザー名:ssmuser に変更する場合を以下に示します。

```
# jdbc_user => "[username]"
jdbc_user => "ssmuser"
```

3) jdbc_password に記載されている DB ユーザーのパスワードについて、[password]から、正しいパスワードに変更します。
例として DB ユーザーのパスワードを ssmpwd に変更する場合を示します。

```
#jdbc_password => "[password]"
jdbc_password => "ssmpwd"
```

4) hosts に記載されている Elasticsearch への接続情報について、[protocol]://[hostname]:[portnumber]から、Elasticsearch サーバーのプロトコル://ホスト名:ポート番号に変更します。
例としてプロトコル:http、ホスト名:eshost、ポート番号:9200 に変更する場合を示します。

```
#hosts => [ "[protocol]://[hostname]:[portnumber]"]
hosts => [ "http://eshost:9200"]
```

5) jdbc_driver_library に記載されている Elasticsearch への接続情報について、opt から、展開したディレクトリパスに更新します。
例として"C:\temp\logstash-8.11.3\logstash\lib\ojdbc7.jar"に変更する場合を示します。

```
#jdbc_driver_library    => "/opt/logstash/lib/ojdbc7.jar"
jdbc_driver_library    => "C:¥temp¥logstash-8.11.3¥logstash¥lib¥ojdbc7.jar"
```

6) last_run_metadata_path に記載されている Elasticsearch への接続情報について、opt から、展開したディレクトリパスに更新します。
例として"C:\temp\logstash-8.11.3\logstash\conf\.logstash_ oracle _process_fil_last_run"に変更する場合を示します。

```
#last_run_metadata_path=> "/opt/logstash/conf/.logstash_oracle_process_sub_fil_last_run"
last_run_metadata_path=> "C:¥temp¥logstash-8.11.3¥logstash¥conf¥.logstash_oracle_process_
fil_last_run"
```

7) statement_filepath に記載されている Elasticsearch への接続情報について、opt から、展開したディレクトリパスに更新します。
例として"C:\temp\logstash-8.11.3\logstash\lib\ojdbc7.jar"に変更する場合を示します。

```
#statement_filepath     => "/opt/logstash/sql/oracle/get_sm_data_from_process_sub_fil.sql"
statement_filepath=> "C:¥temp¥logstash-8.11.3¥logstash¥sql¥oracle¥get_sm_data_from
_process_fil.sql"
```

2. 修正後のファイルは以下のようになります。

・logstash-oracle.conf の場合

```
...（省略）
input {
```

```
  # for Upsert PROCESS_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:oracle:thin:@ccfsphost:1522/ssmdb"
    jdbc_driver_library    => "C:¥temp¥logstash-8.11.3¥logstash¥lib¥ojdbc7.jar"
    jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"

    schedule               => "* * * * *"
    last_run_metadata_path => "C:¥temp¥logstash-8.11.3¥logstash¥conf¥.logstash_ oracle
_process_fil_last_run"
    record_last_run        => "true"
    use_column_value       => "true"
    tracking_column        => "update_ts"
    statement_filepath     => "C:¥temp¥logstash-8.11.3¥logstash¥sql¥oracle
¥get_sm_data_from_process_fil.sql"

    type                   => "get_process_fil"
  }

  # for Upsert PROCESS_SUB_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:oracle:thin:@ccfsphost:1522/ssmdb"
    jdbc_driver_library    => "C:¥temp¥logstash-8.11.3¥logstash¥lib¥ojdbc7.jar"
    jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"
    schedule               => "* * * * *"
    last_run_metadata_path => "C:¥temp¥logstash-8.11.3¥logstash¥conf¥.logstash_ oracle
_process_sub_fil_last_run"
    record_last_run        => "true"
    use_column_value       => "true"
    tracking_column        => "update_ts"
    statement_filepath     => "C:¥temp¥logstash-8.11.3¥logstash¥sql¥oracle
¥get_sm_data_from_process_sub_fil.sql"
    type                   => "get_process_sub_fil"
  }

... (省略)

  # for Upsert PROCESS_TABLE_ITEM_5_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:oracle:thin:@ccfsphost:1522/ssmdb"
    jdbc_driver_library    => "C:¥temp¥logstash-8.11.3¥logstash¥lib¥ojdbc7.jar"
    jdbc_driver_class      => "Java::oracle.jdbc.driver.OracleDriver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"
    schedule               => "* * * * *"
    last_run_metadata_path => "C:¥temp¥logstash-8.11.3¥logstash¥conf¥.logstash_ oracle
_process_table_item_5_fil_last_run"
    record_last_run        => "true"
    use_column_value       => "true"
    tracking_column        => "update_ts"
    statement_filepath     => "C:¥temp¥logstash-8.11.3¥logstash¥sql¥oracle
```

```
¥get_sm_data_from_process_table_item_5_fil.sql"
    type                => "get_process_table_item_5_fil"
  }

... (省略)

output {
  if [type] == "get_process_fil" {
    elasticsearch{
      hosts             => [ "http://eshost:9200"]
      index             => "ssm"
      document_id       => "%{insert_no}"
      action            => "update"
      retry_on_conflict => 10
      doc_as_upsert     => true
    # user              => "[elasticusername]"
    # password          => "[elasticpassword]"
    }
  } else if [type] == "get_process_sub_fil" {
    elasticsearch{
      hosts             => [ "http://eshost:9200"]

... (省略)
```

・logstash-postgresql.conf の場合

```
... (省略)
input {
  # for Upsert PROCESS_FIL records
  jdbc {
    jdbc_connection_string => "jdbc:postgresql://ccfsphost:5432/ssmdb"
    jdbc_driver_library    => "C:¥temp¥logstash-8.11.3¥logstash¥lib¥postgresql-42.2.8.jar"
    jdbc_driver_class      => "org.postgresql.Driver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"

    schedule               => "* * * * *"
    last_run_metadata_path => "C:¥temp¥logstash-8.11.3¥logstash¥conf¥.logstash_ postgresql
_process_fil_last_run"
    record_last_run        => "true"
    use_column_value       => "true"
    tracking_column        => "update_ts"
    statement_filepath     => "C:¥temp¥logstash-8.11.3¥logstash¥sql¥postgresql
¥get_sm_data_from_process_fil.sql"

    type                   => "get_process_fil"
  }

  # for Upsert PROCESS_SUB_FIL records
  jdbc {
    jdbc_connection_string => " jdbc:postgresql://ccfsphost:5432/ssmdb"
    jdbc_driver_library    => "C:¥temp¥logstash-8.11.3¥logstash¥lib¥postgresql-42.2.8.jar"
    jdbc_driver_class      => "org.postgresql.Driver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"
```

```
    schedule              => "* * * * *"
    last_run_metadata_path => "C:¥temp¥logstash-8.11.3¥logstash¥conf¥.logstash_ postgresql
_process_sub_fil_last_run"
    record_last_run       => "true"
    use_column_value      => "true"
    tracking_column       => "update_ts"
    statement_filepath    => "C:¥temp¥logstash-8.11.3¥logstash¥sql¥postgresql
¥get_sm_data_from_process_sub_fil.sql"
    type                  => "get_process_sub_fil"
  }

...(省略)

  # for Upsert PROCESS_TABLE_ITEM_5_FIL records
  jdbc {
    jdbc_connection_string => " jdbc:postgresql://ccfsphost:5432/ssmdb"
    jdbc_driver_library    => "C:¥temp¥logstash-8.11.3¥logstash¥lib¥postgresql-42.2.8.jar"
    jdbc_driver_class      => "org.postgresql.Driver"
    jdbc_user              => "ssmuser"
    jdbc_password          => "ssmpwd"
    schedule               => "* * * * *"
    last_run_metadata_path => "C:¥temp¥logstash-8.11.3¥logstash¥conf¥.logstash_ postgresql
_process_table_item_5_fil_last_run"
    record_last_run       => "true"
    use_column_value      => "true"
    tracking_column       => "update_ts"
    statement_filepath    => "C:¥temp¥logstash-8.11.3¥logstash¥sql¥postgresql
¥get_sm_data_from_process_table_item_5_fil.sql"
    type                  => "get_process_table_item_5_fil"
  }

...(省略)

output {
  if [type] == "get_process_fil" {
    elasticsearch{
      hosts             => [ "http://eshost:9200"]
      index             => "ssm"
      document_id       => "%{insert_no}"
      action            => "update"
      retry_on_conflict => 10
      doc_as_upsert     => true
    # user              => "[elasticusername]"
    # password          => "[elasticpassword]"
    }
  } else if [type] == "get_process_sub_fil" {
    elasticsearch{
      hosts             => [ "http://eshost:9200"]

...(省略)
```

3. ディレクトリパス C:\temp\logstash-8.11.3\logstash を確認します。

   C:\temp\logstash-8.11.3\logstash パスに「conf」フォルダが存在しない場合、「conf」フォ

ルダを作成してください。

4. 設定ファイル pipelines.yml を以下の通り更新します。

ファイルパス: C:\temp\logstash-8.11.3\config \pipelines.yml

(“△” は半角スペースを示します。)

- 1 pipeline の設定を行います。

項目	設定値
pipeline.id	(インデックス名)
pipeline.workers	1
pipeline.batch.size	1
path.config	(設定ファイルのパス)
queue.type	persisted

```
- pipeline.id: ssm
  pipeline.workers: 1
  pipeline.batch.size: 1
  path.config: "/temp/logstash-8.11.3/logstash-definitions/logstash-oracle.conf"
  queue.type: persisted
- pipeline.id: ssm_faq
  pipeline.workers: 1
  pipeline.batch.size: 1
  queue.type: persisted
  path.config: "/temp/logstash-8.11.3/logstash-definitions/logstash-oracle-faq.conf"
- pipeline.id: ssm_filelibrary
  pipeline.workers: 1
  pipeline.batch.size: 1
  queue.type: persisted
  path.config: "/temp/logstash-8.11.3/logstash-definitions/logstash-oracle-
filelibrary.conf"
- pipeline.id: ssm_ci
  pipeline.workers: 1
  pipeline.batch.size: 1
  queue.type: persisted
  path.config: "/temp/logstash-8.11.3/logstash-definitions/logstash-oracle_ci.conf"
```

5. Logstash のログ設定ファイル log4j2.properties を編集します。

ファイルパス:C:\temp\logstash-8.11.3\config\log4j2.properties

以下の設定をファイルの appender.rolling.layout.pattern の後に追加します

```
appender.rolling.strategy.type = DefaultRolloverStrategy
appender.rolling.strategy.action.type = Delete
appender.rolling.strategy.action.basepath = ${sys:ls.logs}
appender.rolling.strategy.action.condition.type = IfLastModified
appender.rolling.strategy.action.condition.age = 7D
appender.rolling.strategy.action.PathConditions.type = IfFileName
appender.rolling.strategy.action.PathConditions.glob = logstash-${sys:ls.log.format}-*
```

修正後のファイルが以下の通りになっていることを確認します

```
...（省略）
appender.rolling.layout.pattern = [%d{ISO8601}][%-5p][%-25c] %-.10000m%n
appender.rolling.strategy.type = DefaultRolloverStrategy
appender.rolling.strategy.action.type = Delete
appender.rolling.strategy.action.basepath = ${sys:ls.logs}
appender.rolling.strategy.action.condition.type = IfLastModified
appender.rolling.strategy.action.condition.age = 7D
appender.rolling.strategy.action.PathConditions.type = IfFileName
appender.rolling.strategy.action.PathConditions.glob = logstash-${sys:ls.log.format}-*
appender.rolling.policies.size.type = SizeBasedTriggeringPolicy
...（省略）
```

3. インストール確認

1. コマンドプロンプトを開いて、以下のコマンドを実行し、Logstash を起動します。
（“△” は半角スペースを示します。）

コマンド
cd△C:¥temp¥logstash-8.11.3¥bin
logstash.bat

図 1-10 Logstash インストール確認

4. Elasticsearch サービスの生成

1. コマンドプロンプトを開いて、以下のコマンドを実行し、サービスを生成します。
（“△” は半角スペースを示します。）

コマンド
cd△%elasticsearch_home%¥bin
elasticsearch-service.bat△install

図 1-11 サービスの生成

2. 生成された Elasticsearch サービスを自動起動する設定に変更する手順を説明します。

   1) 「スタート」メニュー→「管理ツール」→「サービス」を選択します。

   2) 「1.」に生成された Elasticsearch サービスを右クリックし、コンテキストメニューからプロパティを選択します。

   3) プロパティの[全般]タブの[スタートアップの種類]を「自動」に変更し、[OK]ボタンをクリックしてください。

   ※Elasticsearch サービス名はデフォルトでは elasticsearch-service-x64 となります。

5. Logstash サービスの生成

1. nssm-2.24.zip を入手し稼働させたいディレクトリに展開します。
   ※以下、展開したディレクトリを%nssm_home%と記載して説明します。

2. nssm を利用して Logstash サービスを生成します。

コマンド
cd△%nssm_home%¥win64
nssm△install△サービス名

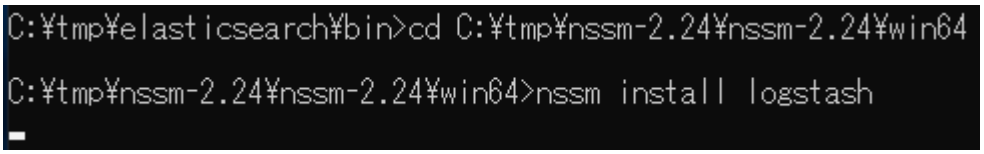

図 1-12 Logstash サービスの生成

3. 「NSSM service installer」設定ダイアログを開き、下記の通り設定を行います。

「Application」タグの設定：

Path：C:¥temp¥logstash-8.11.3¥bin¥logstash.bat
StartUp directory：C:¥temp¥logstash-8.11.3¥bin
Arguments：
Service name：手順 2 で入力した「サービス名」であることを確認

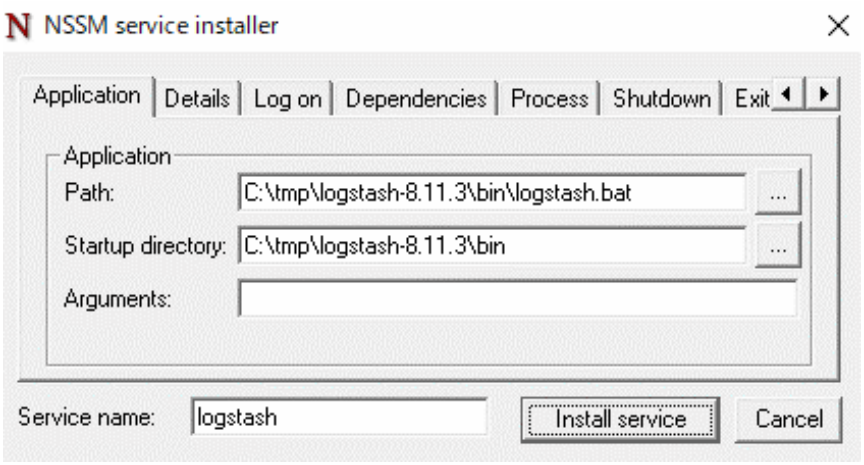

図 1-13 「Application」タグの設定

「Details」タグの設定：

Display name：手順 2 で入力した入力した「サービス名」であることを確認

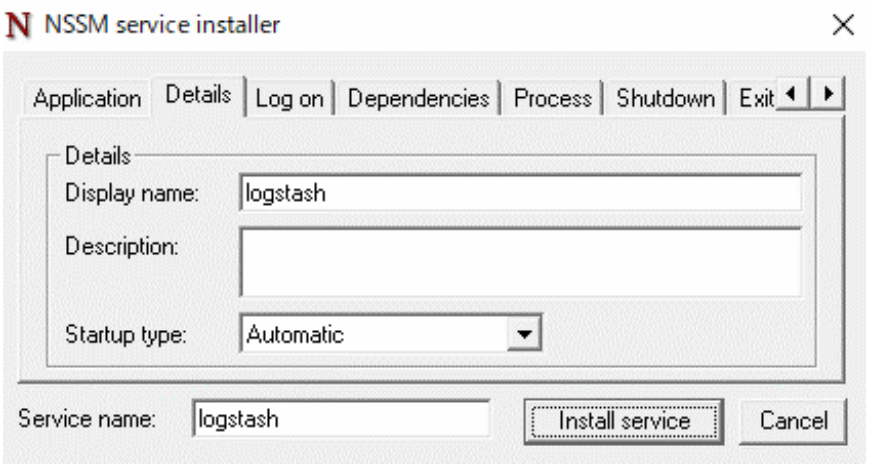

図 1-14 「Details」タグの設定

4. 「Install service」ボタンをクリックするとサービスの生成が完了します。

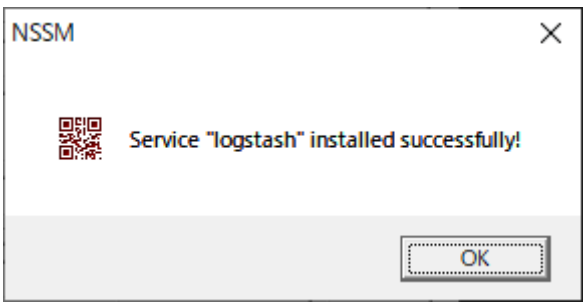

図 1-15 生成完了

6. サービス確認

Elasticsearch と Logstash が正しくサービスとして登録されていることを確認します。

1. Elasticsearch サービスを起動します。

（“△” は半角スペースを示します。）

コマンド
cd△%nssm_home%¥win64 nssm△start△Elasticsearch サービス名


仕様補足

Elasticsearch サービス名はデフォルトでは elasticsearch-service-x64 となります。

2. Logstash サービスを起動します。

（“△” は半角スペースを示します。）

コマンド
cd△%nssm_home%¥win64 nssm△start△Logstash サービス名

1.5.4 Elasticsearch の max_analyzed_offset 設定

Senju Service Manager と連携する Elasticsearch サーバーの、max_analyzed_offset を設定します。

1. curl コマンドを実行して max_analyzed_offset を設定します。

（“△” は半角スペースを示します。）

コマンド
curl△-XPUT△"http://localhost:9200/_settings?pretty"△-H△"Content-Type:application/json" △-d "{¥"index¥":{¥"highlight.max_analyzed_offset¥":51000000}}"

以下のレスポンスが返ってきており、設定が成功したことを確認します。

```
{
    "acknowledged" : true
}
```

2. Elasticsearch と Logstash を再起動します。

（“△” は半角スペースを示します。）

コマンド
nssm△stop△Elasticsearch サービス名 nssm△start△Elasticsearch サービス名 cd△%nssm_home%¥win64 nssm△restart△Logstash サービス名

1.5.5 ウィルススキャンの除外設定

Elasticsearch が稼働している環境において、アンチウィルスソフトのようなセキュリティ関連ソフトや、バックアップソフト等の予期しない動作により、パフォーマンスの影響や動作不調を起こす場合があります。
そのため、アプリケーションやミドルウェアのフォルダやファイルをリアルタイム検索から除外していただく必要があります。

各アンチウィルスソフトの除外設定手順に従い、設定を行ってください。

対象は 資料集「1.1.1 Senju Service Manager システム」の 9)を参照してください。

1.6 Elasticsearch 8.11.3へバージョンアップ

本手順では Elasticsearch を旧バージョンから Elasticsearch 8.11.3 へバージョンアップします。
まず、本章の手順で Elasticsearch と Logstash の削除を行います。
その後 Elasticsearch 8.11.3 をインストールする場合は、**「1.4 Linux 版 Elasticsearch の導入」**または**「1.5 Windows 版 Elasticsearch の導入」**を参照してください。

1.6.1 Linux 版バージョンアップ

1.6.1.1 Elasticsearch のアンインストール

1. 管理者権限で Elasticsearch サーバーにログインします。
2. Elasticsearch のサービスを停止します。

コマンド
systemctl△stop△elasticsearch

3. Elasticsearch のサービス登録を解除します。

コマンド
systemctl△disable△elasticsearch

4. Elasticsearch をアンインストールします。

コマンド
rpm△-e△$(rpm△-qa△|grep△'^elasticsearch-[0-9][0-9]*.*')

5. Elasticsearch の各種ファイルを削除します。

コマンド
rm△-rf△/etc/elasticsearch
rm△-rf△/var/lib/elasticsearch
rm△-rf△/var/log/elasticsearch

1.6.1.2 Logstash のアンインストール

1. Logstash のサービスを停止します。

コマンド
systemctl△stop△logstash

2. Logstash のサービス登録を解除します。

コマンド
systemctl△disable△logstash

3. Logstash のサービス登録を解除します。

コマンド
rpm△-e△$(rpm△-qa△|grep△'^logstash-[0-9][0-9]*.*')

4. Logstash の各種ファイルを削除します。

コマンド
rm△-rf△/opt/logstash
rm△-rf△/etc/logstash
rm△-rf△/var/lib/logstash
rm△-rf△/var/log/logstash

1.6.2 Windows 版バージョンアップ

1.6.2.1 Elasticsearch のアンインストール

1. 管理者権限で Elasticsearch サーバーにログインします。
2. Elasticsearch のサービスを停止します。

コマンド
cd△%nssm_home%¥win64 nssm△stop△Elasticsearch サービス名

3. Elasticsearch のサービスを削除します。

コマンド
cd△%elasticsearch_home%¥bin elasticsearch-service.bat△remove

4. Elasticsearch の各種ファイルを削除します。

   Elasticsearch のディレクトリを削除します。

1.6.2.2 Logstash のアンインストール

1. 管理者権限で Logstash サーバーにログインします。
2. Logstash のサービスを停止します。

コマンド
cd△%nssm_home%¥win64 nssm△stop△Logstash サービス名

3. Logstash のサービスを削除します。

コマンド
cd△%nssm_home%¥win64 nssm△remove△Logstash サービス名

4. Logstash の各種ファイルを削除します。

   Logstash のディレクトリを削除します。

1.7 Elasticsearch連携機能のアップデート

本手順では、Elasticsearch 連携機能のモジュールをアップデートする手順について説明します。

本手順の前に「1.4 Linux 版 Elasticsearch の導入」「1.5 Windows 版 Elasticsearch の導入」を実施している場合は、Elasticsearch 連携機能のモジュールが最新版となっているため、本手順は実施不要です。

アップデート時のパッチにより、手順内に記載されているディレクトリが存在しない場合があります。ディレクトリが存在しない場合は、手順をスキップしてください。

1.7.1 Linux 版アップデート

1. Logstash の確認

1. 以下のコマンドを実行し、サービスの状態を確認します。

（“△” は半角スペースを示します。）

コマンド
systemctl△status△logstash

2. 出力内容中に以下の内容が表示されることを確認します。

Active: inactive (dead)

3. 停止されていない場合は、以下のコマンドを実行し、サービスを停止します。

（“△” は半角スペースを示します。）

コマンド
systemctl△stop△logstash

2. モジュール適用

1. Patch ディレクトリに含まれる logstash ディレクトリ、logstash-definitions ディレクトリ、elasticsearch-definitions ディレクトリをサーバーの任意のディレクトリに格納します。
2. インストールするサーバーに管理者権限のアカウントでログインします。
3. データクローラの設定行います。

   本手順は、手順「1.4.1.2-5 データクローラの設定」を実施してください。

4. logstash ディレクトリをバックアップします。

バージョンアップ失敗時の復旧に使用するバックアップを行います。

/tmp/ディレクトリ配下に logstash_yyyymmdd(実行日付)のディレクトリのバックアップを取得します。

(“△” は半角スペースを示します。)

コマンド
cp△-fr△/opt/logstash/sql△`date△+/tmp/logstash_%Y%m%d`

以降の手順で問題が発生した場合は、以下のコマンドを実行後「8.既にモジュールが配置済みの場合以下のメッセージが表示されるので"y"を指定し上書きします。」以降の手順を実施してください。なお、モジュールをリストアした場合も Elasticsearch の情報を再取得する必要があります。リストア後「5 取り込み履歴の削除」から「10 全文検索機能の確認」までの手順についても実施してください。

例）/tmp/logstash_20210305/ディレクトリへバックアップした場合

(“△” は半角スペースを示します。)

コマンド
cp△-fr△/tmp/logstash_20210305/△/opt/logstash/sql

5. /etc/logstash/conf.d ディレクトリをバックアップします。

バージョンアップ失敗時の復旧に使用するバックアップを行います。

/tmp/ディレクトリ配下に logstash-definitions_yyyymmdd(実行日付)のディレクトリのバックアップを取得します。

(“△” は半角スペースを示します。)

コマンド
cp△-fr△/etc/logstash/conf.dl△`date△+/tmp/logstash-definitions_%Y%m%d`

以降の手順で問題が発生した場合は、以下のコマンドを実行後「8.既にモジュールが配置済みの場合以下のメッセージが表示されるので"y"を指定し上書きします。」以降の手順を実施してください。なお、モジュールをリストアした場合も Elasticsearch の情報を再取得する必要があります。リストア後「5 取り込み履歴の削除」から「10 全文検索機能の確認」までの手順についても実施してください。

例）/tmp/logstash_20210305/ディレクトリへバックアップした場合

(“△” は半角スペースを示します。)

コマンド
cp△-fr△/tmp/logstash-definitions_20210305/△/etc/logstash/conf.d

6. Kibana で作成したオブジェクトをバックアップします。

Kibana 連携を利用している場合は以下の手順を実施してください

Kibana 連携ガイドの「1.7.2 オブジェクトのインポート・エクスポート」エクスポート

手順を実施してください。 以降の手順で問題が発生した場合は、Kibana 連携ガイドの「1.7.2 オブジェクトのインポート・エクスポート」インポート手順を実施してください。

7. logstash/sql ディレクトリをコピーして/opt/logstash/sql 配下に格納します。

例）/tmp/ディレクトリへ配置した場合は、以下のコマンドを実行します。

（“△” は半角スペースを示します。）

コマンド
cp△-fr△/tmp/logstash/sql/△/opt/logstash/sql/

8. 既にモジュールが配置済みの場合以下のメッセージが表示されるので"y"を指定し上書きします。

例) モジュールが get_sm_data_from_process_fil.sql の場合

・.cp: `/opt/logstash/sql/oracle/get_sm_data_from_process_fil.sql' を上書きしますか?

・.cp: `/opt/logstash/sql/postgresql/get_sm_data_from_process_fil.sql' を上書きしますか?

9. 所有者およびグループを logstash ユーザーに変更します。

（“△” は半角スペースを示します。）

コマンド
chown△-R△logstash:logstash△/opt/logstash

3. インデックスの削除

本手順は、手順「1.11.1.1 Linux 版 Elasticsearch の場合」を実施してください

4. インデックスの作成

本手順は、手順「1.4.1.2-6 インデックスの作成」を実施してください

5. 取り込み履歴の削除

本手順は、手順「1.11.1.1 Linux 版 Elasticsearch の場合」を実施してください

6. Logstash の設定

本手順は、手順「1.4.1.3-2 Logstash の設定」を実施してください

7. サービスの起動

サービスを起動します。
(“△” は半角スペースを示します。)

コマンド
systemctl△start△logstash

以下のコマンドを実行し、サービスの状態を確認します。

(“△” は半角スペースを示します。)

コマンド
systemctl△status△logstash

出力内容中に以下の内容が表示されることを確認します。

Active: active (running)

8. 取り込み履歴の確認

1. Logstash の取り込み履歴ファイル格納ディレクトリに移動します。

(“△” は半角スペースを示します。)

コマンド
cd△/opt/logstash/conf

※上記パスはデフォルト設定先になります。履歴ファイルの格納場所については、以下のファイル内容に記載されている"last_run_metadata_path =>"の設定値をご確認ください。

・データベースが Oracle である場合：

<ディレクトリ>　/etc/logstash/conf.d/

ファイル名	説明
logstash-oracle.conf	設定ファイル(プロセス管理)
logstash-oracle_ci.conf	設定ファイル(構成管理)
logstash-oracle-faq.conf	設定ファイル(FAQ)
logstash-oracle-filelibrary.conf	設定ファイル(ファイルライブラリ)

・データベースが PostgreSQL である場合：

<ディレクトリ>　/etc/logstash/conf.d/

ファイル名	説明
logstash-postgresql.conf	設定ファイル(プロセス管理)
logstash-postgresql_ci.conf	設定ファイル(構成管理)
logstash-postgresql-faq.conf	設定ファイル(FAQ)
logstash-postgresql-filelibrary.conf	設定ファイル(ファイルライブラリ)

2. 履歴ファイルを確認します。

コマンド
ls△-a

以下の内容が出力されることを確認します。

```
<Oracle をご利用の場合>
.logstash_oracle_ci_fil_last_run
.logstash_oracle_faq_fil_last_run
.logstash_oracle_file_library_fil_last_run
.logstash_oracle_process_fil_last_run
.logstash_oracle_process_sub_fil_last_run
.logstash_oracle_process_table_item_1_fil_last_run
.logstash_oracle_process_table_item_2_fil_last_run
.logstash_oracle_process_table_item_3_fil_last_run
.logstash_oracle_process_table_item_4_fil_last_run
.logstash_oracle_process_table_item_5_fil_last_run
```

```
<PostgreSQL をご利用の場合>
.logstash_postgresql_ci_fil_last_run
.logstash_postgresql_faq_fil_last_run
.logstash_postgresql_file_library_fil_last_run
.logstash_postgresql_process_fil_last_run
.logstash_postgresql_process_sub_fil_last_run
.logstash_postgresql_process_table_item_1_fil_last_run
.logstash_postgresql_process_table_item_2_fil_last_run
.logstash_postgresql_process_table_item_3_fil_last_run
.logstash_postgresql_process_table_item_4_fil_last_run
.logstash_postgresql_process_table_item_5_fil_last_run
```

3. Logstash のキューファイル格納ディレクトリに移動します。

(“△” は半角スペースを示します。)

コマンド
cd△/var/lib/logstash/

4. キューファイルを確認します。

コマンド
ls

以下の内容が出力されることを確認します。

```
queue  uuid
```

5. インデックス情報を確認します。

```
curl△-XGET△[protocol]://[hostname]:[portnumber]/_aliases?pretty
```

例）Elasticsearch への接続情報でプロトコルを[protocol] → http、ホスト名を[hostname] → eshost、ポート番号を[portnumber] → 9200 とした場合は、以下のコマンドを実行します。

(“△” は半角スペースを示します。)

コマンド
curl△-XGET△http://eshost:9200/_aliases?pretty

以下の結果が出力されることを確認します

```
{
  "ssm" : {
    "aliases" : { }
  },
  "ssm_ci" : {
    "aliases" : { }
  },
  "ssm_faq" : {
    "aliases" : { }
  },
  "ssm_filelibrary" : {
    "aliases" : { }
  }
}
```

9. ログ情報の確認

1. 以下のログファイルの内容を確認します。

   <確認ファイルパス>

   /var/log/logstash/logstash-plain.log

2. Logstash サービス起動後の時間帯にエラー出力がないことを確認します。

10. 全文検索機能の確認

1. Senju/SM にログインし、全文検索機能の動作を確認します。
2. 検索結果として、値が取得可能なことを確認します。

11. Kibana 機能の確認

Kibana 連携を利用している場合は本手順を実施してください。

1. Kibana 機能の動作確認をします。
2. 分析レポートが出力されることを確認します。

1.7.2 Windows 版アップデート

1. Logstash の確認

1. 以下のコマンドを実行し、サービスの状態を確認します。

（“△” は半角スペースを示します。）

コマンド
cd△%nssm_home%¥win64 nssm△status△Logstash サービス名

2. 出力内容中に以下の内容が表示されることを確認します。

SERVICE_STOPPED

3. 停止されていない場合は、以下のコマンドを実行し、サービスを停止します。

（“△” は半角スペースを示します。）

コマンド
cd△%nssm_home%¥win64 nssm△stop△Logstash サービス名

2. モジュール適用

1. Patch ディレクトリに含まれる logstash ディレクトリ、logstash-definitions ディレクトリ、elasticsearch-definitions ディレクトリをサーバーの任意のディレクトリに格納します。
2. インストールするサーバーに管理者権限のアカウントでログインします。
3. データクローラの設定を行います。
   本手順は、手順「1.5.2-4 データクローラの設定」を実施してください。

4. logstash ディレクトリをバックアップします。
   バージョンアップ失敗時の復旧に使用するバックアップを行います。
   %logstash_home%\logstash\sql のディレクトリを任意のフォルダにコピーします
   ※%logstash_home%は\logstash-8.11.3 を格納した先を指します。

   以降の手順で問題が発生した場合は、モジュールをリストアしてください。
   なお、モジュールをリストアした場合も Elasticsearch の情報を再取得する必要があります。 リストア後「5 取り込み履歴の削除」から「10 全文検索機能の確認」までの手順についても実施してください。

5. logstash-definitions ディレクトリをバックアップします。
   バージョンアップ失敗時の復旧に使用するバックアップを行います。
   %logstash_home%\ logstash-definitions ディレクトリを任意のフォルダにコピーします
   ※%logstash_home%は\logstash-8.11.3 を格納した先を指します。

   以降の手順で問題が発生した場合は、モジュールをリストアしてください。
   なお、モジュールをリストアした場合も Elasticsearch の情報を再取得する必要があります。 リストア後「5 取り込み履歴の削除」から「10 全文検索機能の確認」までの手順についても実施してください。

6. Kibana で作成したオブジェクトをバックアップします。

Kibana 連携を利用している場合は以下の手順を実施してください
Kibana 連携ガイドの「1.7.2 オブジェクトのインポート・エクスポート」エクスポート手順を実施してください。 以降の手順で問題が発生した場合は、Kibana 連携ガイドの「1.7.2 オブジェクトのインポート・エクスポート」インポート手順を実施してください。

7. logstash/sql ディレクトリをコピーして%logstash_home%\logstash\sql 配下に格納します。
8. 既にモジュールが配置済みの場合以下のメッセージが表示されるので"ファイルを置き換える(R)"を指定し上書きします。
   例) モジュールが get_sm_data_from_process_fil.sql の場合
   宛先には既に"get_sm_data_from_process_fil.sql"という名前のファイルが存在しますと聞かれるので"ファイルを置き換える(R)"を指定します。

3. インデックスの削除

本手順は、手順「1.11.1.2 Windows 版 Elasticsearch の場合」を実施してください

4. インデックスの作成

本手順は、手順「1.5.2-5 インデックスの作成」を実施してください

5. 取り込み履歴の削除

本手順は、手順「1.11.1.2 Windows 版 Elasticsearch の場合」を実施してください

6. Logstash の設定

本手順は、手順「1.5.3-2 Logstash の設定」を実施してください

7. サービスの起動

サービスを起動します。
(“△” は半角スペースを示します。)

コマンド
cd△%nssm_home%¥win64 nssm△start△Logstash サービス名

以下のコマンドを実行し、サービスの状態を確認します。
(“△” は半角スペースを示します。)

コマンド
cd△%nssm_home%¥win64

コマンド
nssm△status△Logstash サービス名

出力内容中に以下の内容が表示されることを確認します。

SERVICE_RUNNING

8. 取り込み履歴の確認

1. Logstash の取り込み履歴ファイル格納ディレクトリに移動します。

履歴ファイルの格納場所については、以下のファイル内容に記載されている

"last_run_metadata_path =>"の設定値をご確認ください。

・データベースが Oracle である場合：

<ディレクトリ>　C:\temp\logstash-8.11.2\logstash-definitions

ファイル名	説明
logstash-oracle.conf	設定ファイル（プロセス管理）
logstash-oracle_ci.conf	設定ファイル（構成管理）
logstash-oracle-faq.conf	設定ファイル（FAQ）
logstash-oracle-filelibrary.conf	設定ファイル（ファイルライブラリ）

・データベースが PostgreSQL である場合：

<ディレクトリ>　C:\temp\logstash-8.11.2\logstash-definitions

ファイル名	説明
logstash-postgresql.conf	設定ファイル（プロセス管理）
logstash-postgresql_ci.conf	設定ファイル（構成管理）
logstash-postgresql-faq.conf	設定ファイル（FAQ）
logstash-postgresql-filelibrary.conf	設定ファイル（ファイルライブラリ）

2. 履歴ファイルを確認します。

以下の内容が出力されることを確認します。

<Oracle をご利用の場合> .logstash_oracle_ci_fil_last_run .logstash_oracle_faq_fil_last_run .logstash_oracle_file_library_fil_last_run .logstash_oracle_process_fil_last_run .logstash_oracle_process_sub_fil_last_run .logstash_oracle_process_table_item_1_fil_last_run .logstash_oracle_process_table_item_2_fil_last_run .logstash_oracle_process_table_item_3_fil_last_run .logstash_oracle_process_table_item_4_fil_last_run .logstash_oracle_process_table_item_5_fil_last_run

<PostgreSQL をご利用の場合> .logstash_postgresql_ci_fil_last_run .logstash_postgresql_faq_fil_last_run

```
.logstash_postgresql_file_library_fil_last_run
.logstash_postgresql_process_fil_last_run
.logstash_postgresql_process_sub_fil_last_run
.logstash_postgresql_process_table_item_1_fil_last_run
.logstash_postgresql_process_table_item_2_fil_last_run
.logstash_postgresql_process_table_item_3_fil_last_run
.logstash_postgresql_process_table_item_4_fil_last_run
.logstash_postgresql_process_table_item_5_fil_last_run
```

3. Logstash のキューファイル格納ディレクトリに移動します。

(“△” は半角スペースを示します。)

コマンド
cd△C:¥temp¥logstash-8.11.3¥data

4. キューファイルを確認します。

コマンド
dir

以下の内容が出力されることを確認します。

(例)

```
2021/02/09 20:13 <DIR> dead_letter_queue
2021/02/09 20:13 36 uuid
```

5. インデックス情報を確認します。

```
curl△-XGET△[protocol]://[hostname]:[portnumber]/_aliases?pretty
```

例）Elasticsearch への接続情報でプロトコルを[protocol] → http、ホスト名を[hostname] → eshost、 ポート番号を[portnumber] → 9200 とした場合は、以下のコマンドを実行します。

(“△” は半角スペースを示します。)

コマンド
curl△-XGET△http://eshost:9200/_aliases?pretty

以下の結果が出力されることを確認します

```
{
  "ssm" : {
    "aliases" : { }
  },
  "ssm_ci" : {
    "aliases" : { }
  },
  "ssm_faq" : {
    "aliases" : { }
  },
  "ssm_filelibrary" : {
    "aliases" : { }
  }
}
```

9. ログ情報の確認

1. 以下のログファイルの内容を確認します。

   <確認ファイルパス>

   %logstash_home%\logs\logstash-plain.log

2. Logstash サービス起動後の時間帯にエラー出力がないことを確認します。

10. 全文検索機能の確認

1. Senju/SM にログインし、全文検索機能の動作を確認します。
2. 検索結果として、値が取得可能なことを確認します。

11. Kibana 機能の確認

Kibana 連携を利用している場合は本手順を実施してください。

- Kibana 機能の動作確認をします。
- 分析レポートが出力されることを確認します。

1.8 Elasticsearch認証の設定

Elastic Stack の標準の機能で Elasticsarch に接続する際にユーザーの認証を設定することができます。そちらの手順について説明します。

注意 Kibana をご利用する方は必ず本手順を実施してください。

必須設定 ユーザー認証を有効にするためには、Elastic Stack のセキュリティ設定を有効にする必要があります。

必須設定 Elastic Stack のセキュリティ設定を有効にするためには、ノード間通信を TLS で暗号化する必要があります。

ノード間通信を TLS で暗号化するため、Elasticsearch に同梱されているツールを用いて以下を行います。

- 認証局の生成
- ノード証明書（自己証明書）の発行

1.8.1 認証局の生成

仕様補足 Linux 版 Elasticsearch はインストール時にセキュリティ設定が行われ、認証局が生成されるため、本手順を実施する必要はありません。

1) ノード証明書の信頼性を検証するための認証局(CA)を生成します。
（“△” は半角スペースを示します。）

コマンド
> cd△%elasticsearch_home% > bin¥elasticsearch-certutil△ca Please enter the desired output file [elastic-stack-ca.p12]: <Enter> Enter password for elastic-stack-ca.p12 : <認証局のパスワード(任意)>

2) 認証局が生成されたことを確認します。
（“△” は半角スペースを示します。）

コマンド
> dir△%elasticsearch_home%¥elastic-stack-ca.p12 elastic-stack-ca.p12

仕様補足 認証局のパスワードはこの後の手順で使用するため、
メモをする等して忘れないでください。

1.8.2 ノード証明書の発行

仕様補足

Linux 版 Elasticsearch はインストール時にセキュリティ設定が行われ、ノード証明書が発行されるため、本手順を実施する必要はありません。

1) ノード証明書を発行します。

(“△” は半角スペースを示します。)

コマンド
> cd△%elasticsearch_home% > bin¥elasticsearch-certutil△cert△-ca△elastic-stack-ca.p12△-ca-pass△<認証局のパスワード> Please enter the desired output file [elastic-certificates.p12]: <Enter> Enter password for elastic-certificates.p12 : <ノード証明書のパスワード(任意)>

2) ノード証明書が発行されたことを確認します。

コマンド
> dir△%elasticsearch_home%¥elastic-certificates.p12 elastic-certificates.p12

3) ノード証明書のパスワードを Elasticsearch のキーストアに登録します。

キー名	説明
xpack.security.transport.ssl.keystore.secure_password	トランスポート層：キーストアのパスワード
xpack.security.transport.ssl.truststore.secure_password	トランスポート層：トラストストアのパスワード
xpack.security.http.ssl.keystore.secure_password	HTTP 層：キーストアのパスワード

コマンド
> cd△%elasticsearch_home%bin > elasticsearch-keystore add xpack.security.transport.ssl.keystore.secure_password Enter value for xpack.security.transport.ssl.keystore.secure_password: <ノード証明書のパスワード> > elasticsearch-keystore add xpack.security.transport.ssl.truststore.secure_password Enter value for xpack.security.transport.ssl.truststore.secure_password: <ノード証明書のパスワード> > elasticsearch-keystore add xpack.security.http.ssl.keystore.secure_password Enter value for xpack.security.http.ssl.keystore.secure_password: <ノード証明書のパスワード>

仕様補足

ノード証明書のパスワードはこの後の手順で使用するため、
メモをする等して忘れないでください。

1.8.3 ノード証明書の配布

仕様補足

Linux 版 Elasticsearch はインストール時にセキュリティ設定が行われ、ノード証明書が配布されるため、本手順を実施する必要はありません。

1) 配布先のフォルダを作成します。

コマンド

```
> cd△%elasticsearch_home%
> mkdir△config¥certs
```

2) ノード証明書をコピーします。

コマンド

```
> cd△%elasticsearch_home%
> copy△elastic-certificates.p12△config¥certs
```

3) ノード証明書がコピーされたことを確認します。

コマンド

```
> dir△%elasticsearch_home%¥config¥certs¥elastic-certificates.p12
elastic-certificates.p12
```

1.8.4 Elasticsearch 設定ファイルの編集

仕様補足

Linux 版 Elasticsearch はインストール時にセキュリティ設定が行われ、Elasticsearch 設定ファイル末尾にセキュリティ設定が記載された状態となります。
無効にしたセキュリティ設定を元に戻します。

1. Elasticsearch サーバーのサービスを一時停止します。
2. Elasticsearch サーバーの設定ファイル elasticsearch.yml を更新します。

・Elasticsearch サーバーの OS が Linux である場合:

ファイルパス:/etc/elasticsearch/elasticsearch.yml

- 1 Elastic Stack のセキュリティを有効に設定します。

```
xpack.security.enabled: true
xpack.security.enrollment.enabled: true
```

- 2 Elasticsearch ノード間の暗号化・相互認証を有効に設定します。

```
xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/transport.p12
  truststore.path: certs/transport.p12
```

・Elasticsearch サーバーの OS が Windows である場合:

ファイルパス:%elasticsearch_home%\config\elasticsearch.yml

```
#xpack.security.enabled: false
```

```
xpack.security.enabled: true
xpack.security.enrollment.enabled: true

xpack.security.transport.ssl:
  enabled: true
  verification_mode: certificate
  keystore.path: certs/elastic-certificates.p12
  truststore.path: certs/elastic-certificates.p12

xpack.security.http.ssl:
   enabled: false
```

Elasticsearch サーバーのサービスを再起動します。

1.8.5 デフォルトユーザーのパスワード設定

仕様補足

予め用意されているユーザーは以下の通りです。
・elastic
・kibana_system
・kibana
・logstash_system
・beats_system
・apm_system
・remote_monitoring_system

仕様補足

Linux 版 Elasticsearch はインストール時にセキュリティ設定が行われ、デフォルトユーザーのパスワードは設定済みとなります。
以下の手順でパスワードをリセットしてください。

仕様補足

Windows 版 Elasticsearch はインストール時にセキュリティ設定が行われず、デフォルトユーザーのパスワードは未設定となります。
以下の手順でパスワードを設定してください。

仕様補足

表示されたパスワードはこの後の手順で使用するため、
メモをする等して忘れないでください。

Elasticsearch サーバーでコマンドプロンプトを開き、以下のコマンドを実行します。

(“△” は半角スペースを示します。)

・Elasticsearch サーバーの OS が Linux である場合:

コマンド
/usr/share/elasticsearch/bin/elasticsearch-reset-password△-u△[ユーザー名]

・Elasticsearch サーバーの OS が Windows である場合:

コマンド
cd△%elasticsearch_home%¥bin elasticsearch-setup-passwords△auto

1.8.6 Elasticsearch の疎通確認

以下のコマンドを実行し、Elasticsearch の通信でユーザー認証が有効となることを確認します。

(“△” は半角スペースを示します。)

コマンド
curl△-XGET△″http://localhost:9200/?pretty″△-u△elastic:elapwd

「1.5.1-3 Elasticsearch のインストール確認」と同じレスポンスが返ることを確認します。

1.8.7 認証用ユーザーの作成

Elasticsearch の認証するユーザーとパスワードは Kibana を利用することで自由に作成することが可能です。

そちらの手順は別紙 Kibana 連携ガイド「ユーザーの作成」を参照してください。

1.8.8 logstash 設定ファイルの更新

ユーザー認証の設定した状態で logstash により収集したデータを
Elasticsearch に連携する際には logstash 側にも認証の設定を行う必要があります。

1. Logstash サービスを停止します。

   Elasticsearch サーバーの OS が Linux の場合:

   以下のコマンドを実行し、サービスを停止します。

   (“△” は半角スペースを示します。)

コマンド
systemctl△stop△logstash

   Elasticsearch サーバーの OS が Windows の場合:

   以下のコマンドを実行し、サービスを停止します。

   (“△” は半角スペースを示します。)

コマンド
cd△%nssm_home%¥win64 nssm△stop△Logstash サービス名

2. Logstash 設定ファイル を更新します。

Elasticsearch サーバーの OS が Linux の場合:

設定ファイルの格納先： /etc/logstash/conf.d 配下

Elasticsearch サーバーの OS が Windows の場合:

設定ファイルの格納先： C:\temp\logstash-8.11.3\logstash-definitions 配下

・ データベースが Oracle である場合:

ファイル名	説明
logstash-oracle.conf	設定ファイル（プロセス管理）
logstash-oracle_ci.conf	設定ファイル（構成管理）
logstash-oracle-faq.conf	設定ファイル（FAQ）
logstash-oracle-filelibrary.conf	設定ファイル（ファイルライブラリ）

・データベースが PostgreSQL である場合:

ファイル名	説明
logstash-postgresql.conf	設定ファイル（プロセス管理）
logstash-postgresql_ci.conf	設定ファイル（構成管理）
logstash-postgresql-faq.conf	設定ファイル（FAQ）
logstash-postgresql-filelibrary.conf	設定ファイル（ファイルライブラリ）

filter plugin の設定

・ Ruby filter plugin の設定 （構成管理、FAQ、ファイルライブラリ）

1. [elasticusername],[elasticpassword] を、Elasticsearch の認証用ユーザー名とパスワードに変更します。

（Elasticsearch のユーザー認証情報をユーザー名:elastic、パスワード:elapwd に変更する場合）

```
#basic_auth = "Basic " + Base64.strict_encode64("[elasticusername]:[elasticpassword]")
basic_auth = "Basic " + Base64.strict_encode64("elastic:elapwd")
```

2. 以下のコメントアウトを外します。

```
#event.set("[@metadata][basic_auth]", basic_auth)
event.set("[@metadata][basic_auth]", basic_auth)
```

output plugin の設定

・ Elasticsearch output plugin の設定

1. user,password に記載されている Elasticsearch のユーザー認証情報について、[elasticusername],[elasticpassword] から、Elasticsearch の認証用ユーザー名とパスワードに変更します。

（Elasticsearch のユーザー認証情報をユーザー名:elastic、パスワード:elapwd に変

更する場合）

```
#user => "[elasticusrename]"
user => "elastic"
#password => "[elasticpassword]"
password=> "elapwd"
```

・ Http output plugin の設定 （構成管理、FAQ、ファイルライブラリ）

1. 以下のコメントアウトを外します。

```
#"Authorization" => "%{[@metadata][basic_auth]}"
"Authorization" => "%{[@metadata][basic_auth]}"
```

修正後のファイルは以下のようになります。

```
...（省略）

filter {

...（省略）

  ruby {
    code => '
      require "base64"

    basic_auth = "Basic " + Base64.strict_encode64("elatic:elapwd")
    event.set("[@metadata][basic_auth]", basic_auth)

...（省略）

  }
}

...（省略）

output {

  if [type] == "get_ci_fil" {
    elasticsearch{
      hosts             => [ "http://eshost:9200" ]
      index             => "ssm_ci"
      document_id       => "%{ci_id}"
      action            => "update"
      retry_on_conflict => 10
      doc_as_upsert     => true
      pipeline          => "attachment"
      user              => "elastic"
      password          => "elapwd"
    #  ssl_enabled           => true
    #  ssl_verification_mode => none
    #  ssl_keystore_path     => "[keystore_path]"
    #  ssl_keystore_password => "[keystore_password]"
    }
```

```
  if [data1] != "" {
    http {
      url => "http://eshost:9200/ssm_ci/_doc/%{ci_id}?pipeline=attachment"
      http_method => "put"
      headers => {
        "Content-Type" => "application/json"
        "Authorization" => "%[@metadata][basic_auth]]"
      }
      format => "json"
      message => "%{data1}"
    #  ssl_verification_mode => none
    #  ssl_keystore_path     => "[keystore_path]"
    #  ssl_keystore_password => "[keystore_password]"
    }
  }

... (省略)
```

3. Logstash サービスを起動します。

Elasticsearch サーバーの OS が Linux の場合:

以下のコマンドを実行し、サービスを起動します。

(“△” は半角スペースを示します。)

コマンド
systemctl△start△logstash

Elasticsearch サーバーの OS が Windows の場合:

以下のコマンドを実行し、サービスを起動します。

(“△” は半角スペースを示します。)

コマンド
cd△%nssm_home%¥win64 nssm△start△Logstash サービス名

1.8.9 Senju Service Manager の設定

Senju Service Manager で提供する Elasticsearch 連携機能を使用する際にも

Elasticsearch 認証の設定が必要となります。

本手順で設定したユーザーとパスワードを使用してください。

必須設定

Elasticsearch 連携機能を利用する為には以下の設定が必須です。

・[管理者メニュー > 制御情報 > 制御情報 > 共通 > 共通]

管理項目名
Elasticsearch 認証時のアカウント
Elasticsearch 認証時のパスワード

1.9 ElasticsearchのHTTPS設定

Elastic Stack の標準の機能で Elasticsearch に接続する際の通信を TLS で暗号化することができます。そちらの手順について説明します。

注意 本章の設定を行った場合、Logstash や Kibana から Elasticsearch に接続する際の通信が TLS で暗号化されるため、Logstash や Kibana 側で HTTPS 設定が必要となります。

必須設定 本章の設定を行うためには、「1.8.1 認証局の生成」「1.8.2 ノード証明書の発行」を事前に行っておく必要があります。

1.9.1 Elasticsearch 設定ファイルの編集

1. Elasticsearch サーバーのサービスを一時停止します。
2. Elasticsearch サーバーの設定ファイル elasticsearch.yml の末尾に
   以下の記載を追加し、更新します。

・Elasticsearch サーバーの OS が Linux である場合:

ファイルパス:/etc/elasticsearch/elasticsearch.yml

- 1 Elasticsearch とクライアント間通信の TLS 暗号化を有効に設定します。

```
xpack.security.http.ssl:
  enabled: true
  keystore.path: certs/http.p12
```

・Elasticsearch サーバーの OS が Windows である場合:

ファイルパス:%elasticsearch_home%\config\elasticesearch.yml

```
xpack.security.http.ssl:
   #enabled: false
   enabled: true
   keystore.path: certs/elastic-certificates.p12
```

Elasticsearch サーバーのサービスを再起動します。

1.9.2 Elasticsearch の疎通確認

以下のコマンドを実行し、Elasticsearch の通信が TLS で暗号化されたことを確認します。
(“△” は半角スペースを示します。)

コマンド
curl△-XGET△"https://localhost:9200/?pretty"△-u△elastic:elapwd△--insecure

「1.5.1-3 Elasticsearch のインストール確認」と同じレスポンスが返ることを確認します。

注意

Elasticsearch に包含されるツールを利用してノード証明書を発行する場合、ノード証明書は自己証明書となります。
自己証明書で構築されたサイトに対し curl コマンドを実行するとエラーが発生するため、「--insecure」または「-k」オプションを指定して自己証明書を受け入れてください。

1.9.3 ノード証明書の配布

Elasticsearch サーバーの OS が Linux の場合:

以下のコマンドを実行し、Elasticsearch のノード証明書を Logstash 環境にコピーします。

(“△” は半角スペースを示します。)

コマンド
mkdir△/etc/logstash/certs cp△-p△/etc/elasticsearch/certs/http.p12△/etc/logstash/certs chown△logstash:logstash△/etc/logstash/certs/http.p12

以下のコマンドを実行し、ノード証明書がコピーされたことを確認します。

(“△” は半角スペースを示します。)

コマンド
> ls△/etc/logstash/certs/http.p12 http.p12

Elasticsearch サーバーの OS が Windows の場合:

以下のコマンドを実行し、Elasticsearch のノード証明書を Logstash 環境にコピーします。

(“△” は半角スペースを示します。)

コマンド
mkdir△C:¥temp¥logstash-8.11.3¥config¥certs copy△%elasticsearch_home%¥config¥certs¥elastic-certificates.p12△C:¥temp¥logstash-8.11.3¥config¥certs

以下のコマンドを実行し、ノード証明書がコピーされたことを確認します。

(“△” は半角スペースを示します。)

コマンド
> dir△C:¥temp¥logstash-8.11.3¥config¥certs¥elastic-certificates.p12 elastic-certificates.p12

1.9.4 logstash 設定ファイルの更新

Elasticsearch に接続する際の通信を TLS で暗号化した状態で logstash により収集したデータを Elasticsearch に連携する際には logstash 側にも TLS 暗号化の設定を行う必要があります。

1. Logstash サービスを停止します。

   Elasticsearch サーバーの OS が Linux の場合:

   以下のコマンドを実行し、サービスを停止します。

   (“△” は半角スペースを示します。)

コマンド
systemctl△stop△logstash

   Elasticsearch サーバーの OS が Windows の場合:

   以下のコマンドを実行し、サービスを停止します。

   (“△” は半角スペースを示します。)

コマンド
cd△%nssm_home%¥win64 nssm△stop△Logstash サービス名

2. Logstash 設定ファイルを更新します。

   Elasticsearch サーバーの OS が Linux の場合:

   設定ファイルの格納先： /etc/logstash/conf.d 配下

   Elasticsearch サーバーの OS が Windows の場合:

   設定ファイルの格納先： C:\temp\logstash-8.11.3\logstash-definitions 配下

   ・ データベースが Oracle である場合:

ファイル名	説明
logstash-oracle.conf	設定ファイル(プロセス管理)
logstash-oracle_ci.conf	設定ファイル(構成管理)
logstash-oracle-faq.conf	設定ファイル(FAQ)
logstash-oracle-filelibrary.conf	設定ファイル(ファイルライブラリ)

   ・データベースが PostgreSQL である場合:

ファイル名	説明
logstash-postgresql.conf	設定ファイル(プロセス管理)
logstash-postgresql_ci.conf	設定ファイル(構成管理)
logstash-postgresql-faq.conf	設定ファイル(FAQ)
logstash-postgresql-filelibrary.conf	設定ファイル(ファイルライブラリ)

   output plugin の設定

   - Elasticsearch output plugin の設定

     1. hosts に記載されている Elasticsearch への接続情報について、[protocol]を https に

変更します。

（Elasticsearch への接続情報が、ホスト名:eshost、ポート番号:9200 の場合）

```
#hosts => [ "http://localhost:9200"]
hosts => [ "https://localhost:9200"]
```

2. Logstash と Elasticsearch 間の通信を SSL で保護するよう変更します。

ssl_xxxx のコメントアウトを外します。

ssl_keystore_path にノード証明書のパスを設定します。

ssl_keystore_password にノード証明書のパスワード(※1)を設定します。

Elasticsearch サーバーの OS が Linux の場合:

```
#ssl_enabled          => true
#ssl_verification_mode => none
#ssl_keystore_path     => "[keystore_path]"
#ssl_keystore_password => "[keystore_password]"
ssl_enabled          => true
ssl_verification_mode => none
ssl_keystore_path     => "/etc/logstash/certs/http.p12"
ssl_keystore_password => "elanodepwd"
```

Elasticsearch サーバーの OS が Windows の場合:

```
#ssl_enabled          => true
#ssl_verification_mode => none
#ssl_keystore_path     => "[keystore_path]"
#ssl_keystore_password => "[keystore_password]"
ssl_enabled          => true
ssl_verification_mode => none
ssl_keystore_path     => " C:¥temp¥logstash-8.11.3¥config¥certs¥elastic-
certificates.p12"
ssl_keystore_password => "elanodepwd"
```

・ Http output plugin の設定 （構成管理、FAQ、ファイルライブラリ）

1. hosts に記載されている Elasticsearch への接続情報について、[protocol]を https に変更します。

（Elasticsearch への接続情報が、ホスト名:eshost、ポート番号:9200 の場合）

```
#url => "http://eshost:9200/ssm_ci/_doc/%{ci_id}?pipeline=attachment"
url => "https://eshost:9200/ssm_ci/_doc/%{ci_id}?pipeline=attachment"
```

2. Logstash と Elasticsearch 間の通信を SSL で保護するよう変更します。

ssl_xxxx のコメントアウトを外します。

ssl_keystore_path にノード証明書のパスを設定します。

ssl_keystore_password にノード証明書のパスワード(※1)を設定します。

Elasticsearch サーバーの OS が Linux の場合:

```
#ssl_verification_mode => none
#ssl_keystore_path     => "[keystore_path]"
#ssl_keystore_password => "[keystore_password]"
ssl_verification_mode => none
ssl_keystore_path     => "/etc/logstash/certs/http.p12"
ssl_keystore_password => "elanodepwd"
```

Elasticsearch サーバーの OS が Windows の場合:

```
#ssl_verification_mode => none
#ssl_keystore_path     => "[keystore_path]"
#ssl_keystore_password => "[keystore_password]"
ssl_verification_mode => none
ssl_keystore_path     => " C:¥temp¥logstash-8.11.3¥config¥certs¥elastic-
certificates.p12"
ssl_keystore_password => "elanodepwd"
```

修正後のファイルは以下のようになります。

```
...(省略)

output {

  if [type] == "get_ci_fil" {
    elasticsearch{
      hosts             => [ "https://eshost:9200" ]
      index             => "ssm_ci"
      document_id       => "%{ci_id}"
      action            => "update"
      retry_on_conflict => 10
      doc_as_upsert     => true
      pipeline          => "attachment"
      user              => "elastic"
      password          => "elapwd"
      ssl_enabled           => true
      ssl_verification_mode => none
      ssl_keystore_path     => " C:¥temp¥logstash-8.11.3¥config¥certs¥elastic-
certificates.p12"
      ssl_keystore_password => "elanodepwd"
    }

    if [data1] != "" {
      http {
        url => "https://eshost:9200/ssm_ci/_doc/%{ci_id}?pipeline=attachment"
        http_method => "put"
        headers => {
          "Content-Type" => "application/json"
          "Authorization" => "%[@metadata][basic_auth]]"
        }
        format => "json"
        message => "%{data1}"
        ssl_verification_mode => none
```

```
        ssl_keystore_path     => " C:¥temp¥logstash-8.11.3¥config¥certs¥elastic-
certificates.p12"
        ssl_keystore_password => "elanodepwd"
      }
    }

... (省略)
```

※1 …以下のコマンドを実行し、Elasticsearch のキーストアに格納されているノード証明書パスワードを確認します。

Elasticsearch サーバーの OS が Linux の場合:

("△" は半角スペースを示します。)

コマンド
/usr/share/elasticsearch/bin/elasticsearch-keystore△show△ xpack.security.http.ssl.keystore.secure_password

Elasticsearch サーバーの OS が Windows の場合:

("△" は半角スペースを示します。)

コマンド
cd△%elasticsearch_home%¥bin elasticsearch-keystore△show△xpack.security.http.ssl.keystore.secure_password

3. Logstash サービスを起動します。

Elasticsearch サーバーの OS が Linux の場合:

以下のコマンドを実行し、サービスを起動します。

("△" は半角スペースを示します。)

コマンド
systemctl△start△logstash

Elasticsearch サーバーの OS が Windows の場合:

以下のコマンドを実行し、サービスを起動します。

("△" は半角スペースを示します。)

コマンド
cd△%nssm_home%¥win64 nssm△start△Logstash サービス名

1.9.5 Senju Service Manager の設定

Senju Service Manager で提供する Elasticsearch 連携機能を使用する際にも

Elasticsearch の接続形態の設定が必要となります。

接続形態を「HTTPS」に設定してください。

必須設定

Elasticsearch 連携機能を利用する為には以下の設定が必須です。
・[管理者メニュー > 制御情報 > 制御情報 > 共通 > 共通]

管理項目名
Elasticsearch の接続形態

1.10 Elasticsearchの基本的な利用方法

Senju Service Manager で提供する Elasticsearch 連携機能の基本的な利用方法について説明します。

必須設定

Elasticsearch 連携機能を利用する為には以下の設定が必須です。
・[管理者メニュー > 制御情報 > 制御情報 > 共通 > 共通]

管理項目名
Elasticsearch の接続形態
Elasticsearch サーバー名
Elasticsearch ポート番号

仕様補足

Elasticsearch 連携機能では、プロセス管理の以下の項目に登録されている文字列を検索対象とします。

項目名
タイトル
参考 URL1 ～ 参考 URL7
テキスト 1 ～ テキスト 50
ラージテキスト 1 ～ ラージテキスト 4
コンボテキスト 1－テキスト ～ コンボテキスト 2－テキスト
マークダウン 1 ～ マークダウン 10
(表項目 1)テキスト 1 ～ テキスト 20
(表項目 2)テキスト 1 ～ テキスト 20
(表項目 3)テキスト 1 ～ テキスト 20
(表項目 4)テキスト 1 ～ テキスト 20
(表項目 5)テキスト 1 ～ テキスト 20
(経過)タイトル
(経過)内容
(経過)対応先
(経過)テキスト 1 ～ (経過)テキスト 4
(経過)テキストエリア 1 ～ (経過)テキストエリア 4

仕様補足

Elasticsearch 連携機能では、構成管理の以下の項目に登録されている文字列を検索対象とします。

項目名
構成アイテム名
参考 URL
テキスト 1 ～ テキスト 4
テキストエリア 1 ～ テキストエリア 3
コンボテキスト 1－テキスト ～ コンボテキスト 2－テキスト
添付ファイル 1 ～ 添付ファイル 2
JSON テキスト 1 ～ JSON テキスト 5

1.10.1全文検索

プロセス管理・構成管理の全てのレコードから、ユーザーが指定した単語を含むレコードを検索します。

仕様補足　本機能を利用するためには、全文検索ライセンスが必要です。

仕様補足　本機能はサービスデスクユーザーのみ利用可能です。

仕様補足　構成管理の全文検索は「現在の構成」のレコードのみを対象とします。

1. 検索語を指定してプロセス管理・構成管理のレコードを検索

画面右上の青いエリア（‹）をクリックすると、全文検索フィールドが表示され、検索語を入力して、検索アイコン（🔍）をクリックすると、全文検索画面が表示されます。

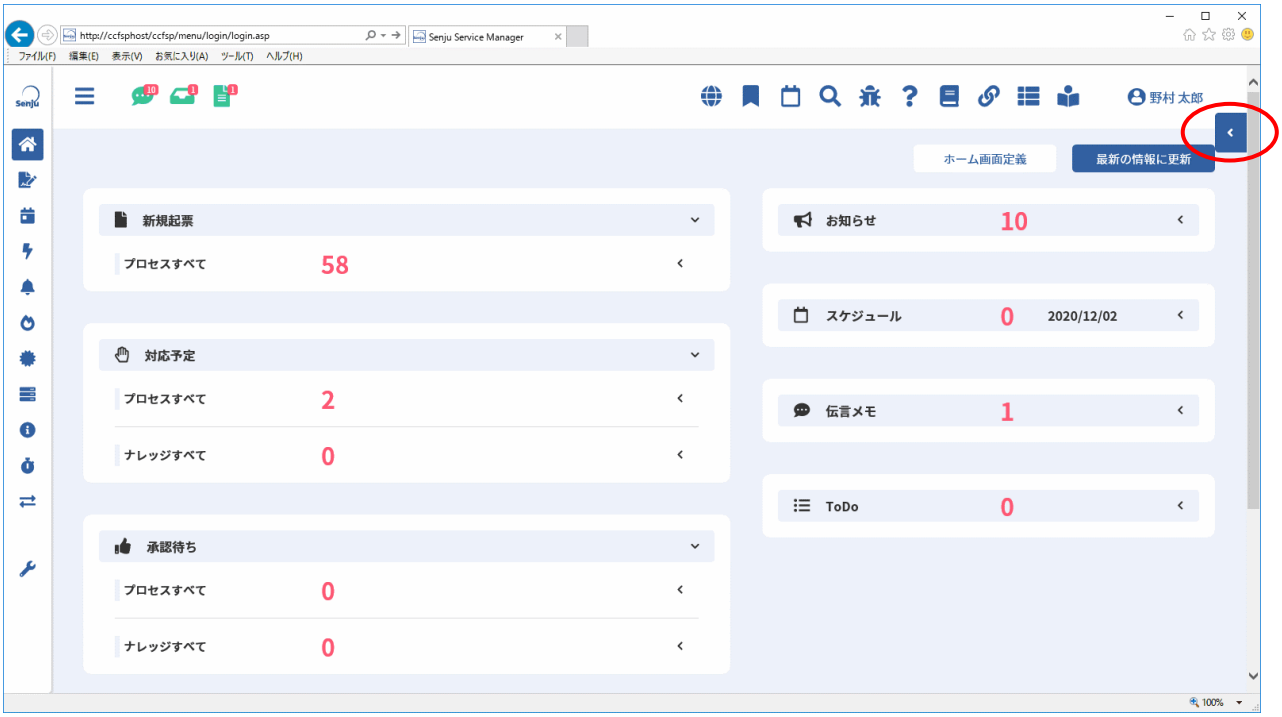

図 1-16 全文検索フィールド（縮小）

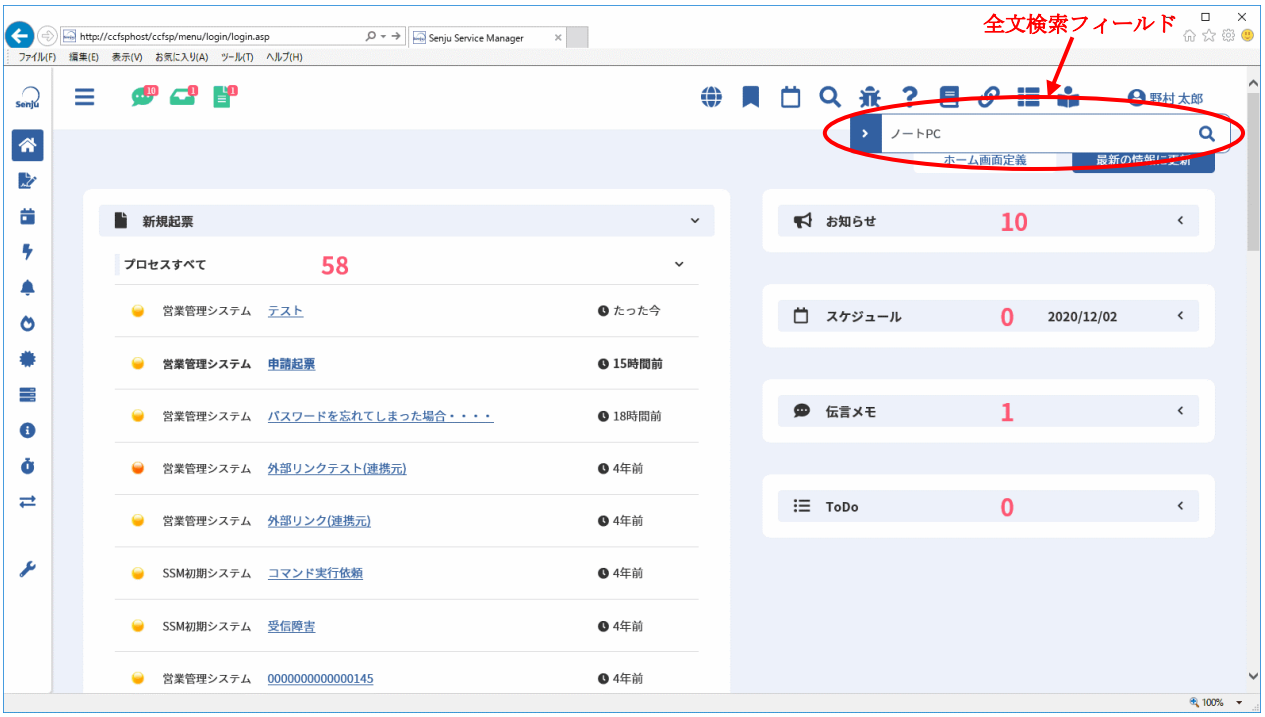

図 1-17 全文検索フィールド（拡大）

仕様補足

全文検索フィールドに検索語を入力して検索アイコンをクリックした場合、
プロセス管理と構成管理を横断した検索を行います。

全文検索画面には、Elasticsearch と連携してログインユーザーが参照可能なプロセス管理・構成管理のレコードの中から、入力した検索語が含まれる全てのレコードを検索結果一覧に表示します。

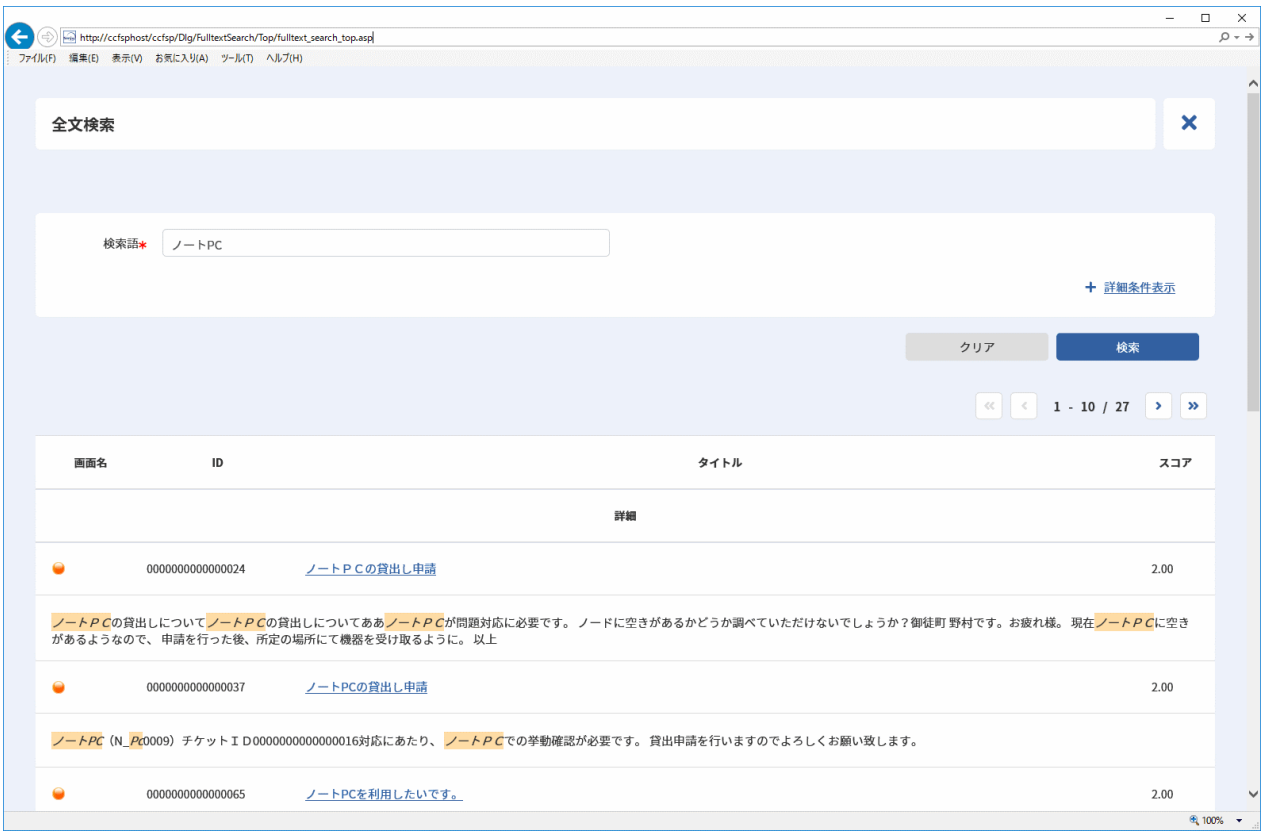

図 1-18 全文検索画面

仕様補足

全文検索では、Elasticsearch が提供する QueryStringQuery を採用し検索を実施しています。

仕様補足

スコアには、検索対象のレコードが検索語とどれだけ類似しているか、Elasticsearch が自動算出した数値を表示します。

仕様補足

詳細には、各レコードの検索語が含まれる項目データを表示します。また、該当の検索語はハイライト表示されます。

制限事項

検索結果に表示される件数は以下の管理項目で変更できます。
デフォルトは 10 件です。

・[管理者メニュー > 制御情報 > 制御情報 > 共通 > 共通]

管理項目名
全文検索一覧 1 ページ表示件数(件)

検索語に複数の単語を入力して検索した場合、入力した単語が全て含まれるレコードを検索結果一覧に表示します。また、複数の単語を入力する場合は、半角スペースまたは全角スペースで区切って入力してください。

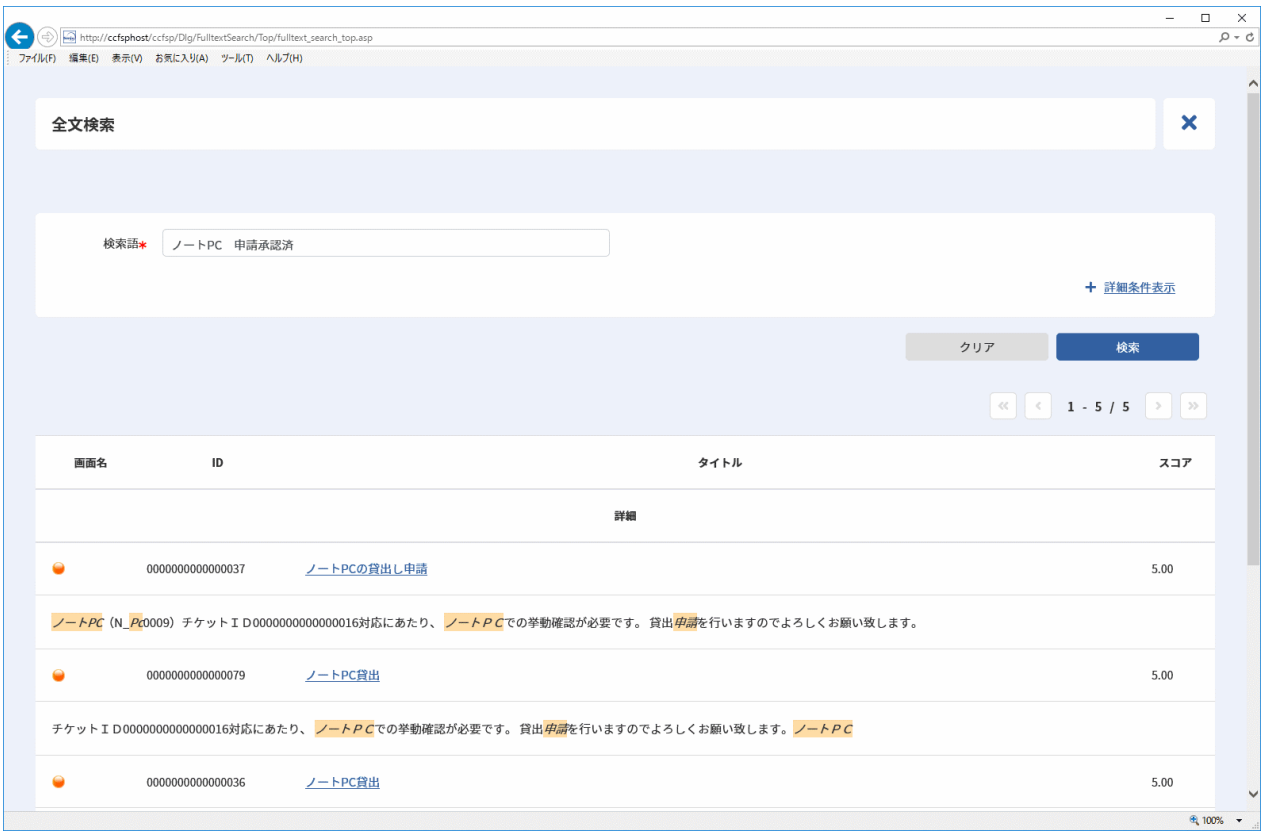

図 1-19 全文検索画面

仕様補足

検索語に複数の単語を入力した場合、AND 検索（指定した単語がいずれも含まれること）になります。

検索語を以下のように入力することで、検索処理方式を切り替えて検索することが可能です。

単語の入力状態	検索処理方式
半角ダブルクォテーション囲みなし	入力した単語を Elasticsearch の形態素解析に従って単語分割し、分割された最小単位の単語がいずれも含まれるレコードを検索する。
半角ダブルクォテーション囲みあり	入力した単語を 1 単語として扱い、その単語が含まれるレコードを検索する。

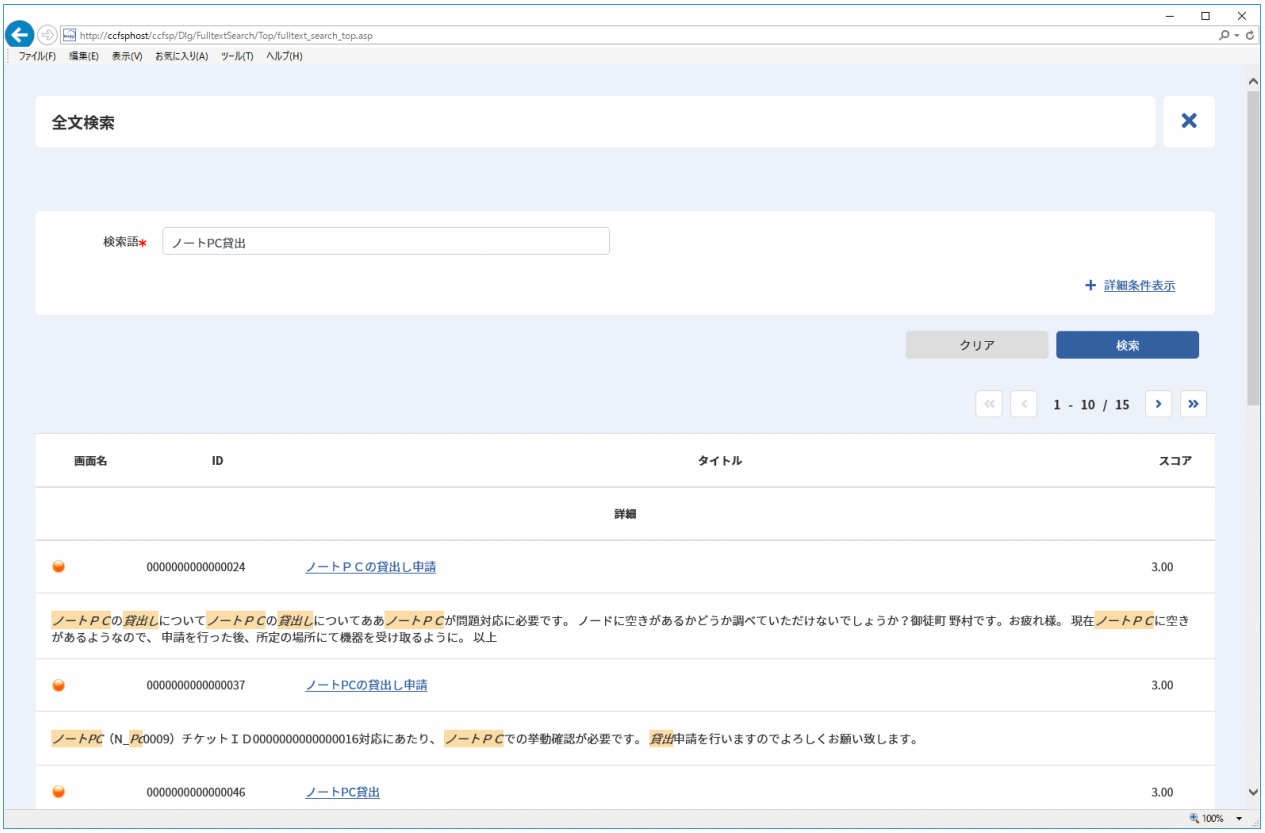

図 1-20 全文検索画面

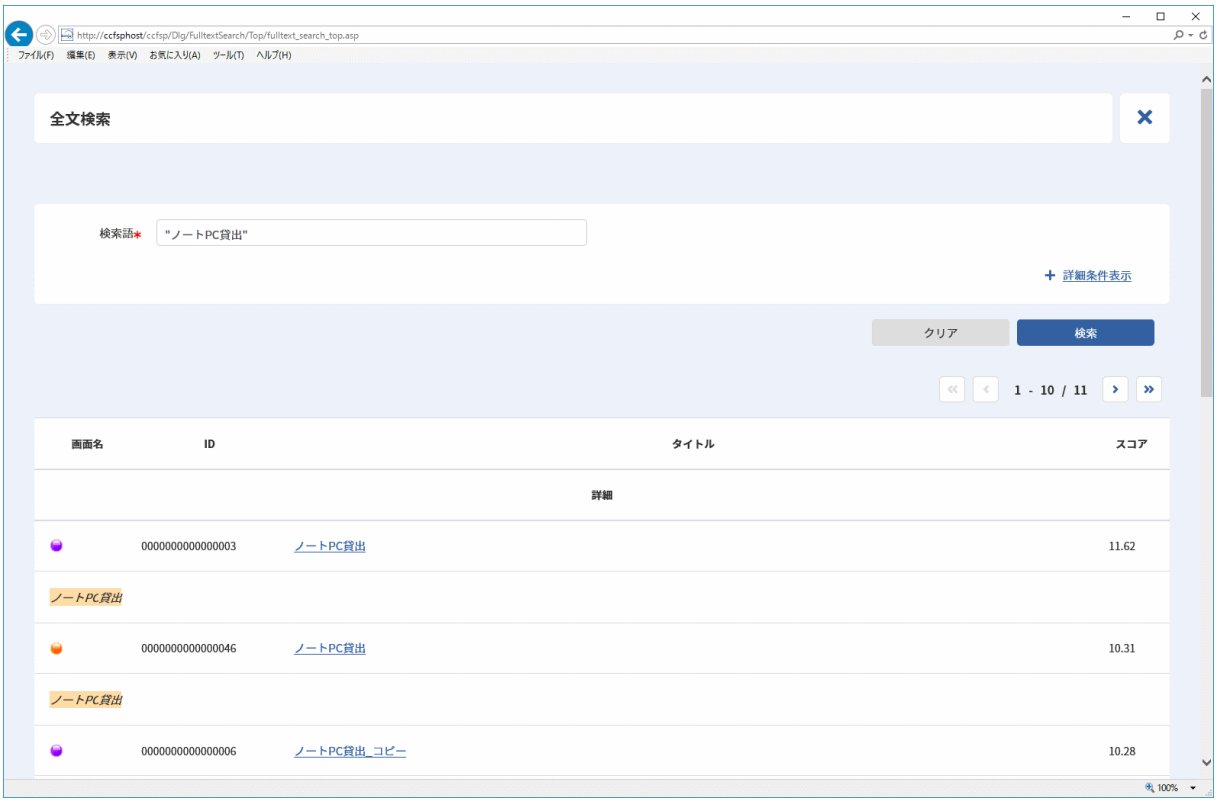

図 1-21 全文検索画面

仕様補足

全文検索は Elasticsearch の検索処理に基づき行われます。意図した検索結果とならないと感じる場合は、カタカナ単語末尾の長音を除く、半角ダブルクォテーションで囲むなどして再検索を実施してください。

仕様補足

半角・全角・ひらがな・カタカナは区別して検索します。

例えば、検索語に「センジュ」を入力して検索をした場合、「ｾﾝｼﾞｭ」が含まれるレコードは検索されません。

制限事項

半角ダブルクォテーション囲みの単語内に半角スペースまたは全角スペースが入力されている場合は、区切り文字として見なされますので、ご注意ください。

例えば、検索語に「"Senju[スペース]SM”」を入力して検索をした場合、
検索語に「Senju[スペース]SM」を入力して検索した結果と同様になります。

制限事項

検索対象文字列に句読点が含まれている場合は、取り除かれて検索されます。

例えば、検索語に「"〇〇□□△△"」を入力して検索をした場合、
「〇〇。□□、△△」の文字列が含まれるレコードは検索されます。

2. 詳細な条件を指定してプロセス管理・構成管理のレコードを検索

より詳細な条件で検索したいときは、[詳細条件表示]ボタンをクリックします。

以下の検索条件項目が表示され、詳細条件を指定することができます。

検索条件	説明
画面名	選択した画面を検索範囲とする。
システム名	選択したシステムを検索範囲とする。
起票日	現在日付から起票日が選択した項目以前のレコードを検索範囲とする。
最終更新日	現在日付から最終更新日が選択した項目以前のレコードを検索範囲とする。

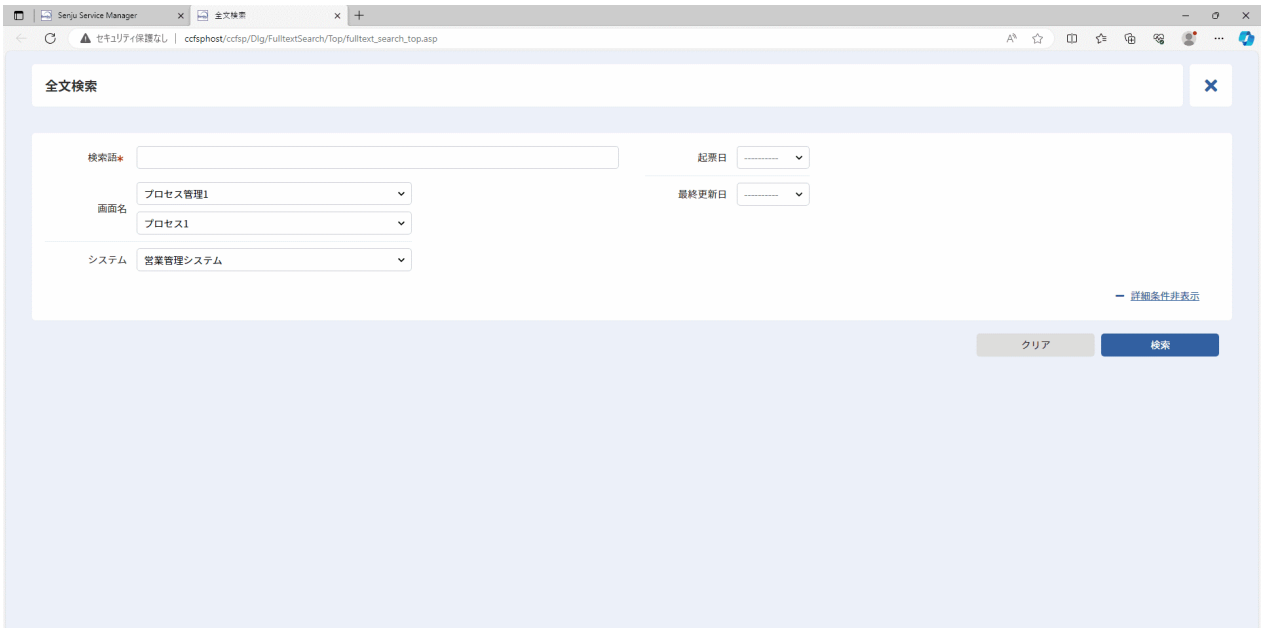

図 1-22 全文検索画面（詳細条件表示）

仕様補足

検索条件で画面の絞り込みを行わない場合、プロセス管理と構成管理を横断した検索を行います。

3. プロセス管理・構成管理の詳細を参照

検索結果一覧でレコードのタイトルをクリックすると、プロセス詳細画面または構成アイテム詳細画面が表示されます。

図 1-23 全文検索画面

図 1-24 プロセス詳細画面

1.10.2類似検索

プロセス管理の全てのレコードから、参照中のプロセスのレコードに対する、類似するレコードを検索します。

本機能を利用するためには、類似検索ライセンスが必要です。

1. 類似プロセスを表示

プロセスの詳細項目に類似プロセスを表示することで類似検索を利用することができます。
類似プロセスには、Elasticsearch と連携して、ログインユーザーが参照可能なプロセス管理のレコードの中から、類似するレコードをスコアの降順で表示します。

類似プロセス　(取得日時 2020/12/02 03:04:53)

種別	ID	タイトル	スコア
	0000000000000014	システムの削除	9.08
	0000000000000133	ビットマップ張り替え(後で削除)	3.86
	0000000000000129	XXシステムの夜間性能遅延について[ス必][バチ]	3.42
	0000000000000034	Server004を占有させてください。	2.61
	20080117-0000054	勤怠管理システムでの不具合	2.49
	0000000000000107	アカウントがロックされてしまいました。	2.46
	0000000000000096	（問合せ）サーバー障害時の対応について	2.12
	0000000000000063	Server004を占有させてください。	2.09
	0000000000000065	パスワードの有効期限	1.73
	0000000000000146	アカウントがロックされてしまいました。	1.62

図 1-25 類似プロセス項目

仕様補足

類似検索では、Elasticsearch が提供する More Like This Query を採用し検索を実施しています。

仕様補足

取得日時には、プロセス画面を表示した際の日時情報を表示します。

仕様補足

スコアには、参照中のレコードと類似するレコードがどれだけ類似しているか、Elasticsearch が自動算出した数値を表示します。

仕様補足

類似検索を使用する場合は、以下の画面より類似プロセスをプロセス詳細画面に表示してください。

・[管理者メニュー ＞ 画面 ＞ レイアウト定義 ＞ サービスデスク画面レイアウト]

対象項目
類似プロセス

制限事項

類似プロセスのレコード表示件数の上限は 10 件です。

制限事項

以下の画面では、類似検索が行われないため、類似プロセスにレコードは表示されません。

・プロセス新規登録画面
・プロセス編集画面　※初期表示時のみ類似検索します

1.10.3チャットボット連携

Senju Service Manager に蓄積した情報から、ユーザーの質問に対する回答を検索します。

仕様補足

本機能を利用するためには、OpenAI または Azure OpenAI との契約が必要です。

仕様補足

本機能を利用するためには、全文検索ライセンスが必要です。

仕様補足

本機能はサービスデスクユーザーのみ利用可能です。

必須設定

チャットボット連携機能を利用する為には以下の設定が必須です。
・[管理者メニュー > 制御情報 > 制御情報 > 共通 > 共通]

管理項目名
OpenAI の接続方式
OpenAI 検索キーワード用事前指示文設定
OpenAI 回答取得用事前指示文設定
OpenAI の GPT モデル
OpenAI の温度
OpenAI の上限トークン数
OpenAI のキー
Azure OpenAI のキー
Azure OpenAI 場所/地域
Azure OpenAI エンドポイント
Azure OpenAI バージョン
Azure OpenAI ネットワークエージェント

1. 利用方法

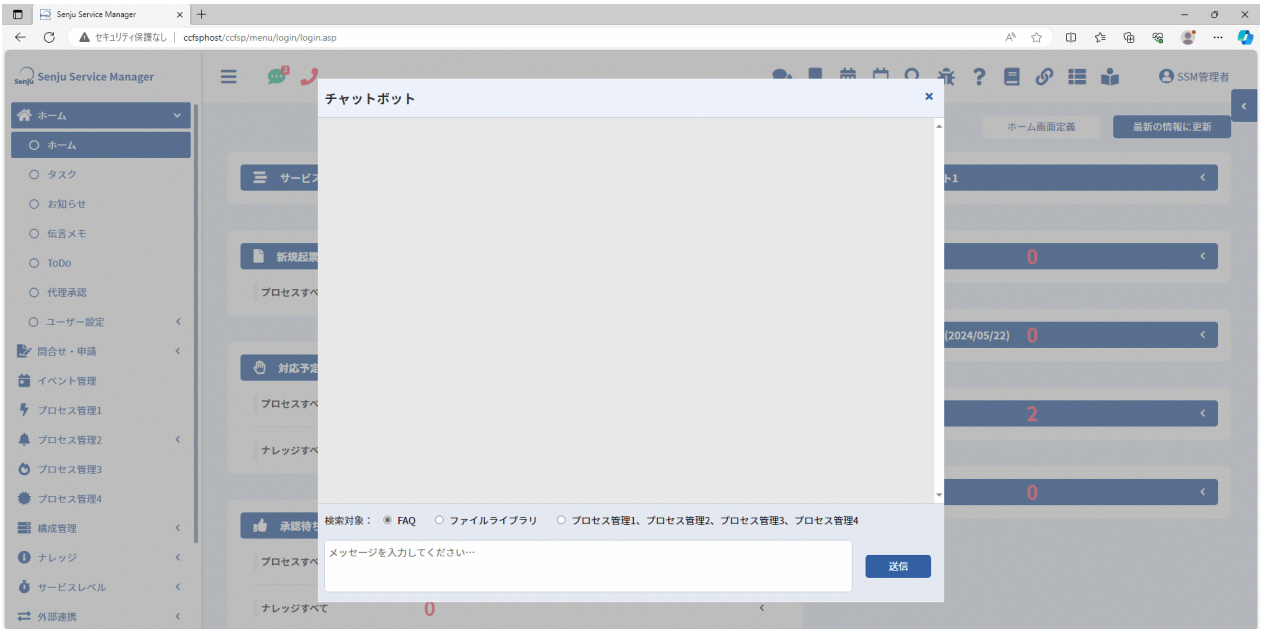

図 1-26 チャットボットダイアログ

ログインフレームに表示されているチャットボットアイコンをクリックすると、チャットボットダイアログが表示されます。

検索対象(FAQ、ファイルライブラリ、プロセス管理)を選択し、メッセージに質問内容を入力後、「送信」ボタンをクリックすると回答が表示されます。

右上の×アイコンをクリックすると、チャットボットダイアログが閉じます。

仕様補足

質問に対する回答が Senju Service Manager に存在しない場合、「申し訳ございません、この質問の回答が見つかりませんでした」というメッセージを返却します。

1.11 トラブルシューティング

ElasticSearch 連携がうまく動作しない場合の確認方法を記載します。

1.11.1データを再収集する

Kibana を利用している方は本手順を実施する前にバックアップとして
別紙 Kibana 連携機能ガイドの「インポート手順」を実施してください。

注意 データの再収集時の手順を誤った際に Kibana で作成したグラフ等が削除されてしまう可能性があります。

1.11.1.1 Linux 版 Elasticsearch の場合

Senju Service Manager から Elasticsearch へのデータ連携のためのアプリケーションとして Logstach を利用していますが、Logstash の動作が不正になった場合に、一度 ElasticSearch に保存されているデータをリセットしたうえで、再収集する必要がある場合があります。

本手順を実施すると再収集が完了するまで、全文検索、類似検索が利用できない状態となります。

データのリセットおよび再収集の手順は以下の通りとなります。

1. Logstash サービスを停止します。
   以下のコマンドを実行し、サービスを停止します。

コマンド
systemctl△stop△logstash

   以下のコマンドを実行し、サービスの状態を確認します。

コマンド
systemctl△status△logstash

   出力内容中に以下の内容が表示されることを確認します。

Active: inactive (dead)

2. Logstash の取り込み履歴を削除します。
   以下のコマンドを実行し、Logstash の取り込み履歴ファイル格納ディレクトリに移動します。

コマンド
cd△/opt/logstash/conf

※上記パスはデフォルト設定先になります。履歴ファイルの格納場所は、以下ファイル内の"last_run_metadata_path =>"の設定値をご確認ください。

・データベースが Oracle である場合：

<ディレクトリ> /etc/logstash/conf.d

ファイル名	説明
logstash-oracle.conf	設定ファイル（プロセス管理）
logstash-oracle_ci.conf	設定ファイル（構成管理）
logstash-oracle-faq.conf	設定ファイル（FAQ）
logstash-oracle-filelibrary.conf	設定ファイル（ファイルライブラリ）

以下のコマンドを実行し、履歴ファイルを削除します。

コマンド
rm△-f△.logstash_oracle_*_last_run

・データベースが PostgreSQL である場合：

<ディレクトリ> /etc/logstash/conf.d

ファイル名	説明
logstash-postgresql.conf	設定ファイル（プロセス管理）
logstash-postgresql_ci.conf	設定ファイル（構成管理）
logstash-postgresql-faq.conf	設定ファイル（FAQ）
logstash-postgresql-filelibrary.conf	設定ファイル（ファイルライブラリ）

以下のコマンドを実行し、履歴ファイルを削除します。

コマンド
rm△-f△.logstash_postresql_*_last_run

以下のコマンドを実行し、Logstash のキューファイル格納ディレクトリに移動します。

コマンド
cd△/var/lib/logstash/

以下のコマンドを実行し、ファイルの情報を確認します。

コマンド
ls

以下の内容が出力されることを確認します。

queue uuid

以下のコマンドを実行し、キューファイルを削除します。

コマンド
rm△-rf△queue rm△-rf△uuid

3. ElasticSearch のインデックス情報を削除します。

・Elasticsearch 認証の設定が未実施の場合：

以下のコマンドを実行し、現在のインデックス情報を確認します。

コマンド
curl△-XGET△"[protocol]://[hostname]:[portnumber]/_aliases?pretty"

例）Elasticsearch への接続情報でプロトコル[protocol]:http、ホスト名[hostname]:eshost、ポート番号[portnumber]:9200 とした場合は、以下のコマンドを実行します。

コマンド
curl△-XGET△"http://eshost:9200/_aliases?pretty"

以下の内容が出力されることを確認します。

```
{
  "ssm" : {
    "aliases" : { }
  },
  "ssm_ci" : {
    "aliases" : { }
  },
  "ssm_faq" : {
    "aliases" : { }
  },
  "ssm_filelibrary" : {
    "aliases" : { }
  }
}
```

以下のコマンドを実行し、インデックス情報を削除します。

コマンド
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm?pretty" curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_ci?pretty" curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_faq?pretty" curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_filelibrary?pretty"

例）Elasticsearch への接続情報でプロトコル[protocol] → http、ホスト名[hostname] → eshost、ポート番号[portnumber] → 9200 とした場合は、以下のコマンドを実行します。

コマンド
curl△-XDELETE△"http://eshost:9200/ssm?pretty" curl△-XDELETE△"http://eshost:9200/ssm_ci?pretty" curl△-XDELETE△"http://eshost:9200/ssm_faq?pretty" curl△-XDELETE△"http://eshost:9200/ssm_filelibrary?pretty"

以下の内容が出力されることを確認します。

```
{
  "acknowledged" : true
}
```

・Elasticsearch 認証の設定が実施済の場合：

以下のコマンドを実行し、現在のインデックス情報を確認します。

コマンド
curl△-XGET△"[protocol]://[hostname]:[portnumber]/_aliases?pretty"△-u△[elasticuser]:[elasticpassword]

例）Elasticsearch への接続情報でプロトコル[protocol]:http、ホスト名[hostname]:eshost、ポート番号[portnumber]:9200、認証ユーザー名[elasticuser]:elastic、認証パスワード[elastipassword]:elapwd とした場合は、以下のコマンドを実行します。

コマンド
curl△-XGET△"http://eshost:9200/_aliases?pretty"△-u△elastic:elapwd

以下の内容が出力されることを確認します。

```
{
  "ssm" : {
    "aliases" : { }
  },
  "ssm_ci" : {
    "aliases" : { }
  },
  "ssm_faq" : {
    "aliases" : { }
  },
  "ssm_filelibrary" : {
    "aliases" : { }
  }
}
```

以下のコマンドを実行し、インデックス情報を削除します。

コマンド
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm?pretty"△-u△[elasticuser]:[elasticpassword] curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_ci?pretty"△-u△[elasticuser]:[elasticpassword] curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_faq?pretty"△-u△[elasticuser]:[elasticpassword] curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_filelibrary?pretty"△-u△[elasticuser]:[elasticpassword]

例）Elasticsearch への接続情報でプロトコル[protocol]:http、ホスト名[hostname]:eshost、ポート番号[portnumber]:9200、認証ユーザー名[elasticuser]:elastic、認証パスワード

[elastipassword]:elapwd とした場合は、以下のコマンドを実行します。

コマンド
curl△-XDELETE△"http://eshost:9200/ssm?pretty"△-u△elastic:elapwd
curl△-XDELETE△"http://eshost:9200/ssm_ci?pretty"△-u△elastic:elapwd
curl△-XDELETE△"http://eshost:9200/ssm_faq?pretty"△-u△elastic:elapwd
curl△-XDELETE△"http://eshost:9200/ssm_filelibrary?pretty"△-u△elastic:elapwd

以下の内容が出力されることを確認します。

```
{
  "acknowledged" : true
}
```

注意

削除コマンドの対象のインデックスには必ず以下を指定してください。
・ssm
・ssm_ci
・ssm_faq
・ssm_filelibrary
「*」を指定した場合、Elasticsearch に関するインデックスも削除されてしまいます。関係ないインデックスを削除してしまった際には Elasticsearch 認証の設定を再度実施してください。
また、Kibana を利用している方は別紙 Kibana 連携機能ガイドの「エクスポート手順」を実施することで作成したグラフ等を復元することができます。

補足

Elasticsearch 認証した状態で curl コマンドを実施する際には-u オプションでユーザー名とパスワードを追加してください。

4. インデックスを再作成します。

   本手順は、手順「1.4.1.2-6 インデックスの作成」を実施してください。

5. Logstash サービスを起動します。

   以下のコマンドを実行し、サービスを起動します。

コマンド
systemctl△start△logstash

以下のコマンドを実行し、サービスの状態を確認します。

コマンド
systemctl△status△logstash

出力内容中に以下の内容が表示されることを確認します。

Active: active (running)

6. Logstash のログ情報を確認します。

   以下のログファイルの内容を確認し、Logstash サービス起動後の時間帯にエラー出力がないことを確認します。

<確認ファイルパス>

/var/log/logstash/logstash-plain.log

1.11.1.2 Windows 版 Elasticsearch の場合

データのリセットおよび再収集の手順は以下の通りとなります。

1. 以下のコマンドを実行し、Logstash サービスを停止します。
(“△” は半角スペースを示します。)

コマンド
cd△%nssm_home%¥win64 nssm△stop△Logstash サービス名

以下のコマンドを実行し、サービスの状態を確認します。
(“△” は半角スペースを示します。)

コマンド
cd△%nssm_home%¥win64 nssm△status△Logstash サービス名

以下のような出力結果があることを確認します。

SERVICE_STOPPED

2. Logstash の取り込み履歴を削除します。
以下のコマンドを実行し、Logstash の取り込み履歴ファイル格納ディレクトリに移動します。

コマンド
cd△C:¥temp¥logstash-8.11.2¥logstash¥conf

※上記パスはデフォルト設定先になります。履歴ファイルの格納場所は、以下ファイル内の"last_run_metadata_path =>"の設定値をご確認ください。

・データベースが Postgres である場合：

<ディレクトリ> C:\temp\logstash-8.11.2\logstash-definitions

ファイル名	説明
logstash-postgresql.conf	設定ファイル(プロセス管理)
logstash-postgresql_ci.conf	設定ファイル(構成管理)
logstash-postgresql-faq.conf	設定ファイル(FAQ)
logstash-postgresql-filelibrary.conf	設定ファイル(ファイルライブラリ)

以下のコマンドを実行し、履歴ファイルを削除します。

コマンド
del△/Q△.logstash_postgresql_*_last_run

・データベースが Oracle である場合：

<ディレクトリ> C:\temp\logstash-8.11.2\logstash-definitions

ファイル名	説明
logstash-oracle.conf	設定ファイル(プロセス管理)
logstash-oracle_ci.conf	設定ファイル(構成管理)
logstash-oracle-faq.conf	設定ファイル(FAQ)
logstash-oracle-filelibrary.conf	設定ファイル(ファイルライブラリ)

以下のコマンドを実行し、履歴ファイルを削除します。

コマンド
del△/Q△.logstash_oracle_*_last_run

以下のコマンドを実行し、Logstash のキューファイル格納ディレクトリに移動します。
(“△” は半角スペースを示します。)

コマンド
cd△C:¥temp¥logstash-8.11.3¥data

以下のコマンドを実行し、ファイルの情報を確認します。

コマンド
dir

以下の内容が出力されることを確認します。
例:

```
2021/02/09  20:13    <DIR>          dead_letter_queue
2021/02/09  20:13                36 uuid
```

キューファイル「dead_letter_queue」とキューファイル「uuid」を削除します。

3. 以下のコマンドを実行し、現在のインデックス情報を確認します
・Elasticsearch 認証の設定が未実施の場合：

コマンド
curl△-XGET△"[protocol]://[hostname]:[portnumber]/_aliases?pretty""

例）Elasticsearch への接続情報でプロトコル[protocol]:http、ホスト名[hostname]:eshost、ポート番号[portnumber]:9200 とした場合は、以下のコマンドを実行します。

コマンド
curl△-XGET△"http://eshost:9200/_aliases?pretty"

・Elasticsearch 認証の設定が実施済の場合：

コマンド
curl△-XGET△"[protocol]://[hostname]:[portnumber]/_aliases?pretty" △-u△[elasticuser]:[elasticpassword]

例）Elasticsearch への接続情報でプロトコル[protocol]:http、ホスト名[hostname]:eshost、ポート番号[portnumber]:9200、認証ユーザー名[elasticuser]:elastic、認証パスワード[elastipassword]:elapwd とした場合は、以下のコマンドを実行します。

コマンド
curl△-XGET△"http://eshost:9200/_aliases?pretty"△-u△elastic:elapwd

以下の内容が出力されることを確認します。

```
{
  "ssm" : {
    "aliases" : { }
  },
  "ssm_ci" : {
    "aliases" : { }
  },
  "ssm_faq" : {
    "aliases" : { }
  },
  "ssm_filelibrary" : {
    "aliases" : { }
  }
}
```

以下のコマンドを実行し、インデックス情報を削除します。
(“△” は半角スペースを示します。)

・Elasticsearch 認証の設定が未実施の場合：

コマンド
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm?pretty"
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_ci?pretty"
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_faq?pretty"
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_filelibrary?pretty"

例）Elasticsearch への接続情報でプロトコル[protocol] → http、ホスト名[hostname] → eshost、ポート番号[portnumber] → 9200 とした場合は、以下のコマンドを実行します。

コマンド
curl△-XDELETE△"http://eshost:9200/ssm?pretty"
curl△-XDELETE△"http://eshost:9200/ssm_ci?pretty"
curl△-XDELETE△"http://eshost:9200/ssm_faq?pretty"
curl△-XDELETE△"http://eshost:9200/ssm_filelibrary?pretty"

・Elasticsearch 認証の設定が実施済の場合：

コマンド
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm?pretty"△-u△[elasticuser]:[elasticpassword]
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_ci?pretty"△-u△[elasticuser]:[elasticpassword]
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_faq?pretty"△-u△[elasticuser]:[elasticpassword]
curl△-XDELETE△"[protocol]://[hostname]:[portnumber]/ssm_filelibrary?pretty"△-u△[elasticuser]:[elasticpassword]

例）Elasticsearch への接続情報でプロトコル[protocol]:http、ホスト名[hostname]:eshost、ポート番号[portnumber]:9200、認証ユーザー名[elasticuser]:elastic、認証パスワード[elastipassword]:elapwd とした場合は、以下のコマンドを実行します。

コマンド
curl△-XDELETE△"http://eshost:9200/ssm?pretty" △-u△elastic:elapwd
curl△-XDELETE△"http://eshost:9200/ssm_ci?pretty" △-u△elastic:elapwd
curl△-XDELETE△"http://eshost:9200/ssm_faq?pretty" △-u△elastic:elapwd
curl△-XDELETE△"http://eshost:9200/ssm_filelibrary?pretty" △-u△elastic:elapwd

以下の内容が出力されることを確認します。

```
{
    "acknowledged" : true
}
```

4. インデックスを再作成します
本手順は、手順「1.5.2-5 インデックスの作成」を実施してください。

5. Logstash サービスを起動します。
以下のコマンドを実行し、サービスを起動します。
（“△” は半角スペースを示します。）

コマンド
cd△%nssm_home%¥win64 nssm△start△Logstash サービス名

以下のコマンドを実行し、サービスの状態を確認します。
（“△” は半角スペースを示します。）

コマンド
cd△%nssm_home%¥win64 nssm△status△Logstash サービス名

以下のような出力結果があることを確認します。

```
SERVICE_RUNNING
```

6. Logstash のログ情報を確認します。

以下のログファイルの内容を確認し、Logstash サービス起動後の時間帯にエラー出力がないことを確認します。

<確認ファイルパス>
C:¥temp¥logstash-8.11.3¥logs¥logstash-plain.log

1.12 制限事項

Senju Service Manager で提供する Elasticsearch 連携機能について制限事項を以下に示します。

1) Elasticsearch へのデータ連携処理は 1 分間隔で行われるため、プロセスの新規登録または編集直後のレコードが、Elasticsearch に反映されるまで多少のタイムラグがあります。

そのため、以下の事象が発生する場合があります。

1. 全文検索
   - 検索語に新規登録または編集直後のレコードがヒットする単語で全文検索した場合、データ連携前のため、新規登録または編集直後のレコードは、検索されません。
2. 類似検索
   - 新規登録直後のレコードが Elasticsearch に連携されるまで、類似するレコードは表示されません。
   - 編集直後のレコードの更新内容が Elasticsearch に連携されるまで、更新前の登録内容で類似検索が行われます。

2) Elasticsearch に連携済のプロセスについて、Elasticsearch 連携機能の検索対象のプロセス項目を詳細レイアウトから非表示にしたとしても、非表示にした項目のデータは Elasticsearch に連携したままとなります。

そのため、以下の事象が発生する場合があります。

1. 全文検索
   - 検索語に非表示にした項目のデータに存在する単語で全文検索した場合、その単語が含まれるレコードは、検索されます。
2. 類似検索
   - 非表示にした項目のデータも含め類似検索が行われます。

3) Elasticsearch に同期済のレコードは Senju Service Manager で削除されたら、ElasticSearch データ削除モジュールによって対象データが ElasticSearch から削除されます。但し、削除が反映されるまで 5 分のほどの時間を要します。
※削除処理についての説明は別紙「コマンドリファレンス」の「1.2.25 ElasticSearch データ削除モジュール(sjSPU_ElasticSearchDelete.vbs)」を参照してください。

そのため、削除後の経過時間が 5 分未満のレコードが引き続き全文検索および類似検索の結果として表示されますが、対象レコードのタイトルをクリックすると「指定された情報は既に削除されています」と表示されます。

4) Elasticsearch より取得できるレコードの上限数はデフォルト 10000 件となっています。

そのため、以下の事象が発生する場合があります。

1. 全文検索

・ 検索結果一覧に 10000 件以降のレコードを表示すると「Elasticsearch エラーが発生しました」とアラート表示されます。

以下のコマンドにて取得できるレコードの上限数は変更可能となりますが、性能に影響がでる可能性があります。

(“△” は半角スペースを示します。)

コマンド

curl△-XPUT△'http://localhost:9200/ssm/_settings?pretty=true'△-d'
{
"index" : {
"max_result_window" : 100000
}
}'