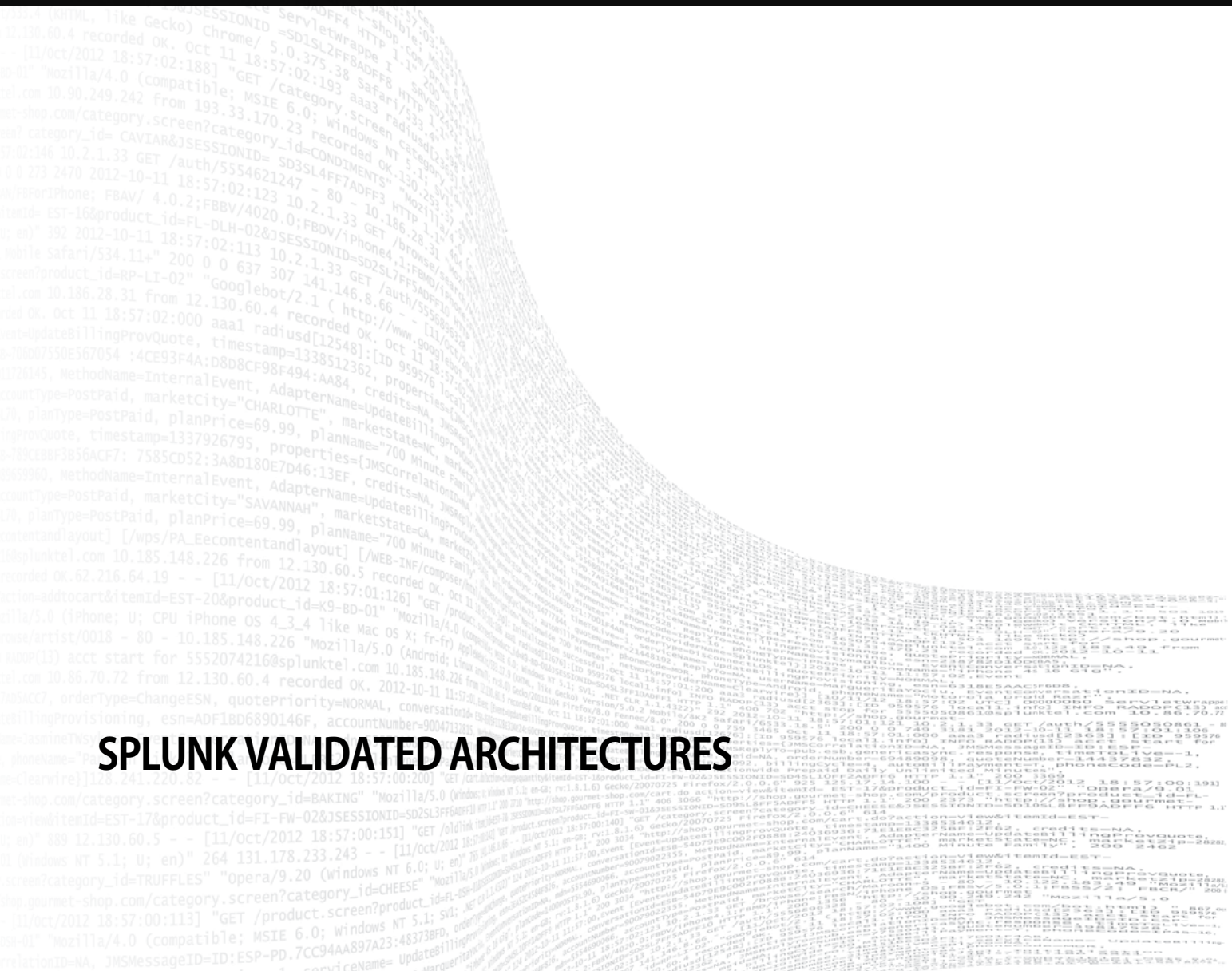
splunk>

ホワイトペーパー

SPLUNK VALIDATED ARCHITECTURES



目次

はじめに

Splunk Validated Architecture（SVA）は、効率的で再現性に優れた Splunk の安定したデプロイを保証する、実証されたリファレンスアーキテクチャです。Splunk の既存ユーザーの多くは、導入と拡張を急いだ結果、デプロイのさらなる拡張において困難に直面しています。また、新しい Splunk ユーザーも、デプロイが最初から確かな基礎の上に構築されることを保証するためのガイドラインと認定アーキテクチャを求めています。SVA は、ますます高まっているこれらのニーズにおいてユーザーを支援するために開発されました。

SVA は、Splunk の新規ユーザーまたは既存ユーザーのどちらであっても、管理を容易にし、トラブルシューティングをシンプルにしてくれる環境の構築に役立ちます。SVA は、総所有コストを最小限に抑えつつ、可能な限り最高の結果をもたらすように設計されています。さらに、Splunk の基礎全体が再現可能なアーキテクチャをベースとするため、時間の経過と共にニーズが増大した場合には、デプロイを安心して拡張できます。

SVA は、幅広い組織の要件を考慮したトポロジーオプションを用意しているため、要件に最適なトポロジを容易に理解して見つけることができます。Splunk Validated Architecture の選択プロセスにより、特定の要件を組織のニーズに最も適したトポロジーに対応させることができます。Splunk を初めて使用する場合は、SVA を実装することをお勧めします。すでに Splunk を使用している場合は、SVA のトポロジーに沿ったオプションを検討することをお勧めします。どうしてもカスタムアーキテクチャを作らなければ対応できないような独特の要件の場合には、高いコスト効果を維持しながら SVA でニーズを満たすのは難しいでしょう。

このホワイトペーパーでは、SVA の概要について解説します。また、このホワイトペーパーには、要件の質問票、デプロイのトポロジーダイアグラム、設計方針、全般的なガイドラインなど、SVA 選択プロセスで必要となるリソースがあります。

Splunk Validated Architecture の実装についてのご質問は、Splunk プロフェッショナルサービス（https://www.splunk.com/en_us/support-and-services/splunk-services.html）までお寄せください。

本書の構成

SVA は、主に 3 つの部分から構成されています。

1. インデックス作成とサーチのトポロジー
2. データ収集アーキテクチャのコンポーネント
3. 設計方針とベストプラクティス

インデックス作成とサーチは、Splunk デプロイの中核となるインデックス作成とサーチ機能を提供するアーキテクチャ層をカバーします。データ収集コンポーネントのセクションでは、特定の要件に適したデータ収集メカニズムを選ぶ方法を説明します。

設計方針とベストプラクティスは、アーキテクチャ全体に適用され、デプロイの詳細を定義する際に正しい選択ができるようにします。

Splunk Validated Architecture を使用する理由

SVA を実装することで、Splunk の設計とデプロイをより確実に行うことができます。SVA は、大半の組織が直面する次のような問題の解決に役立ちます。

パフォーマンス

- パフォーマンスと安定性が改善します。

複雑さ

- カスタムビルドのデプロイ、特に急速にまたは複雑に拡張されたデプロイは、不必要に複雑になりすぎる場合があります。この複雑さは、以降の拡張において大きな障害となることがあります。

効率

- Splunk デプロイの利点を最大限に引き出すためには、運営の効率を改善し、導入効果の発現を加速させる必要があります。

コスト

- 組織は、すべての要件を満足しつつ、総所有コスト（TCO）を削減する道を模索しています。

アジリティ

- 組織が成長して規模が拡張されるに従って、その変化に適応する必要があります。

メンテナンス

- 多くの場合、メンテナンスの労力を抑えるには環境の最適化が必要です。

拡張性

- 組織は、効率的かつシームレスに規模を拡張できる必要があります。

検証

- 組織内の関係者は、Splunk のデプロイがベストプラクティスに基づいて構築されていることを求めます。

Splunk Validated Architecture の柱

Splunk Validated Architecture は、以下の基本的な柱を中心に構築されています。これらの設計の柱の詳細については、付録 A を参照してください。

可用性	パフォーマンス	拡張性	セキュリティ	管理性
システムが**常に稼動**していて、計画的あるいは予期しない停止や中断から復旧できること	利用パターンが変動しても**最適なレベルのサービス**をシステムが維持できること	すべての層で拡張性を保証することで**増大したワークロードを効果的に処理**できるようにシステムが設計されていること	高価値を提供しつつ**データ、構成、そしてアセットを保護**するようにシステムが設計されていること	**すべての層で一括運営、一括管理**できるようにシステムが設計されていること

これらの柱は Splunk Center Of Excellence モデルの **プラットフォーム管理 & サポート** サービスで直接サポートされます。

Splunk Validated Architecture に期待されるもの

SVA にはデプロイ技術やデプロイサイジングは含まれません。その理由は以下の通りです。

- オペレーティングシステムやサーバーハードウェアなどのデプロイ技術は、SVA では実装上の選択肢と考えられています。ユーザーが違えば選択肢も異なるので、一般化することは困難です。
- デプロイサイジングには、取り込むデータ量、データタイプ、サーチボリューム、サーチの使用事例の評価が必要であり、ユーザーごとに大きく異なるうえ、一般的には基礎となるデプロイアーキテクチャそのものには関係ありません。このプロセスは、でプログラムアーキテクチャを確立してから、既存のサイジングツールで対応できます。Splunk ストレージサイジング（https://splunk-sizing.appspot.com/）などのツールを利用できます。

SVA が提供するもの:	SVA が 提供しないもの:
✅ クラスタ化および非クラスタ化デプロイオプション ✅ リファレンスアーキテクチャのダイアグラム ✅ 的確なアーキテクチャを選択するためのガイドライン ✅ 層特有の推奨事項 ✅ Splunk デプロイを構築するためのベストプラクティス	❌ 実装の選択肢（OS、ベアメタルか仮想かクラウドか、等）. ❌ デプロイサイジング ❌ アーキテクチャの規範的な承認。注意： SVA は推奨事項とガイドラインは提供しますが、自分の組織にとって適切な意思決定をするのはユーザー自身です ❌ すべての可能なデプロイシナリオに対するトポロジー提案。場合によっては、独自の要因によってカスタムアーキテクチャの開発が必要になることもあります。必要なカスタムソリューションについては、Splunk の専門家から助言を得ることができます。すでに Splunk を使用している場合は、Splunk アカウントチームまでご連絡ください。初めて Splunk を使用する場合はこちらからご連絡ください（https://www.splunk.com/en_us/talk-to-sales.html）

役割と責任

Splunk Validated Architecture は、意志決定者や管理者に大きく関係しています。SVA 選択プロセスには、エンタープライズアーキテクト、コンサルタント、Splunk 管理者、およびマネージドサービスプロバイダが全員参加してください。それぞれの役割について以下に説明します。

役割	説明
エンタープライズアーキテクト	企業のニーズに合わせて Splunk デプロイを設計します。
コンサルタント	Splunk アーキテクチャ、デザイン、および実装のサービスを提供します。
Splunk エンジニア	Splunk のライフサイクル全体を管理します。
マネージドサービスプロバイダ	Splunk を顧客向けのサービスとしてデプロイして実行するエンティティです。

Splunk Validated Architecture の選択プロセスの概要

Splunk Validated Architecture の選択プロセスにより、組織のニーズをすべて満たす最もシンプルで合理化されたアーキテクチャを見つけることができます。

選択プロセスのステップ	目標	考慮事項
ステップ1：以下の要件を定義する a) インデックス作成とサーチ b) データ収集メカニズム（1 つまたは複数）	要件の定義	• 意志決定者、関係者、および管理者が協力しあって、組織の要件を特定して定義します。 • すでにデプロイを実施している場合は、現在のアーキテクチャを評価して、検証済みモデルへと移行するために何が必要かを決定します。 要件を定義するための質問票については、下記のステップ 1 の説明を参照してください。
ステップ2：以下のトポロジーを定義する a) インデックス作成とサーチ b) 各データ収集メカニズム	特定した要件を満たすトポロジーの選択	• 要件を満たす最適なトポロジーを選択します。 • シンプルにすることを心がけ、SVA に準拠することで、将来の拡張が容易になります。 トポロジーオプションのダイアグラムと説明については、下記のステップ 2 の説明を参照してください。
ステップ3：設計方針とベストプラクティスを適用する	設計方針の優先順位付けと、層特有の実装ベストプラクティスのレビュー	• それぞれの設計方針は、SVA の 1 つまたは複数の柱を補強します。 • 組織のニーズに従って設計方針に優先順位を付けます。 • 層特有の推奨事項は、トポロジー実装のガイドとなります。 設計方針の詳細については、下記のステップ 3 の説明を参照してください。

ステップ 1a:インデックス作成とサーチの要件の定義

適切なデプロイトポロジーを選択するためには、要件を詳細に吟味する必要があります。要件を定義したら、Splunk をデプロイするのに最もシンプルで最もコスト効果の高い手段を選択します。もう少し先に、デプロイのインデックス層とサーチの層の主な要件を定義するための質問票があります。

要件質問票では、デプロイトポロジーに直接影響するエリアに主眼を置いています。そのため、次のステップでトポロジーを選択する前に、以下の質問に答えておくことをお勧めします。

考慮すべき事柄

使用事例のレビュー

要件を定義する際には、Splunk インフラストラクチャの使用事例を検討してください。たとえば、部門ごとの DevOps のトポロジーの使用事例は、通常は（例外もありますが）ミッションクリティカルな使用事例よりもシンプルです。以下について使用事例を熟考してください。

- サーチ
- 可用性
- コンプライアンス要件（常に 100% のデータ信頼度と可用性が必要である場合は特に重要です）
- 組織に特有の他の使用事例シナリオ

使用事例シナリオによっては、デプロイ側で追加のアーキテクチャ特性を設定する必要があります。

将来の成長の考慮

要件を定義するには、今直面しているニーズを検討する必要がありますが、将来の成長や拡張性についても考える必要があります。デプロイの拡張には、コストや増員など、今から計画しておくべきリソースがたくさん関与します。

トポロジーのカテゴリ

SVA の主要なトポロジーカテゴリを下表に示します。これらのカテゴリは、下記の質問票で使用します。以降の SVA 選択プロセスでも、これらのカテゴリを参照します。

インデックス層カテゴリ

カテゴリコード	説明
S	カテゴリ「S」は、単一サーバー Splunk デプロイのインデクサーを示します。
D	カテゴリ「D」は、 最低 2 台のインデクサーで構成される分散インデクサー層が必要なことを示します。
C	カテゴリ「C」は、クラスタ構成のインデクサー層（データの複製）が必要であることを示します。
M	カテゴリ「M」は、複数サイトのクラスタ構成のインデクサー層が必要であることを示します

サーチ作成層カテゴリ

カテゴリコード	説明
1	カテゴリ「1」は、単一のサーチヘッドで要件を満たすことを示します。
2	カテゴリ「2」は、要件を満たすには複数のサーチヘッドが必要であることを示します。
3	カテゴリ「3」は、要件を満たすにはサーチヘッドクラスタが必要であることを示します。
4	カテゴリ「4」は、要件を満たすには複数のサイトにまたがるサーチヘッドクラスタ（「ストレッチ」サーチヘッドクラスタ）が必要であることを示します。
+10	カテゴリ「+10」は、Enterprise Security App のサポートには専用のサーチヘッド（クラスタ）が必要であることを示します。サーチ層トポロジーカテゴリに「10」を追加して、この App 特有の要件に対応するトポロジーの説明を注意深く読んでください。

質問票 1: インデックス層とサーチ層の要件の定義

♦ トポロジーカテゴリコードについては上記の表を参照してください。複数の質問に対する答えが「はい」である場合は、番号が最も大きい質問のトポロジーカテゴリコードを使用します。

#	質問	考慮事項	トポロジーへの影響	インデクサー層のトポロジーカテゴリ ♦	サーチ層のトポロジーカテゴリ ♦
1	予想される毎日の取り込みデータ量は 300GB/日未満ですか?	毎日の取り込み量の短期的な（6 ～ 12 か月程度の）増大を検討してください。	可用性関連の質問への答えによっては単一サーバーデプロイを利用できます。	S	1
2	データ収集/インデックス作成において高可用性が必要ですか?	データを継続的に取り込む必要のある監視使用事例に Splunk を使用するのでなければ、ログデータが失われないという前提で取り込みデータの一時的な中断が許容されます。	継続的な取り込みをサポートするために分散デプロイが必要です。	D	1
3	サーチを実行するサーチヘッドがある場合: データは常にサーチ可	たとえば、使用事例がパフォーマンスメトリックスの計算と集計機	複製データ保持数 が 2 以上のクラスタ構成のインデクサーが必要	C	1

#	質問	考慮事項	トポロジーへの影響	インデクサー層のトポロジーカテゴリ ◆	サーチ層のトポロジーカテゴリ ◆
	能である必要がありますか（サーチ結果が常に完璧でなければなりませんか）？	能を使用した一般的な使用状況の監視であれば、インデクサーが 1 台停止しても大量のイベントの統計計算にはそれほど影響しません。 一方、使用事例がセキュリティの監査と脅威の検出であれば、サーチ結果にブラインドスポットがあることは許容できません。	です。注意: 複製データ保持数 が 2 でもインデクサーノード 1 台の障害には対応できますが、推奨される複製データ保持数は（デフォルトの）3 です。		
4	複数のデータセンターを稼動させて、データセンターの停止時には Splunk 環境を自動的に復旧する必要がありますか?	障害復旧要件を満たすには、2 つの施設の継続的な運用（アクティブ/アクティブ）を確保するか、または手動による障害復旧の RTO/RPO 目標を定める必要があります。	継続的な動作を実現するためには、複数サイトインデクサークラスタリングと最低 2 台のサーチヘッドによってデータ取り込み/インデックス層とサーチ層の両方でフェールオーバーを確保しなければなりません。	M	2
5	継続的なロスレスデータ取り込みを前提として、ユーザーが直接触れるサーチ層で高可用性が必要ですか?	Splunk を継続的なリアルタイム監視に使用する場合は、サーチ層の中断は許容されません。他の使用事例では、許容される場合と許	冗長サーチヘッド（おそらくはサーチヘッドクラスタリング）が必要です。	D/C/M	3

#	質問	考慮事項	トポロジーへの影響	インデクサー層のトポロジーカテゴリ ◆	サーチ層のトポロジーカテゴリ ◆
		容されない場合があります。			
6	多くのユーザーを同時にサポートする、あるいは膨大なスケジュール済みサーチワークロードをサポートする必要がありますか?	50 人までの同時ユーザー/50 件までの同時サーチをサポートするには、サーチ層の水平スケーリングが必要です。	サーチ層でサーチヘッドクラスタを使用するトポロジーが必要な場合があります。	D/C/M	3
7	マルチデータセンター環境で、ユーザーアーチファクト（サーチ、ダッシュボードなどのナレッジオブジェクト）をサイト間で同期させる必要はありますか?	これによって、サイト停止時でもデータの最新性と一貫性を保つことができます。	サイト間で適切に設定された「ストレッチ」サーチヘッドクラスタが必要です。**重要**:「ストレッチ」SHC によってサイト全体の障害が発生している間もサーチの可用性が高まりますが、常に両方のサイトですべてのアーチファクトが複製されるかどうかは保証されません。このため、Splunk App for Enterprise Security のように一貫した最新のアーチファクトに依存する App で影響が出る場合があります。サーチヘッドクラスタリングだけでは、完全なデ	M	4

#	質問	考慮事項	トポロジーへの影響	インデクサー層のトポロジーカテゴリ ◆	サーチ層のトポロジーカテゴリ ◆
			ータ復旧ソリューションは実現できません。サーチヘッドクラスタリングの他の利点は活用できます。		
8	Splunk App for Enterprise Security (ES) をデプロイする予定はありますか?	各トポロジーにおける Splunk App for Enterprise Security の制限事項をよく読んで理解してください。	ES は専用のサーチヘッド環境（スタンドアロンまたはクラスタ）を必要とします。	D/C/M	+10
9	データ保全規制の対象となる地域に分散環境が設置されていますか?	一部の国の規制では、国内で生成されたデータを国外にあるシステムに持ち出すことを禁止しています。	このような規制がある場合は、中央に Splunk インデックス層を配置することができなくなり、Splunk やパートナーとユーザーとの連携によってこのようなデプロイの詳細を検討して、カスタムアーキテクチャを開発する必要があります。つまり、この要件を満たす SVA はありません。	カスタム	カスタム
10	共有サーバー/インデクサーへの特定のログデータソースのコロケーションを禁止するような厳しいセキュリ	企業の方針により、非常に重要なログデータは、リスクの低いデータセットと同じ物理システムや同じネッ	複数の独立した（おそらくは共有ハイブリッドサーチ層を持つ）インデックス作成環境が必要です。これは	カスタム	カスタム

#	質問	考慮事項	トポロジーへの影響	インデクサー層のトポロジーカテゴリ ♦	サーチ層のトポロジーカテゴリ ♦
	ティ方針がありますか?	トワークゾーンへのコロケーションが禁止されている場合があります。	SVA の範囲を超えており、カスタムアーキテクチャ開発が必要です。		

トポロジーカテゴリコードの決め方

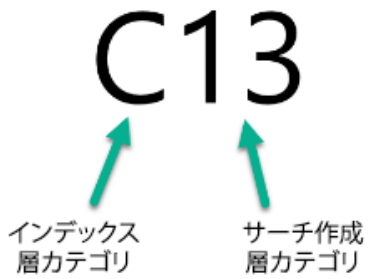
上記の要件質問票への回答から、ニーズに最適なトポロジーを示すトポロジーカテゴリインジケーターが決まります。以下に、トポロジカテゴリコードを決める手順と例を示します。

手順

1. 「はい」と回答した質問の番号を書き留めます。
2. 複数の質問に「はい」と答えた場合は、番号が最も大きい質問を選びます。複数のトポロジーオプション（例: D/C/M）がある場合は、その前の質問から最適なトポロジーを選びます。
3. トポロジーカテゴリコードの最初はインデクサー層を表す文字（C または M）です。その後に、サーチ層を表す数字（1 または 13）が続きます。

例 1

質問 3、5、および 8 に「はい」と回答したとします。この場合のトポロジーカテゴリコードは C13 となり、2 台のサーチヘッドクラスタを持つクラスタ化インデックス層を示します。



例 2

質問 1 への回答のみが「はい」だとします。この場合のトポロジーカテゴリコードは S1 となり、単一サーバー Splunk デプロイが理想的なトポロジーとなります。

ステップ 2a: インデックス作成とサーチのトポロジーの選択

トポロジーは、非クラスタ化デプロイとクラスタ化デプロイに大別されます。非クラスタ化デプロイでは、独立したコンポーネントの数を最小限に抑えることができ、拡張性にも優れています。非クラスタ化デプロイは可用性と障害復旧性が低下しますが、組織にとっては非常に優れた選択肢となり得ます。

メモ: SVA 選択プロセスの最大の目標は、不要なコンポーネントを導入することなく必要なシステムを構築することです。

注意

現在のニーズを満足する以上の利点を提供するトポロジーを実装することもできますが、不要なコストが発生する可能性が高いことを忘れないようにしてください。さらに、複雑さが増大するため、運用効率を考えると逆効果になることがあります。

トポロジーダイアグラムに関する重要な注意

トポロジーダイアグラムのアイコンは、**Splunk の機能ロール**を表しており、これらを実行するための専用インフラストラクチャを意味するものではありません。同じインフラストラクチャ/サーバーにコロケーションできる Splunk ロールについては付録を参照してください。

トポロジーカテゴリコードの使い方

トポロジーオプションを選択する前に、要件質問票への回答を済ませてトポロジーカテゴリコードを決めておくことを強くお勧めします。まだ済んでいない場合は、前のセクションに戻って手順を完了してください。トポロジーカテゴリコードが決まれば、要件に最適なデプロイオプションを特定できます。

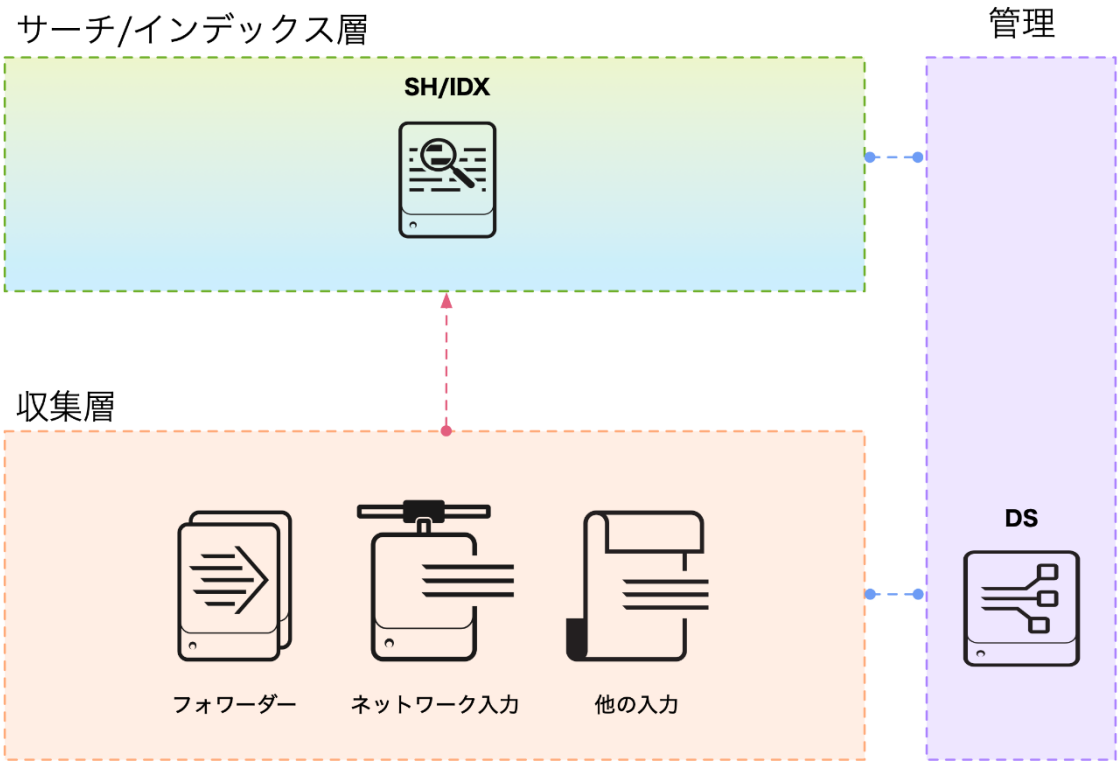
非クラスタ化デプロイオプション

以下のトポロジーオプションがあります。

デプロイ種別	トポロジーカテゴリコード
単一サーバーデプロイ	S1
分散非クラスタ化デプロイ	D1/D11

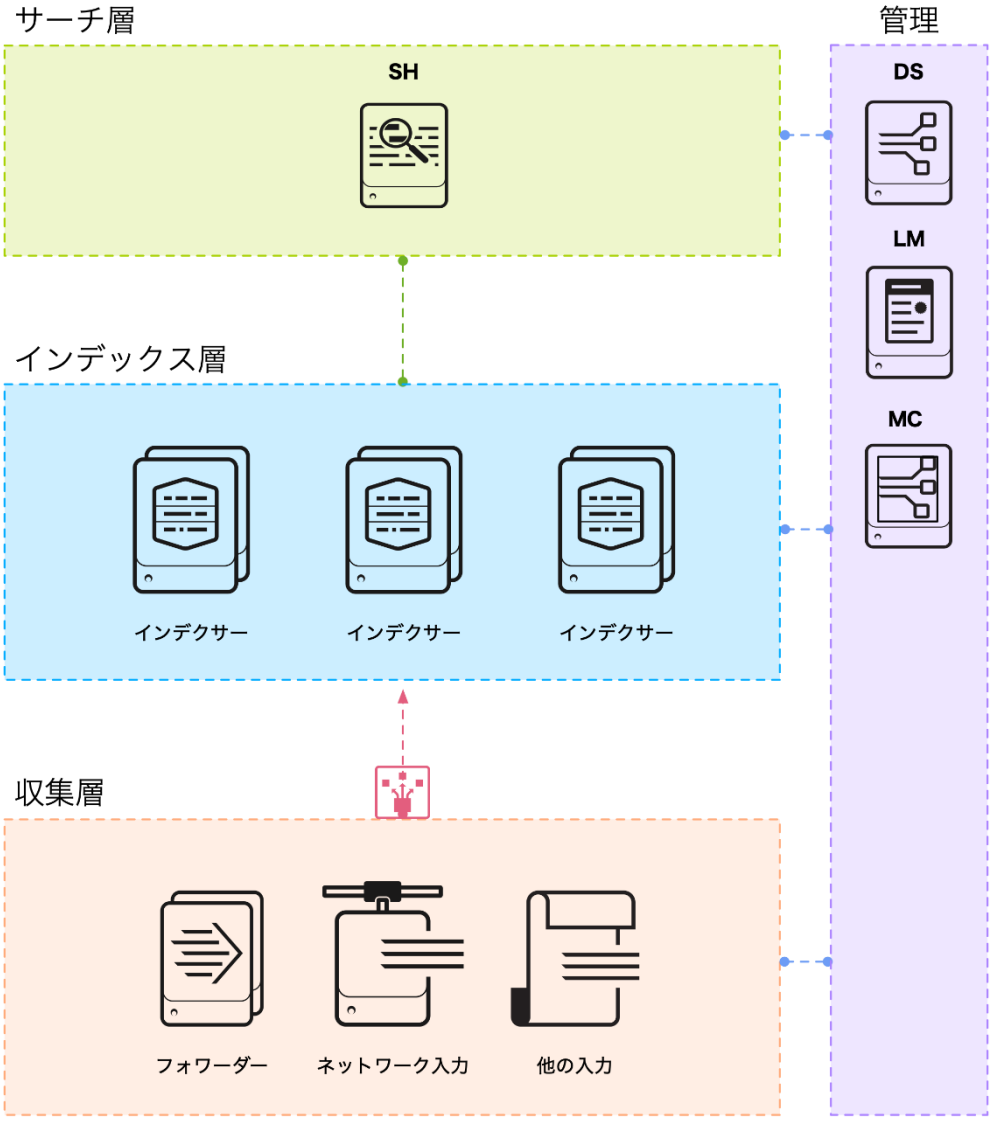
トポロジーコンポーネントの説明は付録 B を参照してください。

単一サーバーデプロイ（S1）



単一サーバーデプロイ（S1）の説明	制限
このデプロイトポロジーは、a）Splunk デプロイに高可用性や自動災害復旧を求めない場合、b）取り込まれるデータ量が 300GB/日未満である場合、そして c）ユーザー数が少なく、サーチ使用事例がクリティカルではない場合に、非常にコスト効果に優れたソリューションを提供します。 通常、このトポロジーはビジネスクリティカルではない（本質的に部門ごとの）使用事例で使用されます。適切な使用事例としては、データオンボードテスト環境、小規模の DevOps 使用事例、App テストおよび統合環境などのシナリオがあります。 このトポロジーの主な利点としては、管理が容易なこと、小規模のデータボリュームではサーチパフォーマンスが高いこと、そして TCO が変動しないことなどがあります。	• サーチ/インデックス作成の高可用性は提供されません。 • 拡張性がハードウェア容量によって制限されます（分散デプロイへの移行によって解消されます）。

分散非クラスタ化デプロイ（D1/D11）



分散非クラスタ化デプロイ（D1/D11）の説明	制限
a）Splunk に取り込まれる毎日のデータ量が単一サーバーデプロイの容量を超えている場合、または b）データの取り込みで高可用性を保証したい/する必要がある場合は、分散トポロジーへの移行が必要になります。複数の独立したインデクサーをデプロイすることで、インデックス作成の容量が単純比例で増加し、データ取り込みの可用性も高まります。 インデクサーノードを追加すると、TCO も（予測可能で）単純比例で増大します。モニタリングコンソール（MC）コンポーネント（推奨）を導入すると、分散デプロイの健全性と容量を監視できます。さらに、MC によって一元化されたアラートシステムが実現し、デプロイの健全性が低下した場合には通知を受け取ることができます。 新しいインデクサーを追加するたびに、利用可能なサーチピアでサーチヘッド（1 台または複数台）を手動で設定する必要があります。**ES ユーザーへの注意**：カテゴリコードが D1 である（Splunk App for Enterprise Security をデプロイする）場合は、**単一の専用**サーチヘ	• サーチ層の高可用性は提供されません。 • インデックス層の高可用性は制限され、ノード障害時には履歴サーチ結果が不完全になることがあります。

分散非クラスタ化デプロイ (D1/D11) の説明	制限
ッドを使用して App をデプロイする必要があります（トポロジーダイアグラムには示されていません)。 新しいインデクサーを追加するたびに、収集層を（デプロイサーバー経由で）ターゲットインデクサーのリストで設定する必要があります。 このトポロジートポロジは、インデクサーノード 1000 台まで単純比例で拡張できるため、膨大な量のデータ取り込みとサーチをサポートできます。 大規模なデータセットでも、多くのインデクサーでの並列サーチ実行（マップ/低減）によってサーチのパフォーマンスを維持できます。 独立したトポロジーとしては分類されていませんが、サーチヘッドクラスタを使用してサーチ層のサーチ容量を増やすことができます（トポロジー C3/C13 のサーチ層参照)。	

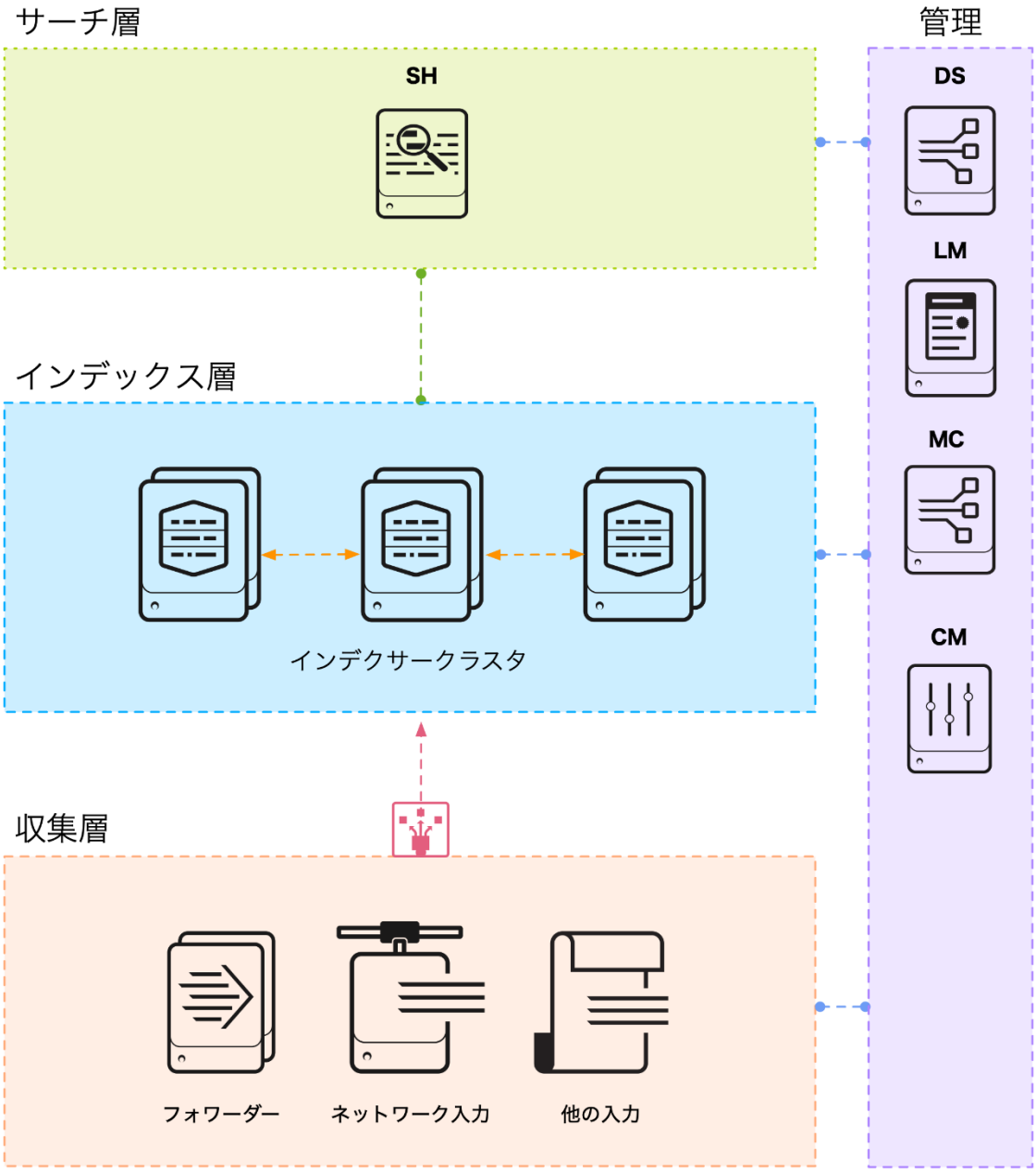
クラスタ化デプロイオプション

以下のトポロジーオプションがあります。

デプロイ種別	トポロジーカテゴリコード
分散クラスタ化デプロイ - 単一サイト	C1/C11
分散クラスタ化デプロイ + SHC - 単一サイト	C3/C13
分散クラスタ化デプロイ - 複数サイト	M2/M12
分散クラスタ化デプロイ + SHC - 複数サイト	M3/M13
分散クラスタ化デプロイ + SHC - 複数サイト	M4/M14

トポロジーコンポーネントの説明は付録 B を参照してください。

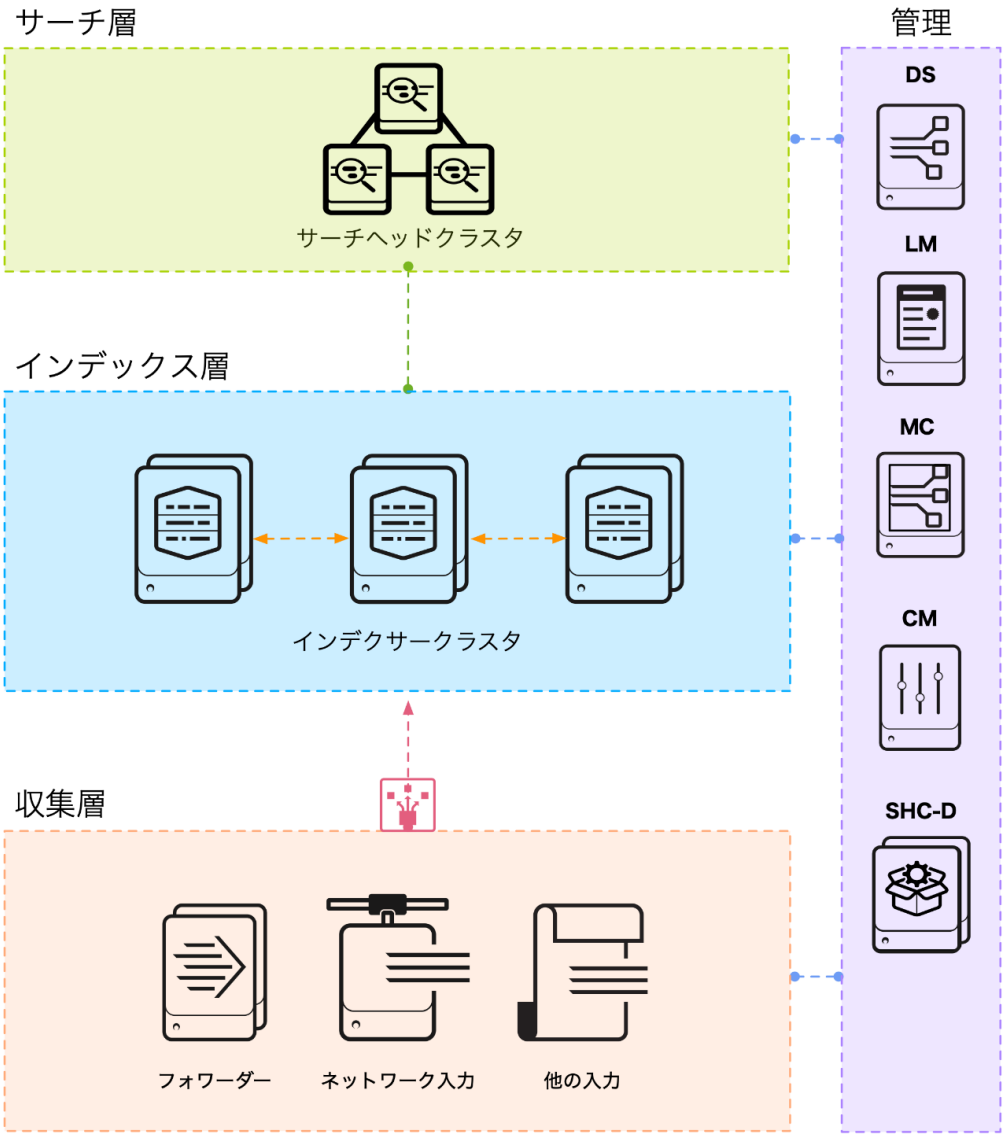
分散クラスタ化デプロイ - 単一サイト (C1/C11)



分散クラスタ化デプロイ - 単一サイト (C1/C11) の説明	制限
このトポロジーは、 インデクサークラスタリングと、適切に設定されたデータ複製ポリシーを併せて実現します。これにより、インデクサーピアノードの障害時でもデータの高可用性が保証されます。ただし、この高可用性はインデックス層にのみ適用され、サーチヘッド障害に対してはデータは保護されません。 **ES ユーザーへの注意**: カテゴリコードが C11 である (Splunk App for Enterprise Security をデプロイする) 場合は、**単一の専用**サーチヘッドを使用して App をデプロイする必要があります (トポロジーダイアグラムには示されていません)。 このトポロジーでは、クラスタマスタ (CM) という追加の Splunk コンポーネントが必要です。CM は、設定したデータ複	• サーチ層の高可用性は提供されません。 • インデクサークラスタの一意の総バケツ数は 500 万 (V6.6+)、総バケツ数は 1500 万に制限されます。 • データセンター停止時には自動障害復旧は利用できません。

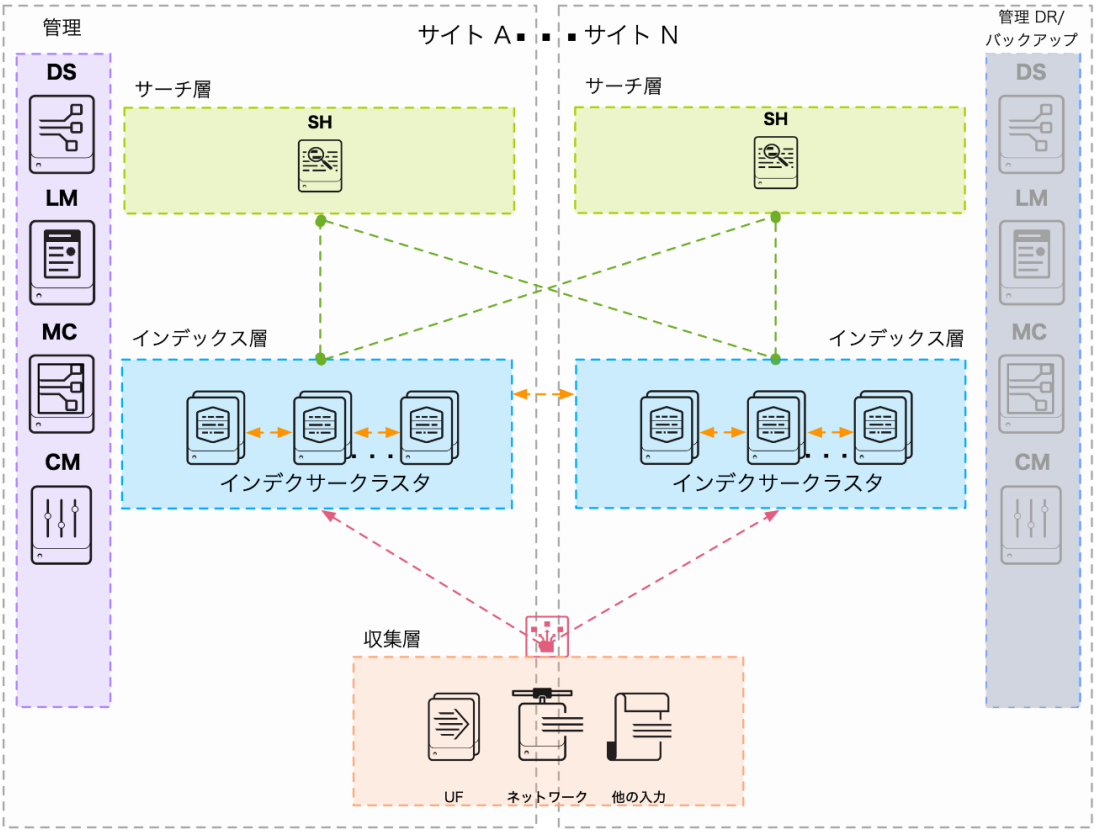
分散クラスタ化デプロイ - 単一サイト (C1/C11) の説明	制限
製ポリシーの調整と実施を担当します。また、CM は、クラスタピア（インデクサー）の信頼できるソースにもなります。各サーチピアではなく CM を設定することで、サーチヘッドの設定を簡素化することができます。 利用可能なインデクサーを CM 経由で検出するように転送層を設定することもできます。これによって転送層の管理が容易になります。 データはクラスタ内で非決定論的に複製されるようにします。要求した各イベントのコピーがどこに格納されるかを制御することはできません。さらに、拡張は単純比例となるため、合計クラスタサイズには制限があります（理想的な条件下でのサーチ可能データは 50PB まで）。 モニタリングコンソール (MC) をデプロイして Splunk 環境の健全性を監視することをお勧めします。	

分散クラスタ化デプロイ + SHC - 単一サイト (C3/C13)



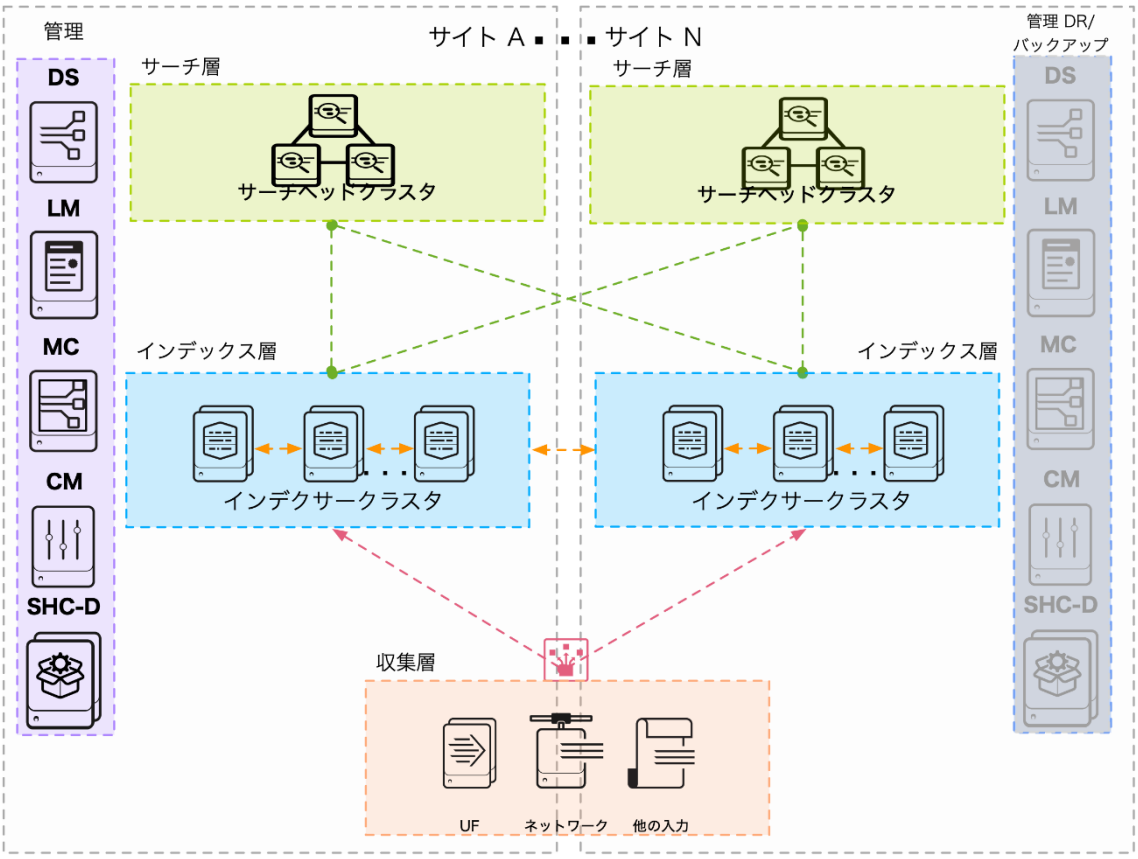
分散クラスタ化デプロイ + SHC – 単一サイト (C3/C13) の説明	制限
このトポロジーは、水平の拡張性を実現し、サーチ層から単一障害点を削除します。SHC の実装には 3 台以上のサーチヘッドが必要です。 SHC 設定を管理するために、さらにもう 1 つのSplunk コンポーネントであるサーチヘッドクラスタデプロイヤが各 SHC に必要です。このコンポーネントは、クラスタ内の設定ファイルに変更をデプロイするのに必要です。サーチヘッドクラスタデプロイヤには HA 要件はありません（実行時のロールなし）。 SHC は、単一のサーチヘッドだけで提供できるサーチ容量を超えて、利用可能なサーチ容量を増やします。また、クラスタ間でスケジュール済みサーチワークロードを分散させます。さらに、サーチヘッド障害時には最適なユーザーフェールオーバーを実現します。 クラスタ間での適切なユーザー負荷の分散を保証するためには、継続的セッションをサポートするネットワークロードバランサーを SHC メンバーの前に配置する必要があります。 **ES ユーザーへの注意**: カテゴリコードが C13 である（Splunk App for Enterprise Security をデプロイする）場合は、**専用の**サーチヘッドクラスタを使用して App をデプロイする必要があります（トポロジーダイアグラムには示されていません）。サーチ層には、容量と組織のニーズに応じてクラスタ化サーチヘッドと非クラスタ化サーチヘッドを配置できます（これもトポロジーダイアグラムには示されていません）。	• データセンター停止時には障害復旧は利用できません。 • ES では SH/SHC が必要です。 • SHC 上での ES デプロイの管理はサポートされていますが容易ではありません（PS が必要）。 • SHC のノード数は 100 台までです。

分散クラスタ化デプロイ – 複数サイト (M2/M12)



分散クラスタ化デプロイ - 複数サイト (M2/M12) の説明	制限
致命的な障害（データセンターの停止など）の発生時に準自動障害復旧を実現するには複数サイトクラスタリングが最適なデプロイアーキテクチャです。複数サイトクラスタを健全に保つには、サイト間のネットワーク遅延をSplunk ドキュメントで指定されている許容範囲内に収める必要があります。 このトポロジーにより、2 つ以上のインデクサークラスタピアグループにデータを**決定論的に**複製します。また、単一サイト複製保持数とサーチ可能データ保持数も設定できます。サイトの複製データ保持数によって、複製するコピーの送り先を指定して、データが複数の場所に分散して格納されるようにすることができます。 管理は単一のクラスタマスターノードによって行い、障害の発生時には DR サイトにフェールオーバーします。 複数サイトクラスタリングにより、物理的に離れた分散ロケーション間でデータの冗長性を保証することができ、地理的に離れた場所でも分散が可能です。 自動的に DR サイトにフェールオーバーすることで、可用性を保証できます。ただし、このトポロジーには、サイト間でサーチ層の設定と実行時アーチファクトを自動的に同期させる仕組みはありません。 サイト間で利用可能なサーチピア（インデクサー）容量は、アクティブ/アクティブモデルでサーチの実行に利用できます。可能であれば、特定のサイトのサーチヘッドにログオンしたユーザーがローカルインデクサーのみをサーチするようにサイトアフィニティを設定することができます。 **ES ユーザーへの注意**: カテゴリコードが M12 である (Splunk App for Enterprise Security をデプロイする) 場合は、**単一の専用**サーチヘッドを使用して App をデプロイする必要があります（トポロジーダイアグラムには示されていません)。ES サーチヘッドでフェールオーバーを実現するには、DR 時にのみアクティブになって使用される「シャドー」サーチヘッドをフェールオーバーサイトで設定します。Enterprise Security デプロイでフェールオーバーメカニズムを設計して実装する場合は、Splunk プロフェッショナルサービスまでご相談ください。	• **サイト間**で、利用可能なサーチヘッド容量の共有とサーチアーチファクトの複製はできません。 • サイト障害時の管理機能の障害には Splunk 外で対処しなければなりません。 • インデックス複製のサイト間遅延は推奨制限値以内でなければなりません。

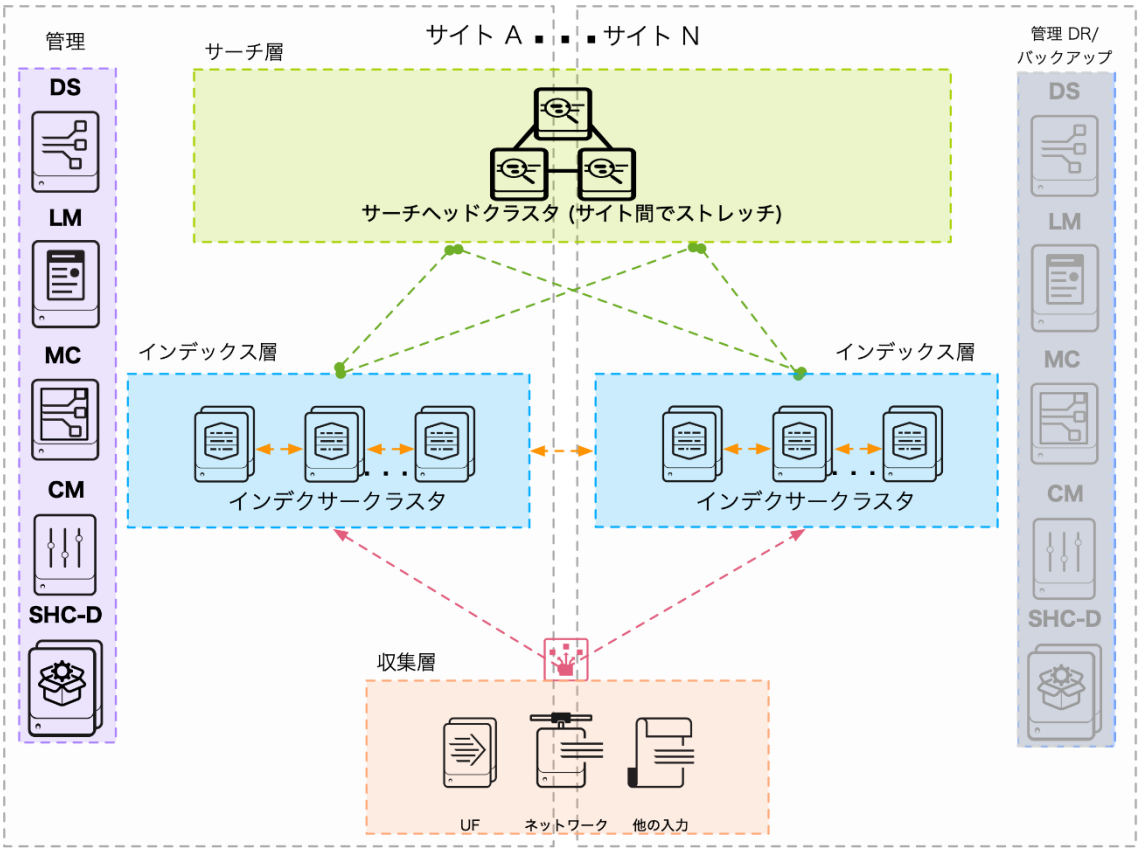
分散クラスタ化デプロイ + SHC - 複数サイト (M3/M13)



分散クラスタ化デプロイ + SHC - 複数サイト (M3/M13) の説明	制限
このトポロジーは、水平の拡張性を実現し、各サイトのサーチ層から単一障害点を削除します。SHC の実装には（サイトごとに）3 台以上のサーチヘッドが必要です。 SHC 設定を管理するために、さらにもう 1 つのSplunk コンポーネントであるサーチヘッドクラスタデプロイヤが各 SHC に必要です。このコンポーネントは、クラスタ内の設定ファイルに変更をデプロイするのに必要です。サーチヘッドクラスタデプロイヤには HA 要件はありません（実行時のロールなし）。 SHC には、a）単一のサーチヘッドだけで提供できるサーチ容量を超えて、利用可能なサーチ容量を増やし、b）クラスタ間でスケジュール済みサーチワークロードを分散させ、c）サーチヘッド障害時には最適なユーザーフェールオーバーを実現するという利点があります。 クラスタ間での適切なユーザー負荷の分散を行うには、継続的セッションをサポートするネットワークロードバランサーを各サイトの SHC メンバーの前に配置する必要があります。 **ES ユーザーへの注意**: カテゴリコードが M13 である（Splunk App for Enterprise Security をデプロイする）場合は、各サイトで単一の専用サーチヘッドクラスタを使用して App をデプロイする必要があります（トポロジーダイアグラムには示されていません）。サイト障害から ES SH 環境を復旧できるようにするには、サードパーティの製品を利用して、サーチヘッドインスタンスのフェールオーバ	• サイト間でサーチアーチファクトは複製されません。SHC はそれぞれ独立しています。 • インデックス複製のサイト間遅延はドキュメントに記載されている制限値以内でなければなりません。 • SHC のノード数は 100 台までです。

分散クラスタ化デプロイ + SHC - 複数サイト (M3/M13) の説明	制限
ーを実行するか、または「ウォームスタンバイ」状態の ES SH をプロビジョニングして主 ES 環境と同期させておきます。HA/DR 環境で ES をデプロイする場合は、Splunk プロフェッショナルサービスにご相談いただくことを強くお勧めします。	

分散クラスタ化デプロイ + SHC - 複数サイト (M4/M14)



分散クラスタ化デプロイ + SHC - 複数サイト (M4/M14) の説明	制限
最も複雑な SVA であり、高可用性と障害復旧の要件が最も厳しいデプロイ向けに設計されています。適切なデプロイを行うために、Splunk プロフェッショナルサービスへのご相談を強くお勧めします。このトポロジーを適切にデプロイすると、データ収集、インデックス作成、そしてサーチを実行する Splunk インフラストラクチャの継続的な動作が確保されます。 このトポロジーでは、1 つまたは複数のサイトにまたがる「ストレッチ」サーチヘッドクラスタを実装します。これにより、サーチノードまたはデータセンターの障害時に、最適なフェールオーバーが行われます。サーチアーチファクトや他の実行時ナレッジオブジェクトは、SHC で複製されます。SHC 自身はサイトを認識しないため（アーチファクトの複製は非決定論的に行われます)、複製がサイト間で行われるように設定を慎重に行う必要があります。	• サイト間のネットワーク遅延は、ドキュメントに記載されている制限値以内でなければなりません。 • クラスタメンバーの過半数で障害が発生した場合は、SHC のフェールオーバーを手動で行わなければならないこともあります。

分散クラスタ化デプロイ + SHC - 複数サイト (M4/M14) の説明	制限
ローカルサーチで結果が得られなかった場合にのみサイト間の WAN リンクを使用するようにサイトアフィニティを設定できます。 クラスタ間での適切なユーザー負荷の分散を保証するためには、継続的セッションをサポートするネットワークロードバランサーを SHC メンバーの前に配置する必要があります。 **ES ユーザーへの注意**: カテゴリコードが M14 である (Splunk App for Enterprise Security をデプロイする) 場合は、各サイトで単一の専用サーチヘッドクラスタを使用して App をデプロイする必要があります (トポロジーダイアグラムには示されていません)。ES では、実行時アーチファクトの一貫したセットが利用可能であることが必要ですが、サイトが停止した場合、これは「ストレッチ」SHC では保証されません。サイト障害から ES SH 環境を復旧できるようにするには、サードパーティの製品を利用して、サーチヘッドインスタンスのフェールオーバーを実行するか、または「ウォームスタンバイ」状態の ES SH をプロビジョニングして主 ES 環境と同期させておきます。HA/DR 環境で ES をデプロイする場合は、Splunk プロフェッショナルサービスにご相談いただくことを強くお勧めします。	

ステップ 1b:データ収集要件の定義

データ収集層は、Splunk デプロイの中核となるコンポーネントです。インデックス層にデータを転送し、Splunk でサーチできるようにします。ここで最も重要な要因は、転送とインデックス作成が最も効率的で信頼性の高い方法で行われるようにすることです。これが、Splunk デプロイの成功と高いパフォーマンスへの鍵を握っています。

データ収集層のアーキテクチャについては、以下の側面を考慮してください。

- データの発生元。ログファイル、syslog ソース、ネットワーク入力、OS イベント記録装置、App、メッセージバスなど。
- データ取り込みの遅延とスループットの要件
- インデックス層のインデクサー間での理想的なイベント分散
- 障害対策と自動復旧 (HA)
- セキュリティとデータ管理の要件

SVA のこのセクションでは、一般的なデータ収集方式に主眼を置いて、各データ収集方式のアーキテクチャとベストプラクティス、そして実装方式を選択する際に考慮すべき潜在的な問題について説明します。

重要なアーキテクチャ上の考慮事項とそれらが重要である理由

データ収集層の役割は重要であるため、アーキテクチャの設計に関係する主要な考慮事項について理解しておくことが大切です。

これらの考慮事項の中には自分の要件には関係のないものもありますが、太字で示されている事項はすべての環境で重要となります。

考慮事項	重要である理由
データが適切に（タイムスタンプ、改行、切り詰めなど）取り込まれること	インデクサー間での理想的なイベント分散は極めて重要です。インデックス層は、利用可能なすべてのインデクサーが均等に利用されている場合に最も効率的に動作します。これはデータの取り込みでもサーチでも同じです。ピアではなく単一のインデクサーで非常に多くのデータ取り込みを処理すると、サーチの応答時間が長くなることがあります。また、ローカルディスクのストレージ容量が少ないインデクサーの場合、イベントが均等に分散されないと、設定されているデータ保存ポリシーの条件が成立する前にストレージが満杯になってしまうことがあります。
利用可能なインデクサー間でデータが最適に分散されること	イベントのタイムスタンプと改行が正しく設定されていないため、データが適切に取り込まれない場合、このデータのサーチは非常に困難になります。その理由は、サーチ時にイベント境界を定めなければならないためです。タイムスタンプ抽出の設定が間違っている、あるいは設定そのものがない場合、暗黙的なタイムスタンプ割り当てが不必要に行われることがあります。この結果、ユーザーが混乱してしまい、データから価値を引き出すことが必要以上に難しくなってしまいます。
すべてのデータが失われることなく高い信頼性でインデックス層まで到達すること	信頼性の高い分析を行うために収集するログデータは、そのデータに対して実行されるサーチが有効で正確な結果を返すように、完全かつ有効でなければなりません。
すべてのデータがインデックス層に最短の遅延で到達すること	データの取り込みで遅延が発生すると、潜在的に重要なイベントが発生してから、そのイベントをサーチして対処できるようになるまでの時間が長くなってしまいます。取り込み遅延を最小限に抑えることは、スタッフにアラートを送ったり自動アクションを実行したりする監視使用事例においては重要になります。
データが安全に転送されること	重要なデータであったり、データが信頼性の低いネットワークを通過するために保護が必要であったりする場合は、第三者によるデータの盗み見を防止するための暗号化が必要です。一般的には、Splunk コンポーネント間のすべての接続では SSL を有効にすることをお勧めします。
ネットワークリソースの使用が最小限に抑えられること	他のビジネスクリティカルなネットワークトラフィックに影響しないように、ログデータの収集によるネットワークリソースへの影響は最小限に抑える必要があります。専用線のネットワークであれば、ネットワーク利用を最小限に抑えることで、デプロイの TCO を抑えることができます。
データソースを認証すること	不正なデータソースによるインデックス作成環境への影響を防止するため、接続に認証を導入することを検討してください。ネットワークコントロールを使用するか、または App レベルのメカニズム（SSL/TLS など）を使用することで対応できます。

デプロイの役割は重要であるため、本書のガイダンスは理想的なイベント分散をサポートするアーキテクチャに主眼を置いています。Splunk 環境で期待するサーチパフォーマンスが得られない場合、ほとんどのケースでは、最低限のストレージパフォーマンス要件を満足していないか、またはイベントの分散が均等ではないためサーチの並列実行が活用されていないことが原因です。

最も重要なアーキテクチャ上の考慮事項について理解したところで、満足すべき特定のデータ収集要件について見ていきましょう。

質問票 2: データ収集要件の定義

下記の質問に答えることで、デプロイで必要なデータ収集コンポーネントを特定することができます。右端の列にあるキーから、各コンポーネントの詳細説明を読んでください。

#	質問	考慮事項	トポロジーへの影響	関連するデータ収集コンポーネント
1	ローカルファイルを監視したり、エンドポイントでデータ収集スクリプトを実行したりする必要はありますか?	これは、ほぼすべてのSplunk デプロイシナリオでの中核的な要件となります。	エンドポイントにユニバーサルフォワーダーをインストールして、設定を一元管理する必要があります。	UF
2	ソフトウェアをインストールできないデバイス（各種機器、ネットワークスイッチなど）からログデータを syslog 経由で収集する必要はありますか?	syslog は、カスタムソフトウェアをインストールできない専用デバイスで多く使用されている伝送プロトコルです。	収集ポイントとして機能する syslog サーバーインフラストラクチャが必要です。	SYSLOG HEC
3	ローカルディスクではなく API にログを書き込む App からのログデータ収集をサポートする必要はありますか？	エンドポイントのログファイルへの書き込みを行う場合は、ディスクスペースの確保とログファイルの管理（ローテーション、削除など）が必要です。ユーザーによってはこのモデルではなく、用意されているロギングライブラリを使用して Splunk に直接ログを書き込む方式を好みます。	Splunk HTTP イベントコレクター（HEC）や、ログシンクとして機能する他の技術が必要です。	HEC
4	ストリーミングイベントデータプロバイダーからデータを収集する必要はありますか?	多くの企業では、一元化されたストリーミングデータプラットフォーム（AWS Kinesis や Kafka など）を介してログデータのプロデューサーとコンシューマーの間でメッセージを転送するイベントハブモデルを採用しています。	この場合は、ストリーミングデータプロバイダーと Splunk との統合が必要です。	KAFKA KINESIS HEC
5	ログプロデューサーがインデックス層と直接 TCP 接続を確立することを禁止している（譲歩できない）セキュリ	ネットワークトポロジーが複数のネットワークゾーンで構成され、ゾーン間のファイアウォールルールの規制に	ネットワークゾーン間でトラフィックを中継する転送層が必要です。	IF

	ティポリシーはありますか?	よって Splunk ポートのトラフィックがゾーン間を移動できない場合があります。また、各ソースおよびターゲット IP アドレスに対してファイアウォールルールを設定して管理する作業も面倒になります。		
6	プログラマティックな手段（REST API 呼び出しやデータベースへのクエリなど）を使用してログデータを収集する必要はありますか?	Splunk には、リレーショナルデータベースからデータを収集するための DBX など、さまざまなデータ取り込み使用事例に合わせて、API に対してスクリプトを実行するための各種モジュール式入力が用意されています。	データ収集層では、Splunk ヘビーフォワーダーで実装された 1 つまたは複数のデータ収集ノード（DCN）が必要です。	DCN
7	Splunk 以外の（Splunk に加えて）他のシステムにデータ（の一部）を転送する必要はありますか?	一部の使用事例では、Splunk でインデックス作成されたデータを他のシステムに転送する必要があります。多くの場合、転送されたデータがソースデータの一部のみで構成されるか、あるいは転送前にデータの修正が必要です。	使用事例の詳細によっては、中継転送層をヘビーフォワーダーで構築して、イベントベースの転送とフィルタリングをサポートする必要があります。あるいは、Splunk App for CEF の cefout コマンドを使用してインデックス作成後のデータを転送できます。	HF
8	リモートサイトのネットワーク帯域幅が制約されているため、転送前に多くのデータをフィルターする必要がありますか?	転送前にデータをフィルターするには、構文解析用の（ヘビー）フォワーダーが必要です。HF が使用するアウトバウンドネットワーク帯域幅は、UF の約 5 倍になるため、大量のイベントがフィルターされる場合にのみフィルタリングが意味を持ちます（目安: ソースデータの 50% 超）。理想的な解決法は、ロギングの詳細レベルを調整することで	ソースでのログボリュームを減らすことができない場合は、リモートサイトに中継 HF を設けて、ソースデータの構文を解析し、設定に応じてイベントをフィルターしてください。	IF HF

		ログボリュームを減らすことです。		
9	インデックス作成のために公共ネットワークにデータを送信する前に、重要なデータをマスク/暗号化する必要がありますか?	フォワーダーのトラフィックを SSL で暗号化するだけでは、公共ネットワーク上に送信する重要なデータを保護するには十分ではない場合があるため、イベントの各部分を転送前にマスクする必要があります（SSN、CC データなど）。このようなデータのマスキングは、ログデータを生成する App で行うのが理想的です。	ログを生成する App でデータをマスクできない場合は、ローカルサイトに中継 HF を設けて、データをインデクサーに送信する前に、ソースデータの構文を解析して、設定に従って必要なマスキングルールを適用する必要があります。	IF HF
10	statsd または collectd を使用してメトリックスを取得する必要はありますか?	statsd と collectd は、ホストシステムや App からメトリックスを収集するのに多く利用されます。	Splunk は、UF、HF、または HEC を使用してこれらのインデックスを供給するために、特定のインデックスタイプや収集方式をサポートしています。	METRICS
11	いずれかのデータ収集コンポーネントで高可用性を実現する必要はありますか?	通常はエンドポイントには適用されませんが、中継フォワーダーやデータ収集ノードといったデータ収集コンポーネントでは高可用性が求められることがあります。	機能停止によって各コンポーネントの可用性がどのように影響されるか、そしてどのように対応すべきかについて考える必要があります。	HA

ステップ 2b:データ収集コンポーネントの選択

質問票を完成させると、デプロイの要件を満たすために必要なデータ収集コンポーネントが決まります。このセクションでは、各データ収集アーキテクチャコンポーネントについて詳しく説明します。最初に、いくつかの全体的なガイダンスを示します。

全般的な転送アーキテクチャのガイダンス

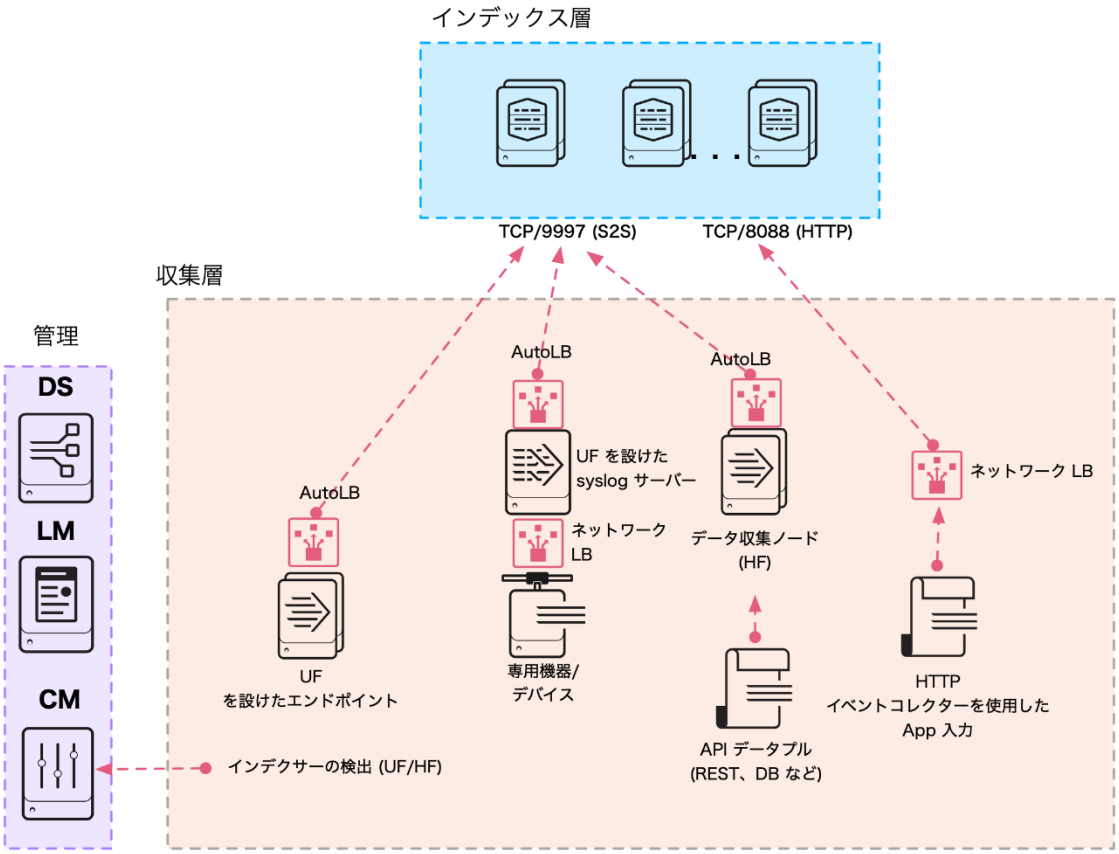
データ収集層は可能な限り「平坦」なのが理想的です。つまり、データソースがユニバーサルフォワーダーによってローカルに収集され、インデックス層に直接転送されるということです。これにより、データ取り込みの遅延（サーチに要する時間）が最小限に抑えられ、利用可能なインデクサー間でイベントが適切に分散されるため、ベストプラクティスであると言えます。このベストプラクティスに従うことにより、管理が容易になり、操作が簡素化されます。多くのユーザーは、中継転送層を設けていますが、この方法は他の方法で要件を満足できない場合にのみ使用してください。中継フォワーダーの潜在的な影響を考え、本書ではこのトピックについて独立したセクションを設けて説明します。

ユニバーサルフォワーダーをインストールできないため、syslog プロトコルによるログ収集ができないエンドポイントもあります（たとえばネットワークデバイスや各種機器）。このようなデータソースからデータを収集するためのベストプラクティスアーキテクチャについては、「syslog によるデータ収集」で説明します。

プログラマティックな手段（API、データベースアクセスなど）によって収集する必要のあるデータソースの場合は、Splunk Enterprise の完全版をベースとしたデータ収集ノード（DCN）のデプロイをお勧めします。これはヘビーフォワーダーとも呼ばれます。開発環境以外では、サーチヘッド層でこれらの種類の入力を実行することは推奨されません。

このガイダンスを反映させた一般的なデータ収集アーキテクチャを下図に示します。

データ収集トポロジーの概要



上の図では、データ収集コンポーネントの設定を管理するデプロイサーバー（DS）が管理層に示されています。また、データ収集ノードは Splunk Enterprise の機能を有効にするためにライセンスマスター（LM）にアクセスする必要があるため、LM も示されています。クラスタマスター（CM）がある場合は、フォワーダーがインデクサーを検出するために使用できるため、フォワーダー出力設定で利用可能なインデクサーを管理する必要がなくなります。

上の図では、AutoLB は Splunk に組み込まれている自動負荷分散メカニズムを表します。このメカニズムは、Splunk 固有の S2S プロトコル（デフォルトポート 9997）を使用して送信されるデータでイベントが適切に分散されることを保証します。注意: S2S トラフィックでの外部ネットワークロードバランサーの使用は現時点ではサポートされておらず、推奨もされません。

業界標準プロトコル（HTTP や syslog など）で通信を行うデータソースからのトラフィックを負荷分散するには、ネットワークロードバランサーを使用して、インデックス層のインデクサー間で負荷とイベントを均等に配分します。

(UF) ユニバーサルフォワーダー

ユニバーサルフォワーダー (UF) は、環境内のシステムから大量のデータを収集する場合には最適な選択肢です。UF は、リソース要求量を極めて少なく抑えたデータ収集専用メカニズムです。UF は、ログデータの収集と転送のデフォルトオプションです。UF は以下を提供します。

- チェックポイント/再起動機能によるロスレスデータ収集
- ネットワーク帯域幅の使用を最小限に抑える効率的なプロトコル
- 抑制機能
- 利用可能なインデックス間での組み込みロードバランシング
- SSL/TLS を使用したネットワーク暗号化 (オプション)
- データ圧縮 (SSL/TLS 非使用時のみ)
- 複数の入力方式 (ファイル、Windows イベントログ、ネットワーク入力、スクリプト入力)
- 制限付きのイベントフィルター機能 (Windows イベントのみ)
- 並列取り込みパイプラインのサポートによるスループットの向上と遅延の削減

構造がしっかりした一部のデータ (json、csv、tsv) は例外として、UF はログソースをイベントに構文解析しないため、ログ形式の理解が必要となるアクションは実行できません。また、Python の縮小版も付属しているため、フル Splunk スタックの機能が必要なモジュール式入力 App とは互換性がありません。

Splunk 環境のエンドポイントやサーバーでは、大量の UF (数百から数万) がデプロイされることもあり、Splunk デプロイサーバーまたはサードパーティの設定管理ツール (Puppet や Chef など) で一元管理されます。

(HF) ヘビーフォワーダー

ヘビーフォワーダー (HF) は、インデックス作成が無効化されたフォワーダーとして機能する Splunk Enterprise の完全版デプロイです。通常、HF は他の Splunk ロールは実行しません。UF と HF の主な違いは、HF には完全な構文解析パイプラインがあってインデクサーと同一の機能を実行しますが、インデックス作成イベントを実際にディスクに書き込むことはしません。これにより、HF は個別のイベントを理解して、データをマスクしたり、イベントデータに従ってフィルタリングやルーティングを実行したりといった処理が行えます。Splunk Enterprise の完全版デプロイであるため、データ収集機能を適切に実行する Python スタックを必要とするモジュール式入力をサポートしたり、Splunk HTTP イベントコレクター (HEC) 用のエンドポイントとして機能したりすることができます。HF は以下の機能を実行します。

- データのイベントへの構文解析
- 個別のイベントデータに基づいたフィルタリングとルーティング
- UF より大きなリソースフットプリント
- UF より大きな (最大 5 倍) ネットワーク帯域幅フットプリント
- 管理用 GUI

一般に、HF はデータ収集を目的としたエンドポイントにはインストールされません。HF は、データ収集ノード (DCN) または中継転送層を実装するためのスタンドアロンシステムで使用されます。**UF だけでは他のシステムからのデータ収集要件に対応できない場合にのみ HF を使用してください。**

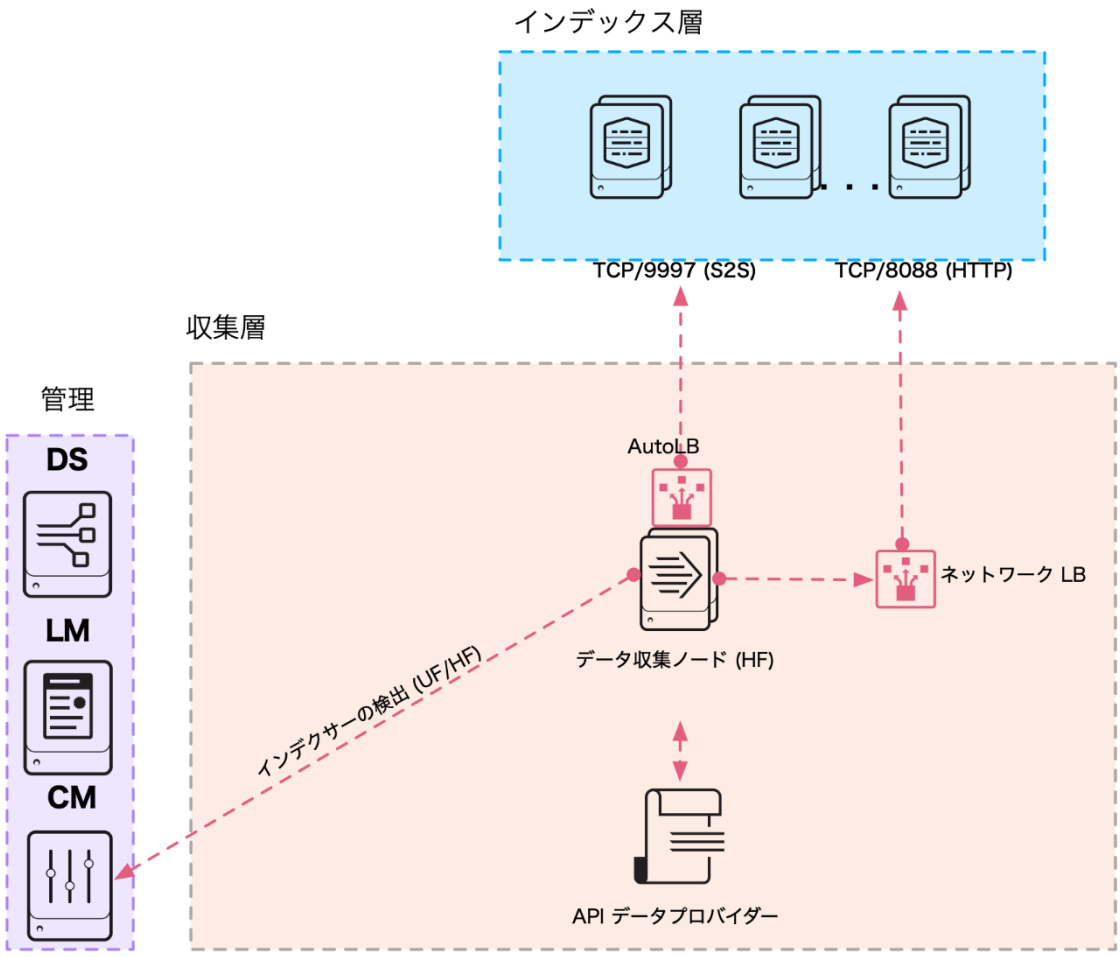
以下のような要件が該当します。

- Splunk に取り込むデータを RDBMS から読み込む場合（データベース入力）
- API でアクセスできるシステムからデータを収集する場合（クラウドサービス、VMWare 監視、固有システムなど）
- HTTP イベントコレクターサービスの専用層を提供する場合
- ルーティング、フィルタリング、またはマスキングを行うために構文解析が可能なフォワーダーを必要とする中継転送層を実装する場合

(DCN) データ収集ノードとしてのヘビーフォワーダー

一部のデータソースでは、ある種の API を使用した収集が必要になります。このような API としては、REST、ウェブサービス、JMS、クエリメカニズムとしての JDBC などがあります。Splunk やサードパーティの開発会社は、これらの API インタラクションをサポートするさまざまな App を提供しています。これらの App の大半は Splunk モジュール式入力フレームワークで実装されており、Splunk Enterprise の完全版が適切に動作していることが必要です。この使用事例を実現するためのベストプラクティスは、1 台または複数台のサーバーをヘビーフォワーダーとしてデプロイし、データ収集ノード (DCN) として機能するように設定することです。

データ収集ノードのトポロジー

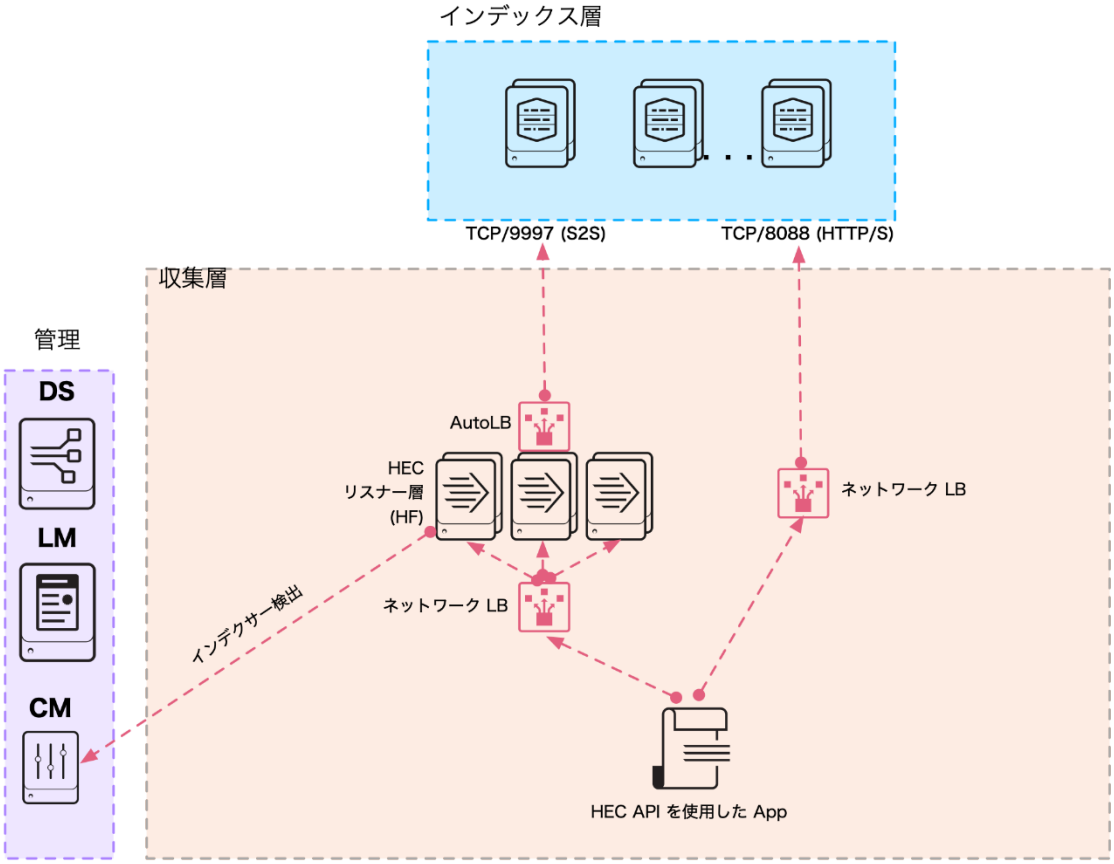


(HEC) HTTP イベントコレクター

HEC は、サーバー側では HTTP/S 接続を受け入れるリスナーサービスを提供し、クライアント側では、App がログデータペイロードを直接インデックス層もしくは 1 台または複数台のヘビーフォワーダーで構成される専用の HEC レシーバー層に書き込めるようにする API を提供します。HEC は、raw 形式または JSON 形式でのデータ送信をサポートする 2 つのエンドポイントを提供します。JSON を利用することで、イベントペイロードに追加のメタデータを含めることができるため、後でデータをサーチする際に柔軟性が向上する場合があります。

HEC の 2 つのデプロイオプションを下図に示します。

HEC トポロジーの選択肢



管理層には（HF が必要とする）ライセンスマスターと、リスナーコンポーネント上の HTTP 入力を管理するデプロイサーバーがあります。注意: インデックス層をクラスタ化して HEC トラフィックを直接受け取る場合、HEC 設定はデプロイ サーバーではなくクラスタマスターによって管理されます。

どのデプロイトポロジーを選ぶかは、それぞれのニーズによって異なります。専用の HEC リスナー層を使用する場合は、さらに別のアーキテクチャコンポーネントがデプロイに追加されます。このトポロジーの長所としては、HEC リスナー層を独立してスケーリングでき、管理をインデックス層と分離できるという点があります。また、専用の HEC 層は HF を必要とするため、すべてのインバウンドトラフィックが構文解析されるので、インデクサーの負担が減ります。

一方で、HEC リスナーをインデクサー上で直接ホスティングするトポロジーでは、すべてのネットワークロードバランサーは HTTP プロトコルをよく理解しているため、最もビジーではないインデクサーに最初にデータが転送されるようにロードバランシングポリシーを設定することで、インデックス層全体でイベントが均等に配分されるようになります。

要件を満たす最もシンプルなアーキテクチャをデプロイするという考えに従い、十分なシステム容量があるのなら、HEC リスナーをインデクサー上でホスティングするトポロジーをお勧めします。このトポロジーを選んだ場合は、後でトポロジーの変更が必要になった場合でも、適切なサイズの HF 層を設定してから、LB の転送先 IP アドレスをインデクサーから HF の IP アドレスに変更するだけで済みます。この変更は、クライアント App に対して透過的である必要があります。

注意: HEC 経由で送信されたデータでインデクサー応答確認が必要な場合は、インデクサーのローリング再起動による重複メッセージを最小限に抑えるために、専用の HEC リスナー層の使用をお勧めします。

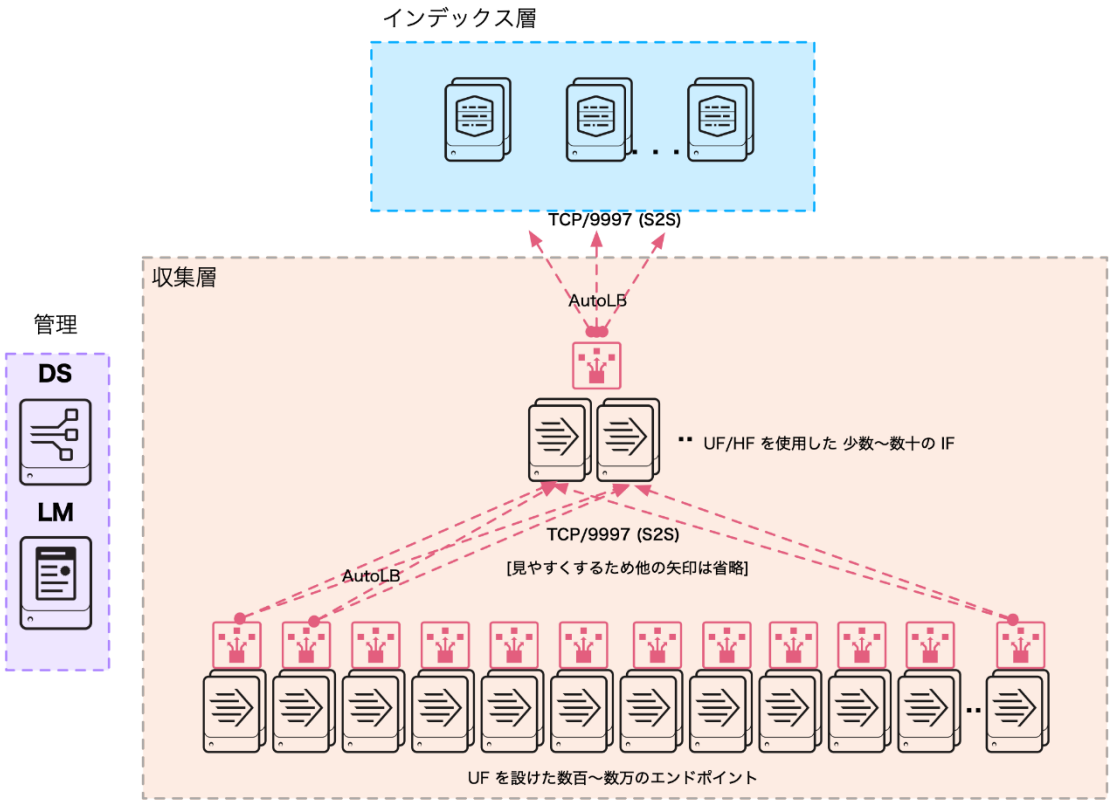
注意: この HEC デプロイアーキテクチャは、後で説明する他のデータ収集コンポーネントの一部（syslog とメトリックス データ収集）のトランスポートを提供します。

（IF）中継転送層

データの転送に中継フォワーダーが必要な場合もあります。中継フォワーダーは、エンドポイントからログストリームを受け取ってインデクサー層に転送します。中継フォワーダーを使用する場合は、Splunk 環境全体への悪影響を回避するために、アーキテクチャを慎重に設定する必要があります。最も重要な点は、中継フォワーダーには数百から数万ものエンドポイントフォワーダーからの接続が集中し、転送先となるインデクサーの接続数はそれよりはるかに少ないという点です。この結果、特定の時点では一部のインデクサーしかトラフィックを受け取らないことになるため、インデックス層でのデータ配分に大きな影響があります。ただし、この影響は、サイジングと設定を適切に行うことで緩和できます。

この課題を下図に示します。

中継転送トポロジー



中継フォワーダーが 1 台のシナリオでは、すべての（数千にも及ぶこともある）エンドポイントがこのフォワーダーに接続しますが、中継フォワーダーは、一度に 1 台のインデクサーとしか接続できません。この結果、以下のような結果が生じることがあるため、このシナリオは最適ではありません。

- 多くのエンドポイントから送られて来た大量のデータストリームが単一のパイプで詰まってしまい、システムやネットワークのリソースを使い尽くしてしまう
- IF の障害時にはエンドポイントのフェールオーバーターゲットが限定されてしまう（停止リスクと IF 数は逆比例の関係となる）
- 特定の時点において稼動しているインデクサー数が少ない。短時間のサーチでは並列実行の効果がそれほど発揮できない

中継フォワーダーは、デプロイにアーキテクチャ層を追加するため、管理やトラブルシューティングが複雑になったり、データ取り込みパスの遅延が増大したりすることもあります。要件を満たすためにどうしても使用しなければならない場合を除いて、中継転送層は使用しないでください。以下の場合は、中間層を設けてもよいでしょう。

- 重要なデータをネットワーク経由でインデクサーに転送する前に暗号化/削除する必要がある場合。たとえば、公共ネットワークを使用する場合など
- セキュリティポリシーにより、エンドポイントとインデクサーとの直接接続が禁止されている場合（マルチゾーンネットワークやクラウドベースインデクサーなど）
- エンドポイントとインデクサーの間の帯域幅の制約により、大量のイベントをフィルタリングしなければならない場合
- ターゲットへのイベントベースの動的ルーティングが必要な場合

中継転送層の可用性を確保し、すべてのトラフィックを処理するのに十分な処理容量を提供して、インデクサー間でイベントを均等に配分するため、中継転送層のサイズと設定を慎重に検討してください。IF 層には以下の要件があります。

- 十分な数のデータ処理パイプラインを設けること
- 冗長 IF インフラストラクチャを設けること
- Splunk ロードバランシング構成を適切に調整すること。例: autoLBVolume、EVENT_BREAKER、EVENT_BREAKER_ENABLE、そして必要に応じて forceTimeBasedAutoLB

一般的な目安として、必要な IF 処理パイプラインの数はインデックス層のインデクサー数の 2 倍です。

注意: 処理パイプラインは物理 IF サーバーと同等ではありません。十分なシステムリソースが確保されている必要があります。たとえば、CPU コア、メモリ、および NIC 帯域幅が十分に利用可能であれば、単一の IF を複数の処理パイプラインで構成できます。

IF 層が必要な場合は（質問票参照）、システムとネットワークのリソースが少ない場合は UF の方が高いスループットを実現するため、UF の使用を検討してください。UF だけでは要件を満足できない場合は、HF を使用してください。

(SYSLOG) syslog データ収集

syslog プロトコルは、企業のログデータの転送によく使用されています。最も拡張性と信頼性の高いデータ収集層に syslog 取り込みコンポーネントを配置します。syslog データを Splunk に取り込むための手段はたくさんあります。以下の方法を検討してください。

- **ユニバーサルフォワーダー (UF)/ヘビーフォワーダー (HF)**: Splunk UF または HF は、syslog サーバーが書き出したファイル（rsyslog や syslog-ng など）の監視（取り込み）に使用します。
- **HEC への syslog エージェント**: Splunkの HEC への出力に対応した syslog エージェントを使用します(HEC に出力可能な rsyslog および syslog-ng 用のサードパーティモジュールが用意されています)。.
- **直接 TCP/UDP 入力**: Splunk では、TCP または UDP ポートでデータを待ち受け（デフォルトポートは UDP 514 です)、ここにソースを取り込むことができます（実働環境では**推奨されません**)。

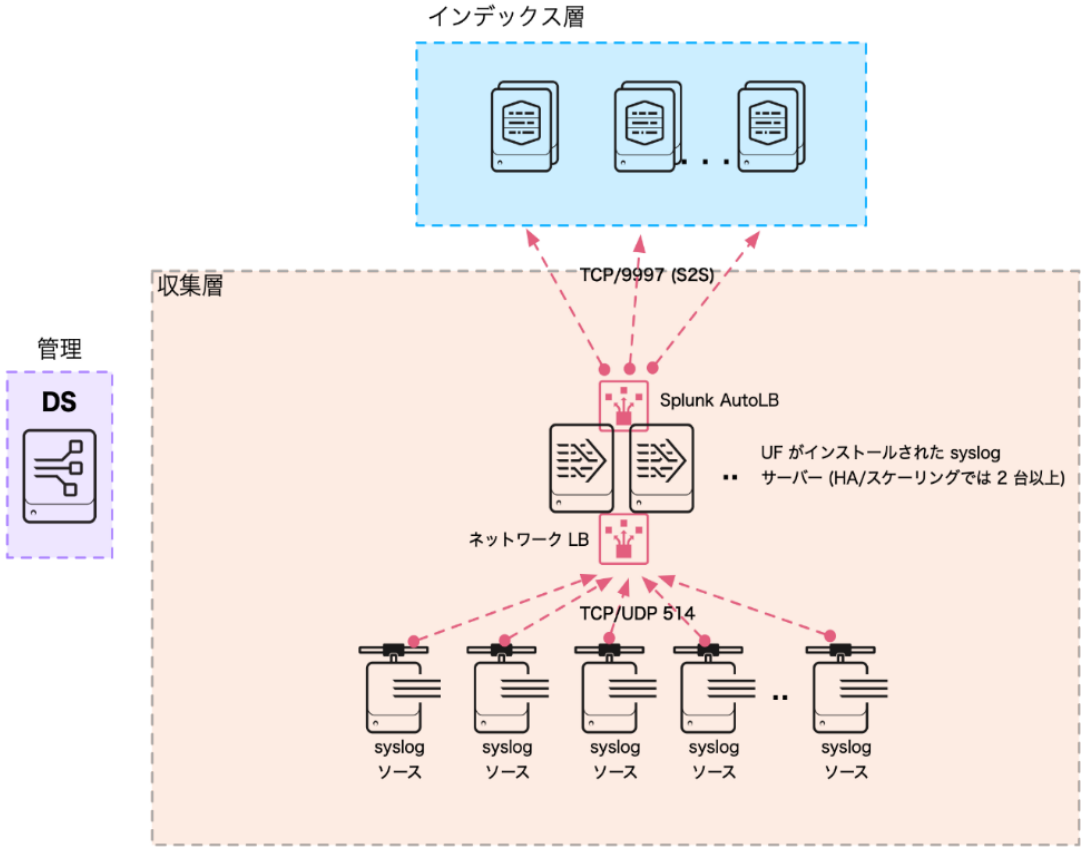
syslog（SCD との併用によるファイル監視）

Splunk では、inputs.conf の設定により、UF/HF 上で監視を実行し、syslog 収集デーモン (SCD) によってエンドポイントでディスクに書き込まれた syslog ソースを処理して取り込むことができます。最も一般的な rsyslog、syslog-ng、および Fastvue は、ボリュームの少ない環境でも大規模な分散環境でも、拡張性が高く、統合と管理が容易な有料および無料のソリューションを提供しています。

監視の設定の詳細については、『データの取り込み』の「ファイルとディレクトリの監視」を参照してください。

このアーキテクチャでは、他のエンドポイントでユニバーサルフォワーダーが行うのと同じ方法で適切なデータのオンボードをサポートします。SCD を適切に設定することで、複数の異なるログタイプを特定して、ログイベントを Splunk のフォワーダーが検出できるファイルとディレクトリに書き込むようにすることができます。イベントをディスクに書き込むことで、syslog ログストリームの継続性が強化され、信頼性の低い UDP をトランスポートとして使用した場合に発生しうるメッセージのデータロスを抑えることができます。

UF を使用した syslog データ収集トポロジー



この図は、ポート 514 で TCP または UDP を使用して syslog サーバーのロードバランシングプールにデータを送信する syslog ソースを示しています。複数のサーバーにより、収集層の高可用性が保証され、メンテナンス中のデータロスを防止できます。各 syslog サーバーは、syslog イベントが各ソースタイプ（ファイアウォールイベント、OS syslog、ネットワークスイッチ、IPS など）ごとの専用のファイル/ディレクトリに書き込まれるようにするためのルールを syslog ストリームに適用するように設定されます。各サーバーにデプロイされる UF はこれらのファイルを監視して、データが適切なインデックスに処理されるようにインデックス層に転送します。Splunk AutoLB を使用することで、利用可能なインデクサー間でデータが均等に配分されます。

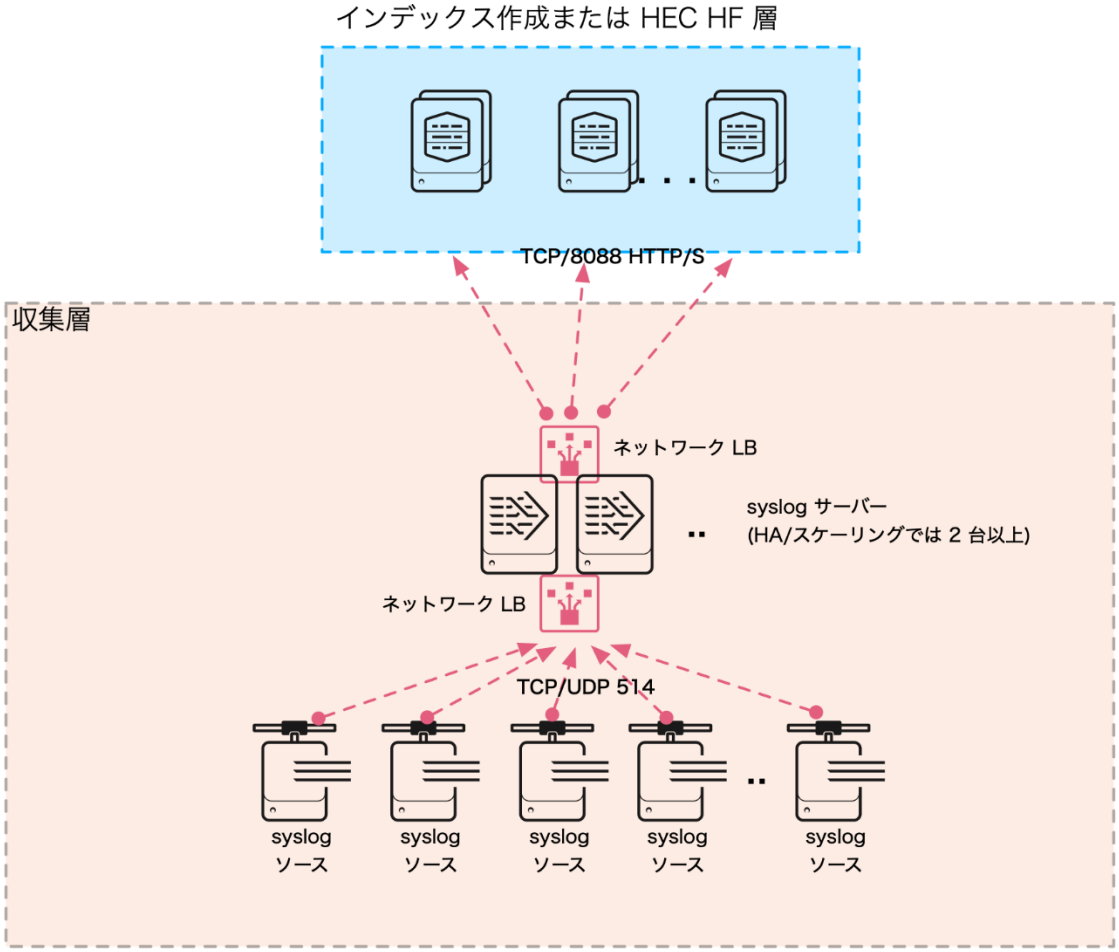
管理層に示されているデプロイサーバーを使用することで、UF 設定を一元管理できます。

HEC への syslog エージェント

HEC の利用が増えたため、HEC を利用して syslog を取り込むデプロイが増えています。詳しくは、Splunk ブログ記事「syslog-ng と HEC: Splunk での拡張性の高い集計データ収集」を参照してください。

下図は、ネットワークロードバランサーを使用してポート 514 から syslog サーバーファームにデータを送信する syslog ソースを示しています。syslog の送り先が指定された適切な syslog ポリシー（HEC API を使用した Python スクリプト）が適用され、イベントが HEC リスナーに送られ、インデクサー用のネットワークトラフィックロードバランサーが使用されています。

HEC を使用した syslog データ収集トポロジー



このトポロジーの利点は、UF/HF のデプロイと設定が不要なことです。HTTP ロードバランサーは、インデクサー（または専用 HEC リスナー層）上で HEC リスナーとして機能し、HEC エンドポイント間でデータを均等に配分します。このロードバランサーは、「最小接続」ポリシーで設定します。

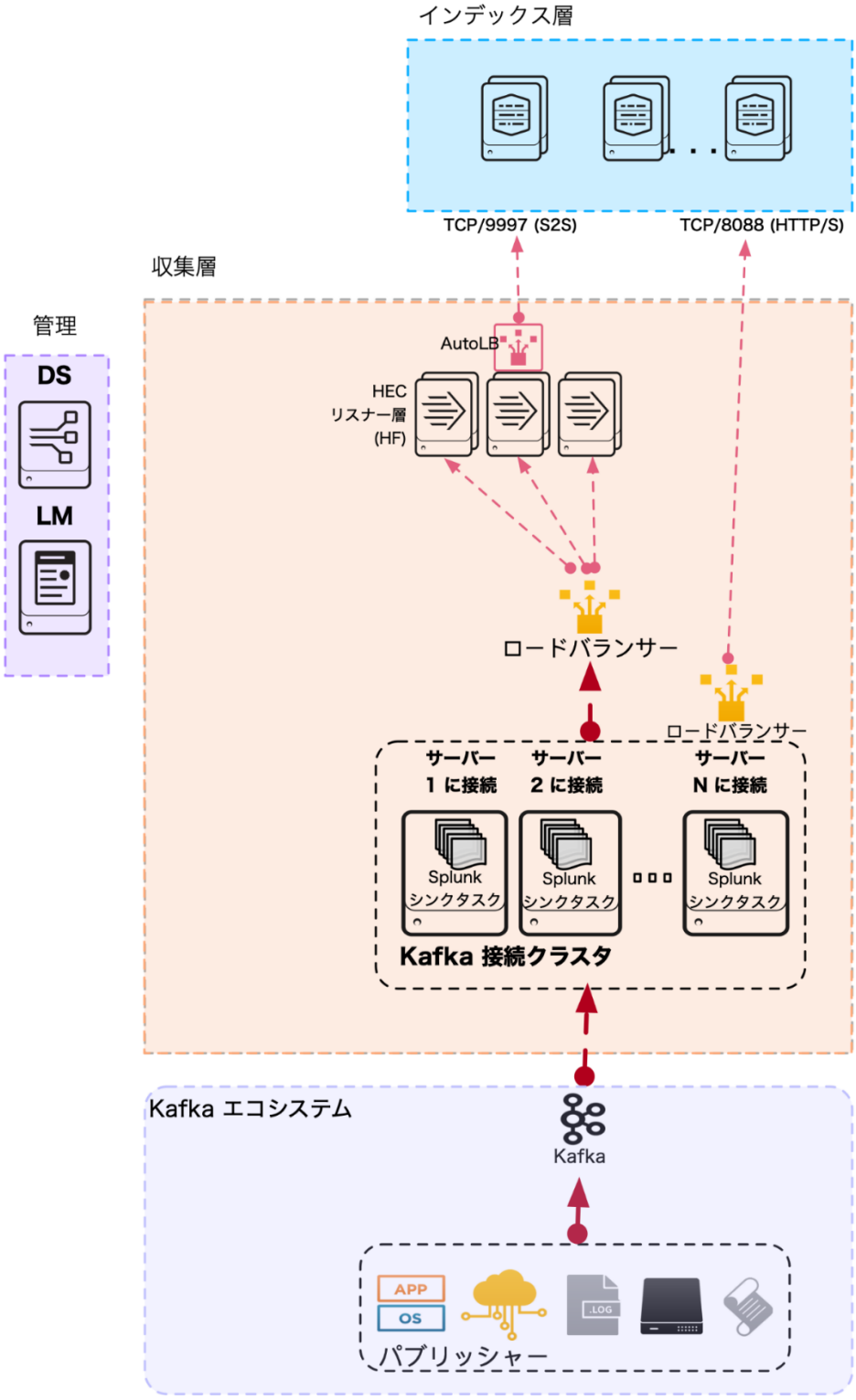
Splunk UDP 入力

Splunk では、UF または HF 上で直接 UDP 入力を使用して syslog からデータを受け取ることができます。TCP および UDP ポートの設定については、『データの取り込み』の「TCP および UDP ポートからのデータの取り込み」を参照してください。UDP 514 でデータを受け取れるかどうかは、ルートとして動作している UF/HF の能力に依存します。また、データロスを防止するため、エージェントは常に利用可能でなければなりません。フォワーダーは変更を適用するために頻繁に再起動することがあり、データロスの原因となります。この理由から、**このトポロジーは実働環境ではベストプラクティスとは考えられません。**

(KAFKA) Kafka トピックからのログデータの消費

Splunk は、Splunk Connect for Kafka というシンクコネクタによって Kafka からのデータの消費をサポートしています。製品の詳しい説明については、『Splunk Connect for Kafka』マニュアルの「Apache Kafka Connect」を参照してください。Splunk Connect for Kafka パッケージは、適切なサイズの Kafka Connect クラスタ（Splunk 外）にインストールされ、設定に従ってトピックにサブスクライブして、消費したイベントを HEC を使用してインデックス処理用に送信します。

Kafka と HEC を使用したデータ収集トポロジー

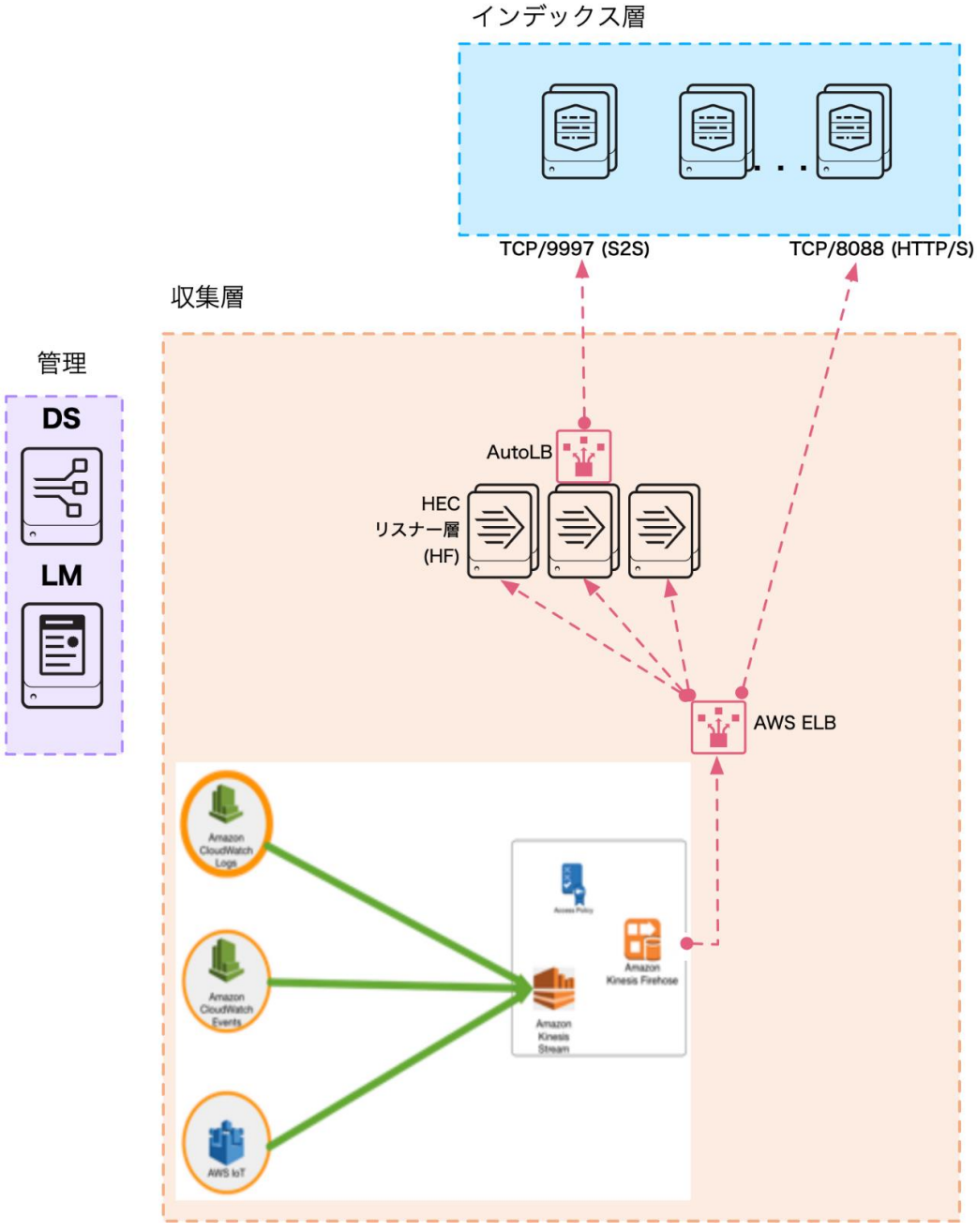


この図は、Kafka バスにメッセージを送信する Kafka パブリッシャーを示しています。Kafka Connect クラスタでホスティングされるタスクは、Splunk Connect for Kafka 経由でこれらのメッセージを消費し、ネットワークロードバランサーを使用して HEC リスニングサービスにデータを送信します。ここでも、HEC リスニングサービスをインデクサー上で直接ホスティングするか、または専用の HEC リスナー層を設けることができます。詳細については HEC のセクションを参照してください。管理層コンポーネントは、専用の HF 層をデプロイして HEC リスナーをホスティングする場合にのみ必要です。

(KINESIS) Amazon Kinesis Firehose からのログデータの消費

Splunk と Amazon は、Kinesis と Splunk HEC を統合することで、AWS から HEC エンドポイント（AWS コンソールから設定可能）にデータを直接ストリーミングできるようにしました。この機能は、AWS で発生する各種データソース用の CIM 準拠ナレッジを提供する Splunk Add-On for Kinesis Firehose によって補完されています。

Amazon Kinesis を使用したデータ収集トポロジー



この図は、Kinesis を Firehose に送信する AWS ログソースを示しています。Firehose は、適切な設定により、AWS ELB 経由で HEC リスニングサービスにデータを送信します。ここでも、HEC リスニングサービスをインデクサー上で直接ホスティングするか、または専用の HEC リスナー層を設けることができます。詳細については HEC のセクションを参照してください。

図中の管理層コンポーネントは、専用の HF層をデプロイして HEC リスナーをホスティングする場合表示にのみ必要です。

(METRICS) メトリックスの収集

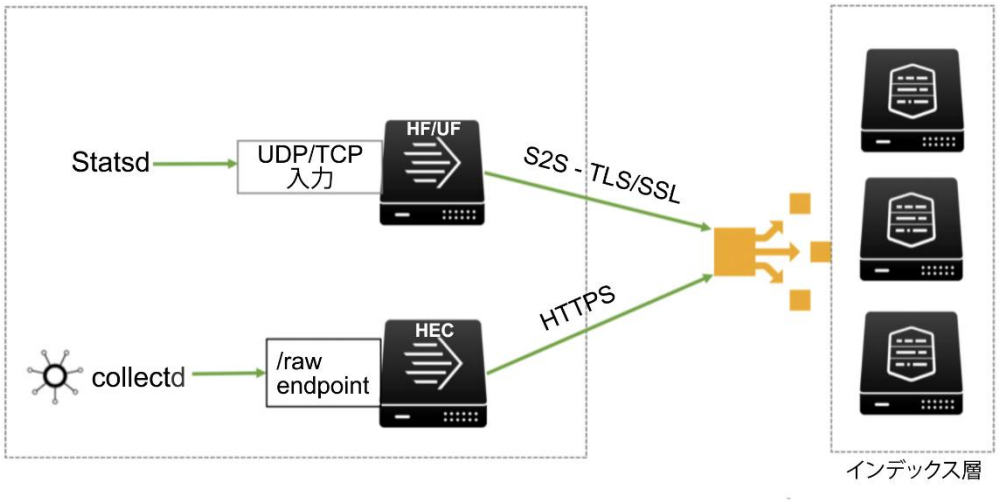
Splunk には、システムと App のパフォーマンスデータ（メトリックスデータ）をさまざまなサードパーティソフトウェアから受け取って収集する機能があります。Splunk プラットフォームでのメトリックスは、メトリックスストレージと取得に最適化されたカスタムインデックスタイプを使用します。

メトリックスデータを消費する方法はいくつかあり、収集方式は使用する技術によって決まります。最も一般的なメトリックス収集形態は **collectd** や **statsd** などのソフトウェアデーモン、またはカスタムメトリックスデータファイルと有効なデータソース設定を使用する方法です。

statsd や **collectd** などのエージェントを使用して Splunk にメトリックスを取り込む方法は主に 2 つあります。**直接 TCP/UDP** 入力を使用する方法と **HEC** を経由で取り込む方法です。

HEC エンドポイントの高い回復性と拡張性、そして収集層の水平スケーリングの容易さから、**HEC** を使用する方法がベストプラクティスと考えられます。

メトリックスデータ収集トポロジー



statsd は現在、**UDP** および **TCP** トランスポートをサポートし、Splunk フォワーダーまたはインデクサー上の直接入力として使用できます。ただし、このアーキテクチャは回復性が低く、Splunk フォワーダーの再起動によるイベントロスが発生しやすいので（syslog 収集参照）、実働環境では TCP/UDP トラフィックを直接フォワーダーに送信することがベストプラクティスです。

(HA) 転送層コンポーネントの高可用性の考慮事項

デジタルの世界には、高可用性（HA）の共通コンセプトがあります。しかし、その意味は組織によって異なることがあり、高可用性よりも障害復旧（DR）に近い意味となることもあります。これら 2 つのコンセプトは似ていますが意味は異なります。HA とはシステムの特性であり、求められる動作パフォーマンス（アップタイム）を通常より長い間保証することを目的とします。DR にはいくつかのポリシー、ツール、そして手順が含まれ、障害発生後に健全な技術インフラストラクチャやシステムの状態まで戻す、あるいはその状態を継続させることを目的とします。

中間層と集計層での HA の形態を以下に示します。

中間層

- 中間層/集計層をデプロイしている場合は、フォワーダーの HA が重要です。現時点では、Splunk はアプリケーション層の HA をネイティブサポートしていません。オペレーティングシステムレベルでは、Splunk のネイティブオプションではない方法で HA を実現する戦略がいくつかあります。代表的なオプションとしては、VMWare VMotion、AWS Autoscaling Groups、Linux クラスタリングなどがあります。利用できる設計オプションについては、Splunk アーキテクトにお問い合わせください。
- 専用 HEC 層の HA 要件がある環境では、NGINX などのネットワークトラフィックロードバランサー（NTLB）を Splunk ヘビーフォワーダーの前に配置することがベストプラクティスです。これにより、スループット、スケール、そして可用性の利点を最大限に引き出すことができます。HTTP イベントコレクターインスタンスの専用のプールがあり、そのジョブのみがデータを受信および転送します。HEC インスタンスをさらに追加できますが、必ずしもインデクサーを追加する必要はありません。インデクサーがボトルネックになる場合は、インデクサーをさらに追加します。
- syslog 収集の HA 要件がある環境では、HAProxy や F5 などのロードバランシングソリューションでホスティングされたクラスタ（仮想）IP アドレスで管理される複数の syslog サーバーを使用することで、スループット、スケール、そして可用性を最大限に高めることがベストプラクティスです。Splunk インスタンスの専用のプールがあり、そのジョブのみがデータを受信および転送します。インスタンスをさらに追加できますが、必ずしもインデクサーを追加する必要はありません。インデクサーがボトルネックになる場合は、インデクサーをさらに追加します。

転送層

- 転送（エンドポイント）層では、エージェント自身の HA は基礎となる OS に依存します。最低限、ホスト OS の再起動時には転送機能を実行するサービスも再起動することを保証してください。それ以外のフォワーダーのベストプラクティスとしては、複数のインデックスに対して AutoLB from フォワーダーを適切に設定して使用することがあります。これには、データがインデックス層に到着したことを保証するためのインデクサー応答確認も含まれます。

ステップ3：設計方針とベストプラクティスの適用

以下に、デプロイ層ごとの設計方針とベストプラクティスを示します。

デプロイ層

SVA の設計方針は、以下のデプロイ層をすべてカバーします。

層	定義
サーチ	• サーチヘッド
インデックス	• インデクサー
収集	• フォワーダー • モジュール入力 • ネットワーク • HEC（HTTP イベントコレクター） • その他
管理/ユーティリティ	• CM • DS • LM • DMS • SHC-D

トポロジーとベストプラクティスとの調整

デプロイに適した設計方針とベストプラクティスを選択するためには、要件とトポロジーを念頭に置いて検討する必要があります。そのため、ベストプラクティスを検討するのは、Splunk Validated Architecture 選択プロセスのステップ 1 と 2 を完了した後にしてください。

ベストプラクティス: 層特有の推奨事項

以下に、各デプロイ層で推奨される設計方針とベストプラクティスを示します。それぞれの設計方針は、SVA の柱 （可用性、パフォーマンス、拡張性、セキュリティ、そして管理性）のいずれか 1 つあるいは複数を補強します。

サーチ層の推奨事項

設計方針/ベストプラクティス （適用すべきベストプラクティスは要件によって決まります。）		SVA の柱				
		可用性	パフォーマンス	拡張性	セキュリティ	管理性
1	サーチ層を（ネットワーク上で）インデックス層の近くに配置する サーチ層とインデックス層の間のネットワーク遅延は、サーチパフォーマンスに直接影響します。		✓			
2	複数の独立したサーチヘッドの使用を避ける 独立したサーチヘッドは、ユーザーが作成した Splunk アーチファクトの共有ができません。また、サーチ層でのリソース利用に関しては拡張性が乏しくなります。分離されたサーチヘッド環境が必要な場合を除いて、拡張性を重視したオプションを使用すべきです。	✓		✓	✓	✓
3	サーチ層のスケーリングにサーチヘッドクラスタリングを利用する サーチヘッドクラスタでは、クラスタ間でユーザーアーチファクトが複製され、すべてのクラスタメンバー間でインテリジェントなサーチワークロードスケジュールが可能に	✓		✓		

	なります。また、高可用性ソリューションも実現します。					
4	すべてのサーチヘッドの内部ログをインデックス層に転送する すべてのインデックス化データはインデックス層いのみ格納する必要があります。これにより、サーチヘッド層で高性能ストレージを用意する必要がなくなり、管理が簡素化されます。注意: これは他のすべての Splunk ロールにも適用されます。		✓			✓
5	可能であれば LDAP 認証の使用を検討する 認証を目的としてユーザー ID を一元管理することは、一般的な企業にとってベストプラクティスとなり、Splunk デプロイの管理が簡素化される上、セキュリティが向上します。				✓	✓
6	同時サーチのニーズに対応するため十分なコアを確保する すべてのサーチでは CPU コアによる実行が必要です。サーチを実行するコアが利用できない場合は、サーチはキューに登録され、遅延が発生します。注意: インデックス層にも適用されます。	✓	✓	✓		
7	スケジュール済みサーチ時間ウィンドウを可能な限り利用する/スケジュール済みサーチ負荷を均等化する スケジュール済みサーチは特定の時間に実行されます（毎時、毎時5分/15分/30分、深夜など）。サーチが実行できる時間ウィンドウを提供することで、サーチが混み合うホットスポットを回避できます。		✓	✓		
9	インデックス層に負担をかけないように個別のサーチヘッドクラスタ数を制限する	✓		✓		

	サーチワークロードを自動的に管理できるのは SH 環境においてのみです。独立した SHC が多いと、インデクサー（サーチピア）層が処理できる能力を上回る同時サーチワークロードが発生することがあります。スタンドアロンサーチヘッドの数を計画する際にも同じ配慮が必要です。					
10	サーチヘッドクラスタを構築する際にはノード数を奇数（3、5、7 など）にする SHC キャプテンの選出は過半数ベースのプロトコルで行われます。ノード数が奇数であれば、ネットワーク障害時に SHC が同数に分けられることを回避できます。	✓				✓

インデックス層の推奨事項

設計方針/ベストプラクティス （適用すべきベストプラクティスは要件によって決まります。）		柱				
		可用性	パフォーマンス	拡張性	セキュリティ	管理性
1	対応するサーバーでは並列パイプラインを有効にする 並列化により、アイドル状態で利用可能なシステムリソースを有効利用できます。並列取り込み機能を有効にする前に I/O パフォーマンスが適切であることを確認してください。		✓	✓		
2	ホット/ウォームボリュームとサマリーでは SSD の使用を検討する SSD は価格も下がり、サーチパフォーマンス低下の原因になっていた I/O の制約も無くなりました。		✓			

3	インデックス層を（ネットワーク上で）サーチ層の近くに配置する ネットワーク遅延を最小限に抑えることで、サーチ時のユーザーエクスペリエンスを改善できます。		✓			
4	履歴データ/レポートの HA が必要な場合はインデックス複製を使用する インデックス複製により、クラスタ内ですべてのイベントの複数コピーが保存され、サーチピア障害からデータを保護できます。SLA に合わせてコピー数（複製データ保持数）を設定してください。	✓				
5	データの形式を正しく保ち（例: 改行、タイムスタンプ抽出、TZ、ソース、ソースタイプ、ホストを各データソースに対して適切かつ明示的に定義する）、モニタリングコンソールを使用したリアルタイム監視を行う データソースを明示的に設定し、Splunk の自動検出機能を利用することで、特に高ボリュームデプロイではデータ取り込み容量とインデックス作成遅延を改善できます。		✓	✓		✓
6	処理能力が過剰なインデクサーではバッチモードのサーチ並列化設定を検討する サーチの並列化機能により、特定タイプのサーチではサーチパフォーマンスを改善することができ、未使用のシステムリソースを活用できます。		✓	✓		
7	インデクサーノード（サーチピア）間でデータの均等な配分を監視する サーチパフォーマンスを高めて適切なデータ保存ポリシーを保証するためには、サーチピア間でイベント/データを均		✓	✓		✓

	等に配分することが重要です。					
8	分散/クラスタ化デプロイではインデクサー上のウェブ UI を無効化する インデクサー上で直接ウェブ UI にアクセスする理由はありません。		✓		✓	✓
9	よく知られているデータソースでは Splunk の構築済み技術アドオンの利用を検討する よく知られているデータソースの場合、自分でデータオンボードの設定を構築するのではなく、Splunk が提供している技術アドオンを利用することで、すぐに実働環境に移行して最適な実装を保証することができます。		✓			✓
10	重要なインデクサーメトリックスを監視する Splunk は、インデックス層のパフォーマンスに関する主要なメトリックスを提供するモニタリングコンソールを用意しています。これには、CPU やメモリの利用状況、内部 Splunk コンポーネント（プロセス、パイプライン、キュー、サーチ）の詳細なメトリックスが含まれます。	✓	✓			

収集層の推奨事項

設計方針/ベストプラクティス （適用すべきベストプラクティスは要件によって決まります。）		**柱**				
		可用性	**パフォーマンス**	**拡張性**	**セキュリティ**	**管理性**
1	可能な限り UF を使用してデータを転送する。ヘビーフォワーダーの使用は、必須である場合に限定する 組み込み autoLB は再起動に対応し、一元設定が可能で、		✓			✓

	リソース要求量も少なくて済みます。					
2	多くの UF からデータが転送される場合は、中継転送パイプライン数をインデクサーの 2 倍以上にする 多くのエンドポイントフォワーダーを少数の中継フォワーダーで処理しようとすると、インデクサー間の均等なイベント分散ができなくなり、サーチパフォーマンスが低下します。**中継フォワーダーは絶対に必要な場合にのみ使用してください。**	✅	✅			
3	SSL による UF-IDX トラフィックのセキュリティを検討する				✅	
4	ネイティブ Splunk LB を使用してインデックス層でデータを分散させる 現時点では、フォワーダーとインデクサーの間ではネットワークロードバランサーはサポートされていません。	✅		✅		
5	syslog 収集には専用の syslog サーバーを使用する syslog サーバーは、TCP/UDP トラフィックをソースごとにディスクに書き込み、ユニバーサルフォワーダーでの処理に合わせたソースタイプ設定が可能です。フォワーダーが再起動してもデータロスは発生しません。	✅				✅
6	HEC を（TCP/UDP の代わりに）使用してエージェントレス収集を行う HTTP イベントコレクター（HEC）は、HTTP[/S] プロトコルでイベントを収集するリスニングサービスです。インデクサー上で直接有効にすることも、ヘビーフォワーダー層で設定することもでき、どちらもロードバランサーによって管理されます。	✅				✅

管理/ユーティリティ層の推奨事項

設計方針/ベストプラクティス (適用すべきベストプラクティスは要件によって決まります。)		柱				
		可用性	パフォーマンス	拡張性	セキュリティ	管理性
1	LM、CM、SHC-D、および MC を単一インスタンスに統合して環境規模を小さくする これらのサーバーロールはリソースをほとんど消費しないため、コロケーション候補としては最適です。大規模なインデクサークラスタでは、クラスタを効率的に管理するために CM を専用サーバーとする必要があります。					✅
2	中規模～大規模デプロイでは DS を独立したインスタンスとする デプロイサーバーで大量のフォワーダーを管理する場合、リソースのニーズが高まり、サービスの維持には専用サーバーが必要となります。					✅
3	超大規模のデプロイでは、LB の後ろに複数の DS の配置を検討してください。 注意: このデプロイを適切にセットアップして設定するには、Splunk プロフェッショナルサービスの補助が必要な場合があります。	✅		✅		✅
4	DS の問い合わせ間隔をデフォルトの 60 秒より長くする 問い合わせ間隔が長いほど DS の拡張性が向上します。			✅		
5	専用/安全な DS を使用して App デプロイ経由でのクライアントのデータ漏洩を回避する デプロイサーバーにアクセスできる人は、フォワーダーエンドポイントへの悪意のある App のデプロイも含め、誰で				✅	

	もその DS が管理する Splunk 設定を変更できてしまいます。このロールのセキュリティを適切に確保することが重要です。					
6	モニタリングコンソール（MC）を使用してデプロイの健全性を監視し、健全性の問題についてのアラートをトリガーする モニタリングコンソールは、構築済みの Splunk 特有監視ソリューションセットを提供し、環境の健全性が低下したときに通知を出す拡張可能なプラットフォームアラートがあります。	✅	✅			✅

サマリーと次のステップ

このホワイトペーパーでは、Splunk Validated Architecture の概要を説明しました。SVA により、最もコスト効果が高く、管理が容易で拡張性に優れた方法によって組織の要件を満たすことができます。SVA は、以下の基本的な柱に基づいたベストプラクティスと設計方針を提供します。

- 可用性
- パフォーマンス
- 拡張性
- セキュリティ
- 管理性

また、このホワイトペーパーでは、3 ステップの Splunk Validated Architecture の選択プロセスを紹介しました。3 つのステップとは、1) 要件の定義、2) トポロジーの選択、そして 3) 設計方針とベストプラクティスの適用です。Splunk Validated Architecture のいろいろな利点を理解しましたので、次のステップに進んで組織に適したデプロイトポロジーを選択する準備ができたことになります。

次のステップ

では SVA を選択した後は何をすべきなのでしょうか。実働環境へと向けた次のステップは以下の通りです。

カスタマイズ

- 選択したトポロジーが特定の要件を満たすために必要なカスタマイズについて検討してください。

デプロイモデル

- デプロイモデル（ベアメタル、仮想、クラウド）を決定します。

システム

- Splunk のシステム要件に従ってテクノロジ（サーバー、ストレージ、オペレーティングシステム）を選択します。

サイジング

- デプロイのサイズを決めるための関連データを集めます（データ取り込み、予想されるデータボリューム、データ保存ニーズ、複製など）。Splunk ストレージサイジング（https://splunk-sizing.appspot.com/）が役に立つでしょう。

要員

- デプロイを実装して管理するのに必要な要員を評価してください。このステップは、Splunk Center of Excellence を構築するための重要なパートとなります。

Splunk は、SVA プロセスと次のステップをお手伝いいたします。ご質問がある場合は、Splunk アカウントチームまでお尋ねください。アカウントチームは、Splunk のあらゆる技術リソースやアーキテクチャリソースにアクセスして、必要な情報を提供します。

ぜひご活用ください。

付録

この付録では、SVA で使用する追加の参考情報を提供します。

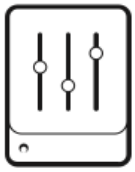
付録 A:SVA の柱の説明

柱	説明	主な目標/設計方針
可用性	常に稼動していて、計画的あるいは予期しない停止や中断から復旧できること。	1. 単一障害点の排除/冗長性の追加 2. 計画的および無計画の障害/サービス停止の検出 3. 計画的および無計画のサービス停止への（できれば自動的な）対応 4. ローリングアップグレードの計画
パフォーマンス	利用可能なリソースを効果的に利用して、変動する利用パターンにおいて最適なサービスレベルを維持すること。	1. ハードウェアの追加によるパフォーマンスの改善（CPU、ストレージ、メモリ） 2. 「ボトムアップ」のボトルネックの解消 3. すべての並列処理手段の活用 4. ローカリティの活用（コンポーネント分散の最小限化） 5. 共通ケースの最適化（80/20 ルール） 6. 不要な一般化の回避 7. タイムシフト計算（事前計算、レイジー計算、共有/バッチ計算） 8. 時間の確実性と精度のトレードオフ（ランダム化、サンプリング）
拡張性	システムが、すべての層で拡張できるように設計されていて、増大したワークロードに効果的に対応できること。	1. 垂直および水平スケーリング 2. 個別にスケーリングする必要のある機能コンポーネントの分離 3. コンポーネント間の依存関係の最小限化 4. 分かっている将来の成長へのできるだけ早期の対応 5. システム全体の設計への階層の導入

柱	説明	主な目標/設計方針
セキュリティ	システムが価値を提供しつつデータや設定/アセットを保護するように設計されていること。	1. 最初から安全なシステムの設計 2. すべての通信での最新のプロトコルの採用 3. イベントデータへの広範アクセスと詳細アクセスの実現 4. 一元認証の導入 5. 監査手順の実装 6. 攻撃または悪意のある領域の削減
管理性	すべての層で一括運営、一括管理できるようにシステムが設計されていること	1. 一元管理機能の提供 2. 設定オブジェクトライフサイクルの管理（リソースコントロール） 3. App（Splunk）使用の測定と監視/プロファイリング 4. システム健全性の測定と監視

付録 B:トポロジーコンポーネント

層	コンポーネント	アイコン	説明	備考
管理	デプロイサーバー（DS）		デプロイ サーバーは、フォワーダー設定を管理します。	専用のインスタンスにデプロイする必要があります。仮想化することで障害復旧が容易になります。
	ライセンスマスター（LM）		ライセンスマスターは、ライセンスが必要な機能を他の Splunk コンポーネントが有効にして日々のデータ取り込みボリュームを追跡するために必要です。	ライセンスマネージャーロールは最小限の容量と可用性の要件しか持たないため、他の管理機能とコロケーションすることができます。仮想化することで障害復旧が容易になります。
	モニターコンソール（MC）		モニタリングコンソールは、環境の使用状況と健全性を監視するためのダッシュボードを提供します。また、多くのパッケージ済みプラットフォームアラートが用意されて	クラスタ化環境では、MC をマスターノードとコロケーションすることができます。非クラスタか環境では、さらにライセンスマスターやデプロイサーバー機

層	コンポーネント	アイコン	説明	備考
			おり、カスタマイズすることで稼動に関する問題を通知することができます。	能ともコロケーションが可能です。仮想化することで障害復旧が容易になります。
	クラスタマスター（CM）		クラスタマスターは、クラスタ化デプロイにおけるすべてのアクティビティの調整役として必要です。	インデックスバケツが多い（データボリューム/保存量の多い）クラスタでは、通常はクラスタマスターを専用サーバーで動作させる必要があります。仮想化することで障害復旧が容易になります。
	サーチヘッドクラスタデプロイヤー（SHC-D）		サーチヘッドクラスタデプロイヤは、SHC をブートストラップして、クラスタの Splunk 設定を管理するために必要です。	SHC-D は実行時コンポーネントではなく、システム要件も最小限です。他の管理ロールとのコロケーションが可能です。注意: 各 SHC は、それぞれの SHC-D 機能が必要です。仮想化することで障害復旧が容易になります。
サーチ	サーチヘッド（SH）		サーチヘッドは、Splunk ユーザーに UI を提供し、スケジュール済みサーチアクティビティを調整します。	サーチヘッドは分散デプロイにおける専用 Splunk インスタンスです。適切な CPU とメモリリソースがあれば、サーチヘッドを仮想化することで障害復旧が容易になります。
	サーチヘッドクラスター（SHC）		サーチヘッドクラスタは、クラスタ化された 3 台以上のサーチヘッドから構成されます。サーチヘッド層の水平拡張性を実現市、サービス停止時には透過的なユーザーフェールオーバーを保証します。	サーチヘッドクラスタは、理想的には同じシステム仕様を持つ専用サーバーが必要です。適切な CPU とメモリリソースがあれば、サーチヘッドクラスタメンバーを仮想化することで障害復旧が容易になります。
インデックス	インデクサー		インデクサーは、Splunk の頭脳と心臓とも言える重要なコンポーネントです。インデクサーは、受け取ったデータを処理してインデックスを作成	分散デプロイでもクラスタ化デプロイでも、インデクサーは常に専用サーバーで動作する必要があります。単一サーバーデプロイで

層	コンポーネント	アイコン	説明	備考
			し、サーチ層で開始されたサーチリクエストを処理するサーチピアとしても機能します。	は、インデクサーはサーチ UI とライセンスマスター機能も提供します。インデクサーは、ベアメタルサーバー、または適切なリソースが確保されている専用の高性能仮想マシンで最高のパフォーマンスを発揮します。
データ収集	フォワーダーと他のデータ収集コンポーネント		データ収集に関与する任意のコンポーネントの一般アイコンです。	ユニバーサルフォワーダーとヘビーフォワーダー、ネットワークデータ入力、および他の形式のデータ収集（HEC、Kafka など）が含まれます。