

# データセンター（DC）ネットワークを 極めて効果的に運用するための 7つの習慣



Packet Pushersホワイトペーパー

## 目次

はじめに	3
7つの習慣	4
1. ビジネスの成果を中心に設計する	4
2. 信頼性と反復性に優れた自動化を実行する	4
3. 1回学んだら何度でも使用する	5
4. 適切な機器を選択する	5
5. 作業を進めながら検証する	5
6. 後手に回らずプロアクティブに行動する	6
7. 作業をドキュメント化する	6
ApstraとIBNの概要	7
コア原則	7
Apstraのコンポーネント	8

# はじめに

データセンターのネットワークエンジニアになるということは、複雑さと不確実性を受け入れるということです。1か所を変更すると、他の場所に想定外の影響が及ぶ可能性があります。パフォーマンスの低下や、ワークロードが脅威にさらされるリスク、ネットワークのクラッシュが発生する可能性もあります。

ネットワーク要件が絶えず変化する不確実な環境で運用をうまく行うためには、ネットワークエンジニアが良い習慣を身に付けることが必要です。経験豊富なネットワークエンジニアは慎重で、思慮深く、几帳面な傾向があります。ネットワークエンジニアは従来、変更を行う前にネットワークに関する情報をSNMPトラップ、syslog、デバイス構成、パケットキャプチャ、ストリーミングテレメトリから探し出してきました。何か異常があると、問題を発見、診断、修正するためのプロセスを実行します。

これらの一般的な慣行もエンジニアの役に立ちますが、本書ではデータセンターの管理と運用における不確実性にネットワークエンジニアが適切に対処するために効果的な7つの習慣を提案します。

これらの習慣はベストプラクティスとして独立して実践することが可能である一方、IBN（インテントベースネットワーク構築）の概念に基づいて構築されたソフトウェアプラットフォームのJuniper Apstraと連携するものであります。IBNは、データセンターネットワークにおいて信頼性の高い自動化とオーケストレーションを実現します。マルチベンダー環境で動作するApstraソフトウェアは、変更を自動化、オーケストレーション、検証し、結果がインテント（意図）に一致する成果をもたらします。

7つの習慣は次のとおりです。

1. ビジネスの成果を中心に設計する
2. 信頼性と反復性に優れた自動化を実行する
3. 1回学んだら何度でも使用する
4. 適切な機器を選択する
5. 作業を進めながら検証する
6. 後手に回らずプロアクティブに行動する
7. 作業をドキュメント化する

本書では、データセンターネットワークを効果的に運用するための7つの習慣を紹介し、Apstraでその習慣を確実に実施する仕組みを説明します。

# 7つの習慣

## 1. ビジネスの成果を中心に設計する

データセンターネットワークが存在する意義は、ビジネスを推進するアプリケーションとサービスをサポートすることにあります。しかし、一般的なデータセンター設計では、期待される成果ではなくベンダーの選択が中心に据えられる傾向があります。多くの場合、ベンダーの選択はビジネス目標以外の要因に左右されます。営業担当者と一緒にすばらしいランチを何度も楽しんだエグゼクティブと、ベンダー認定に時間とコストを費やしたエンジニアでは、求めるプロジェクトが異なる可能性があります。

製品間にわずかであっても本質的な違いが存在する場合があります。あるベンダーのコードには企業が望まない不要な付加機能が詰まっていることがあります。それでも保守と更新が必要です。あるベンダーのハードウェア仕様は理想的かもしれませんが、ネットワークOSはバグが多い可能性があります。ソフトウェアに重要なプロトコルを実装する仕組みを機能させるために、エンジニアに余計な負担がかかる場合があります。

結果として、実際のビジネスアプリケーションのニーズを、製品の傾向や制約の範囲内に収まるように調整しなければなりません。つまり、運用が複雑化し、問題のリスクが増大することになります。ネットワークで新しいアプリケーションやサービスがサポートされるペースが鈍化して、ネットワークがボトルネックになります。

より良い習慣は、ビジネスの成果を第一に考慮し、その成果を中心にネットワークを設計することです。Apstraは、インテントにフォーカスすることで成果に優先順位を付けます。そしてデバイスレベルで機能し、宣言によって成果を実装します。ネットワークの状態を継続的に監視および検証して、成果が確実に達成されるように処理します。

Apstraは幅広いネットワークハードウェアおよびソフトウェアをサポートしているため、企業はビジネスに最適な製品を選択できます。ベンダーの傾向によって設計を決める必要はありません。

## 2. 信頼性と反復性に優れた自動化を実行する

ネットワーク業界では数十年にわたり自動化が話題になっていましたが、企業はネットワークエンジニアがポケットナイフのような多くの機能を持つ自社のスクリプトを使い続けています。

実は、企業のほとんどのデータセンターは脆弱なため、複数のデバイスとサービスがオーケストレーションされる広範で信頼性の高い自動化はサポートできないのです。理由の1つは、自動化プロセスにより、ネットワーク全体をダウンさせる想定外の一連のイベントが発生する可能性があることです。さらに、セットアップ後、構成がベースラインから逸脱することも珍しくありません。スクリプトとプレイブックを記述するエンジニアは、ネットワークデバイス上で実行される各種ソフトウェアバージョンの微妙な違いを考慮するために、絶えずコードを調整しなければなりません。

また、ネットワークエンジニアは、広範囲にわたる自動化をサポートするための重要な要素を与えていません。たとえば、ネットワークの状態の高度な可視性や、期待されるデバイス構成に関する信頼できる単一の情報源、本番環境にプッシュする前に変更をテストする機能、プロセスの成果を検証する仕組みなどです。

Apstraを導入することで、信頼性が高く反復可能な自動化が可能になります。Apstraは、想定外の問題を防いで変更内容をオペレーターのインテントに一致させるガードレールの役割を果たします。たとえば、インテントに一致しない構成変更をエンジニアが試みた場合、それがコマンドのタイプミスであるか、既存のポリシーに違反する構成変更であるかにかかわらず、Apstraはエンジニアに通知して変更を防ぎます。

Apstraではネットワーク全体の状態がグラフデータベース（以下参照）に保持されるため、潜在的な問題を本番環境へのプッシュ前に特定できます。このように、Apstraでは変更管理が合理化されます。一般的な変更構成は予測可能性のある検証済みの方法で実行し、想定外の結果を防ぐことができます。

変更により問題が発生する場合に備えて、Apstraにはデバイスを既知の良好な状態にロールバックする機能もあります。そのため、信頼性が高く反復可能な自動化が可能となります。

## 3.1 回学んだら何度でも使用する

ビジネスアプリケーションを時折使用するユーザーは皆、タスクを完了するためにアプリケーションを再度使用するたびに時間をかけて機能に慣れなくてはなりません。

管理、監視、自動化のツールにも同じことが言えます。これらのツールが通常のワークフローに含まれていない場合、インターフェイスの操作に手間取り、タスクに遅れが生じます。ツールの価値を最大限に活用するためにツールを熟知することは良い習慣です。

Apstraは数日で習得して使用できる、わかりやすいツールです。他のソリューションはアーキテクトと運用が大きく変わるため習得に数か月かかることがあります。それらと比べて性能はまったく遜色ありません。また、Apstraは監視、構成、分析などの機能が豊富なため、ネットワークエンジニアはソフトウェアを短期間で習得して定期的に使用し、有用なツールをマスターできます。

同時に、Apstraはマルチベンダー環境で実行できるため、各NOSとベンダーのプロトコル実装の傾向が抽象化されます。つまり、ネットワークエンジニアは作業を完了するために各NOSの微妙な違いを熟知する必要はありません。

## 4. 適切な機器を選択する

機器の特定部分の機能に運用を合わせるのではなく、目の前の仕事に適した機器を選択するのは賢い習慣です。

Apstraのマルチベンダー戦略により、ネットワークチームは企業の目標をサポートするのに最適なハードウェアとソフトウェアを柔軟に選択できます。また、企業は購買力を活用して、ベンダーロックインを回避できることになります。さらに、さまざまなサプライヤから機器を調達できます。現在のサプライチェーンの制約を踏まえると、このような対応が必要になる場合があります。

## 5. 作業を進めながら検証する

テクノロジー関連のソーシャルメディアには、急いで実行したCLIコマンドや、設定ミス、タイプミスから刺激的でやりがいのあるさまざまな結果が生じたというエピソードがあふれています。実際、人間はミスをします。そのため、優秀なエンジニアは本番環境で運用しながら作業をチェックします。

チェックの方法は多様です。一部のネットワークOSは、コマンド入力時に文字列の各ワードをチェックし、タイプミスエンジニアに通知します。エンジニアは、チームメイトに構成のサニティーチェックを依頼したり、スクリプトをリポジトリにアップロードして他のエンジニアにレビューしてもらう場合があります。ITILベースの変更管理プロセスを利用している企業もあります。

Apstraは、デバイスの変更を継続的に検証し、その変更をインテントと一致させることで、この良い習慣を適用します。たとえば、新たに作成したVLANを、タグのない既存のVLANインターフェイスに間違っただけで追加した場合、Apstraはエラーを返すだけでなくその理由を示します。

継続的な検証により、人為的ミスを制限すると同時に、新たな変更から望ましいインテントがもたらされます。つまり、新たな変更は、ビジネスで定義された到達可能性、パフォーマンス、セキュリティ、コンプライアンスに関するパラメーターにうまく適合します。

## 6. 後手に回らずプロアクティブに行動する

小さな問題に、大きくなる前に対処することは良い習慣です。たとえば、スイッチポートが断続的にダウンする場合、根本的原因を見つけて、ポートが完全に故障する前に対処します。

早急に対処しなければ、問題を解明しようとする間、大量のアラートとログ（そしておそらくは怒りのメール）を受け取ることになります。問題がケーブルの緩みであれば幸いです。ネットワークカードや光モジュールが故障した場合は、スペアを用意しておけばよかったと思われません。

Apstraは分析機能によりプロアクティブな運用をサポートします。物理デバイスの状態を監視し、すぐに対処しなければ大きな問題に発展しそうな異常を警告する機能があります。また、帯域幅の問題、使用率、全体の容量を監視し、エンジニアに事前に知らせるため、エンジニアはニーズの拡大にスムーズに効率良く対応でき、苦境に陥らずに済みます。

## 7. 作業をドキュメント化する

野菜を食べて定期的に運動することは良い習慣です。これと同様に、ネットワーク図の更新や構成変更への注釈付けなど、作業をドキュメント化することが良い習慣であることは誰もが知っています。ドキュメントは、何が行われたか、なぜ行われたかを説明するものです。何が起こったかについての記録を提供するものです。この記録はその後の新たな変更、トラブルシューティング、監査などに役立ちます。

エンジニアは異動になったり転職するため、ドキュメントは重要です。エンジニアは異動するとき、運用と企業に関する多くの知識を携えていきます。この知識が記録されているドキュメントを企業で保管して、オペレーター間で共有できます。

問題は、作業のドキュメント化は退屈で時間がかかることです。難しい新規導入への対処時や、危機対応時には、後回しにされることが多いでしょう。今回リストアップした習慣の中で、信頼性の高い定期的なドキュメント化は継続するのが最も難しいかもしれません。

Apstraは、実質的に自動ドキュメント化を実行する設計になっており、この習慣をサポートできます。また、定期的に個々のデバイスからテレメトリを収集し、ネットワーク全体の状態を維持します。Apstraは変更内容とバージョン履歴を保存し、一種のタイムマシンの役割を果たします。チームメイトは過去に遡り、どのような変更や更新が行われたか、どのような成果があったかを確認できます。

前述のように、Apstraは必要に応じてネットワークを既知の良好な状態にロールバックできます。

# ApstraとIBNの概要

ApstraはIBN（インテントベースネットワーク構築）のコンセプトで構築されています。Apstraは、ビジネスレベルのインテント、つまり期待される成果を第一に考慮し、成果の達成に必要なデバイスレベルの構成に変換します。

ビジネスインテントには全般的な成果が含まれます。たとえば、アプリケーションのセットに適した特定のサービスレベルの確保です。また、固有の成果も含まれます。たとえば、ファブリックの導入、VLANの稼働と正しいポートの接続、アクセスポリシーの適用です。

Apstraはネットワークの全体的な状態と個々のネットワークデバイスの状態を継続的に監視します。そのため、ネットワークは常に企業のインテントに一致した状態となります。パケットの損失や混雑、インターフェイスの問題など、企業のインテントに違反するイベントが発生した場合、Apstraはエンジニアに通知し、問題に関連する詳細を提供します。ケースにより、Apstraは次のステップを提案することも、必要に応じて問題を自動修復することもできます。

既存のインテントと競合する構成変更をエンジニアが試みた場合、Apstraは想定外の結果を回避するために変更前の分析を提供することで、エンジニアに通知します。新しいネットワークサービスがデータセンターに追加されると、Apstraはそのサービスをサポートするために必要なネットワーク構成が既存のインテントと競合しないように処理します。

## コア原則

IBNを有効にするために、Apstraはリファレンスデザイン、マルチベンダーサポート、継続的な検証と統合などの基本原則に基づいて構築されています。

**リファレンスデザイン**：データセンターの複雑さを招く重大な原因は、データセンターネットワークが長期的な戦略なしに統合される場合が多いことです。緊急のビジネス要件を満たすために、カスタマイズした専用の実装が組み込まれます。その結果、時間が経つにつれ、自動化が難しく、時間のかかる手作業での手入れや保守が必要な複雑な構成となってしまうのです。このようなネットワークについて、エンジニア間で伝承される企業の知識が記録されたドキュメントはほとんど、あるいはまったく存在しません。技術的負債が累積した結果、エンジニアはビジネスを継続させるためだけの応急処置に頼らざるを得なくなります。

Apstraでは、限られた数のリファレンスデザインの使用により、クリーンな状態から開始できます。リファレンスデザインには、スイッチングハードウェアとケーブリングを含む物理インフラストラクチャのテンプレートが記載されています。代表的なリファレンスデザインはClosネットワークのリーフスパインです。

リファレンスデザインやテンプレートは制約が多いと感じるエンジニアがいるかもしれませんが、このアプローチは業界のベストプラクティスに準拠しています。一般的なデータセンターネットワークに悪影響を及ぼす、1回限りの応急的な個別の実装から生じる問題を大幅に軽減します。信頼できる単一の情報源と変更に対するコントロールを確保するためには、このレベルの規範と厳格な基準に従う必要があります。

Apstraは、「設計、導入、運用」のライフサイクル全体を自動化するために、ブループリントを使用することで、設定をコードで行うことを可能にし、バージョン管理の実装、依存関係の検証を実現しています。このアプローチはクラウドプロバイダの実証済み自動化技術の多くを企業にもたらし、社内多くの専門家の関与は不要となります。

**マルチベンダーサポート**：Apstraは、Cisco、Artista、Dell、ジュニパーをはじめ、ホワイトボックスハードウェアメーカーなどのハードウェアベンダーをサポートしており、その数は増え続けています。また、ApstraはさまざまなネットワークOSをサポートしています。商用NOSから、SONiCなどのオープンソースオプションまで、多岐にわたります。

このマルチベンダーサポートにより、企業はビジネスニーズと技術的ニーズに合致する適切なデバイスを選択し、企業のエンジニアの知識とスキルセットに合ったデバイスを使用できます。また、お客様にとっても選択肢が広がり、自由度が増します。価格を抑え、サプライチェーンの多様性を考慮し、運用面の要件を満たすといった目的で、ラックやポッドに複数のベンダー製品とNOSを混在させることができます。

**クローズドループ検証**：Apstraは信頼できる単一の情報源に基づくクローズドループ検証を通してインテントを確保します。クローズドループ検証では、変更や更新が実際に行われたこと、およびその変更や更新がインテントに一致していることを確認します。単純な自動化スクリプトとは対照的に、クローズドループ検証ではコマンドのセットを自動化できます。単純な自動化スクリプトは実行されて停止するのみで、変更が行われたかどうか、また変更によって望ましい成果が得られたかどうかに関するコンテキストは含まれていません。

クローズドループ検証は、信頼性の高い自動化には不可欠です。なぜなら、この検証は、自動化されたプロセスによってビジネス要件に合致する成果が実際にもたらされたか、またネットワークが望ましい状態で継続して運用されるかをチェックします。この検証により信頼性が実現し、ネットワークチームに自動化システムに対する信頼感が生まれます。

**統合**：ApstraはVMware NSX-TおよびVMware vSphereと緊密に統合されており、サーバーチームとネットワークチームの間で運用を合理化します。Apstraは、物理アンダーレイと仮想オーバーレイの両方をエンドツーエンドに可視化するだけでなく、仮想オーバーレイでの変更により物理アンダーレイでの調整が必要となる場合を特定します。たとえば、ファブリックで設定済みのMTUサイズがオーバーレイの要件に合致しない場合などです。また、Apstraは異常と設定ミスを検知して修復します。

このようにして、アンダーレイはワークロードの接続とボリュームのダイナミックな変化に反応し、最適化されます。さらに、Apstraソフトウェアには、問題が発生した場合にアンダーレイに根本的原因があるのか仮想オーバーレイの問題なのかをすばやく特定する機能があります。Apstraによって問題への対処に必要な時間が短縮されるため、企業は平均修復時間や運用コストを大幅に削減できます。

また、オープンAPIはServiceNow、チャットボット、Slackなどの一般的なワークフローツールとの統合をサポートしています。

## Apstraのコンポーネント

Apstra製品は、3つのソフトウェア要素で構成されています。デバイスエージェント、データストア、グラフデータベースです。これらの要素が連携してインテントベースネットワーク構築を実現します。それぞれについて簡単に説明します。

**エージェント**：Apstraでは、ネットワークデバイスが物理デバイスか仮想デバイスかにかかわらず、デバイス上のデバイスエージェントを使用してデバイスを構成し、テレメトリデータをApstraのデータストアに送信します。サードパーティエージェントを実行できないネットワークデバイスの場合は、オフボックスエージェントをApstraサーバーのLinuxコンテナとして実行できます。オフボックスエージェントはネットワークデバイスから状態に関する情報をSSHまたはAPI経由で取得できます。

**データストア**：データストアはサーバー上で実行され、エージェントテレメトリ、ネットワーク設計の詳細、ネットワークの異常などのデータを収集します。ユーザーインテントもデータストアに保持されます。グラフデータベースも同様に保持されます。

**グラフデータベース**：Apstraでは、データセンターネットワークの各要素や各オブジェクトとすべてのネットワーク構成をグラフデータベース内に表現します。このようにして、Apstraはネットワークの状態を把握します。グラフデータベースはネットワークの仮想モデルとして使用でき、さまざまなチームにとって信頼できる単一の情報源の役割を果たします。ネットワークとデバイスのテレメトリが継続的に入力されて、このモデルはリアルタイムで更新されます。このモデルとユーザーインテントが比較され、ネットワークとデバイスの構成が適切な成果をもたらすように処理されます。