



ホワイトペーパー

# コードの活用によって ITインフラへの投資が伸びている理由

クラウド対応インフラによる差別化と価値創出

## クラウド対応インフラ：継続的イノベーションの原動力

今日の組織を取り巻く状況が日々激しく変化していることはすでに周知の事実ですが、その最先端に立っているのは新興の先進テクノロジー企業です。これらの企業は、ソフトウェアを常に Web スケールでリリースおよび更新しており、大きな成功を収めています。多くの場合、全社を挙げて DevOps フレームワークを導入したことが成功に結び付いているようです。DevOps を導入すると、継続的な革新、迅速なソフトウェア開発、ソフトウェアベースの機能や製品に対する頻度の高い更新が可能になります。企業の成功は、組織の基盤であるデータセンターのインフラとプロセスが DevOps を支える優れたしくみであるかどうかにかかっています。

従来の企業やこれからの企業は、この成功に続くことができるでしょうか。トップダウンからデータセンター インフラのレベルにいたるまで、この取り組みをサポートするためには何が必要でしょうか。現在、ほとんどの企業では、新規顧客を引き付け、既存顧客を維持し、他社をリードするために、ソフトウェア ソリューションを重要な差別化要因として位置付けています。この状況を、「未来を拓くためにテクノロジー企業になる必要はない。... 今や、好むと好まざるとにかかわらず、すべての企業はテクノロジー企業だと言えるからだ」<sup>1</sup> と表す評論家もいます。

企業は、アプリケーションの開発や導入において従来のフレームワークが役割を果たせなくなっていることも認識しています。今注目されているのは、迅速なマイクロサービスの導入を実現する新しい DevOps モデルです。多くの企業が、人事からデータセンターにいたるまで幅広い分野に独自の DevOps 手法を組み込もうとしています。独自のプロセスとデータセンター インフラを採用し、テクノロジーや新たな革新的ソフトウェアソリューションの開発を推進しているのです。

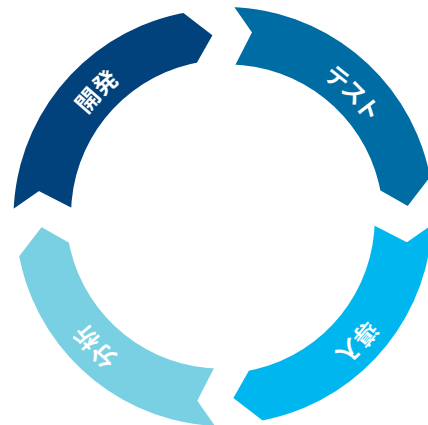
大局的に見れば、この事実どこにも問題はありません。しかし、ミクロな視点で見たとき、この変革は何を意味するのでしょうか。この革新志向の DevOps を、データセンター インフラに対する実際の行動や投資に、どのように落とし込めばよいのでしょうか。

このホワイトペーパーでは、その具体的な手順と、次世代データセンターを実現するまでに企業がたどる段階について説明します。また、このような即応性、行動力、革新性に優れた新興企業に影響を与えている動向についてもいくつかご紹介します。

### データセンターの新たな革新の「エンジン」となるもの

クラウド対応インフラでは、新しいアイデアや高品質なソフトウェアを迅速かつシームレスに創造、テストし、急速に増え続けているお客様に向けて提供、導入して、競争力の強化につなげることができます。この新しいしくみに必要なのは、効果的な DevOps プロセスと、開発チームのニーズを運用チームやテスト チームのニーズと調整できるデータセンターの基本インフラです。新しい次世代データセンターを構築できれば、高品質なソフトウェアや新しい機能の継続的開発、継続的テスト、継続的統合、継続的導入を促進するスムーズなワークフローが実現します。

インフラの視点で見ると、これは本番環境と同品質の開発環境やテスト環境を自動化し、短時間で環境を構築および解体できるしくみであると言えます。あるいは、ネイティブ API やシームレスな統合機能を使用して、基本インフラ コンポーネントを円滑にプロビジョニング、変更、監視できる、拡張可能なインフラとも言えます。開発チームとテスト チームがコードを介してインフラ コンポーネントを容易に利用できる環境です。そのために用いるのは、基本的な定義ファイル、API 呼び出し、インフラ プラグインです。これらは、Kubernetes、Ansible、Jenkins、JFrog、VMware、OpenStack、Splunk など、チームが使い慣れている DevOps 用のツール、フレームワーク、プラットフォームに統合され、簡単に利用できます。



運用チームは、時間のかかるリソースのプロビジョニングや手動によるストレージ管理から解放されます。自動化を活用することで、エコシステムの監視に集中し、企業の革新を加速させるのに適したインフラ ツール、統合、自動化ポリシーを使用できるようになります。

このような新しいデータセンター インフラは、現在の貴社の状況にどれだけ近いものなのでしょうか。このしくみを実現するには、何が必要でしょうか。

### DevOps の普及

Damon Edwards 氏は著書『Short History of DevOps』の中で、「DevOps こそが、Twitter のハッシュタグ (#devops) から始まり、そのハッシュタグを中心として Twitter で展開されていく最初のテクノロジー革新となるかもしれない」<sup>3</sup> と述べています。DevOps という用語は、それが使われる状況に応じて、草の根運動から理念、劇的な革新、文化の変化、ソフトウェア フレームワーク、一連の処理やツールにいたるまで、幅広い意味を表します。

「拡張性とは、  
あるテクノロジーにおいて  
既存の構造に新しい要素や  
機能をどの程度  
追加できるかを表す尺度。  
たとえば、あるソフトウェア  
プログラムで、アドオンや  
プラグインを使用して  
操作内容を強化できる場合、  
そのプログラムには  
拡張性があるとみなされる」<sup>2</sup>

出典：Techopedia.com

ソフトウェア開発ライフサイクル（SDLC）においては、これまで、時間がかかりがちな「ウォーターフォール」型のソフトウェア開発 / 導入手法が広く使用されてきましたが、DevOps はこれとはまったく対照的なアプローチです。DevOps は、SDLC の主要事項（開発、テスト、導入、分析）に対応しながらも、開発、テスト、運用のチームの間で発生しがちな障壁や手動処理による遅延を解消し、このサイクルを大幅に高速化します。DevOps を実践することで、企業内の複数の部門にまたがるプロジェクトやイノベーションの合理化も実現できます。ビジネス上の利点と技術的な利点を兼ね備えていることが、DevOps への移行をいっそう魅力的なものにしています（図 1 を参照）。

今や DevOps の波は大企業にも及びつつあります。典型的な例として Walmart の事例が挙げられます。Walmart では、DevOps の実践と、OpenStack や Puppet などのオープンソースソフトウェアの利用を積極的に進めてきました。この巨大小売チェーンでは、これらのアプローチやフレームワークを利用し、サーバ構築プロセスの自動化や、クラウド間のアプリケーション移動の合理化などを実現して、インフラ管理に役立っています。<sup>4</sup>

DevOps はオープンソース型と従来型のどちらのエンタープライズソフトウェアフレームワークとも相性が良いため、革新的な SaaS プロバイダをはじめとして、この手法を採用している急成長中の IT 企業に多大な影響が現れ始めています。Puppet の『2018 State of DevOps Report』<sup>5</sup> で紹介されている 30,000 人のテクニカルプロフェッショナルを対象とした調査によれば、DevOps の原理を用いている好業績企業は、そうでない企業に比べてコードの導入頻度が 46 倍に達しています。また、コミットから導入までのリードタイムが 2,555 倍高速化し、障害の発生率も 7 分の 1 に減っています。

DevOps にどのような定義を当てはめるにせよ、その影響を見過すことはできません。この事実は現在だけでなく将来のインフラのあり方をも左右し始めています。

基本インフラがどれほどシームレスかつ透過的に DevOps 業務を支援できるかが、企業にとって今後ますます重要な意味を持つことになっていくでしょう。この点はイノベーションを加速して新たな市場ニーズにすばやく対応するうえでも重要になります。

IT インフラや運用の動向を研究する Forrester のアナリスト、Robert Stroud 氏はブログの中で、コードを介してインフラを使用する動きは今後ますます加速していくだろうと示唆しています。同氏は多くの企業のクラウド戦略に関連して次のように述べています。「クラウドファーストへ移行するには、まずインフラの提供、管理、保守の方法を変革し、再利用可能なソフトウェアコンポーネントとしてインフラの提供や利用をサポートできる体制を確立することが必要不可欠です。仮想インフラか物理インフラかを問わず、そのようなインフラは構築や導入に長い時間をかけることなく、いつでも利用できます」<sup>7</sup>

DevOps の普及は新しい現実を呼び起こしつつあります。新たな標準として、コードを介したインフラの利用に対するニーズが広がりを見せるとともに、コードベースのフレームワークを利用した基本インフラコンポーネントの自動化、管理、変更、サポートを求める声も高まりつつあります。



#### 技術的な 利点

ソフトウェアを  
継続的に提供

問題の複雑さを軽減

可視性が向上

問題の解決時間が  
短縮

作業のやり直しや  
予定外の作業が減少



#### ビジネス上の メリット

機能の提供を高速化

運用環境がより安定

リソースの利用効率が  
向上

価値の創出に費やす  
時間が増大

問題の解決や  
メンテナンスに費やす  
時間が減少

システムの成果に対する  
可視性が向上

図1) DevOpsのメリット

「その場で迅速に反復や調整を行えることには、計り知れない価値があります。3 カ月も前の決定事項に縛られることはないのです」<sup>6</sup>

—Polaris Alpha 情報システム シニア バイス プレジデント David Coker 氏

それに伴い、今後の開発では、仮想マシンやコンテナ環境に必要なストレージ ポリユームの作成やマウントといった作業を行うためにコードを使用するようになって予想されます。また、テスト担当者は、コードを介してテスト ベッド環境のプロビジョニングや構築をシームレスかつ迅速に行えるようになり、ストレージ使用量をほとんど増やすことなく、本番環境と変わらない品質の最新データをテストに利用できるようになるものと思われます。IT に対する新たな需要に対応できるかどうかは、リソースをリアルタイムかつオンデマンドで提供できるかどうかにかかっています。同様に、コードを介してインフラ コンポーネントの呼び出しやプロビジョニングをシームレスに行えることも重要です。

このような変革を進めるにあたり、運用チームやインフラ チームは次のような質問を自らに投げかけるとともに、インフラ ベンダーにも問いつけ合わせる必要があります。

- 現在の基本インフラはどの程度の拡張性を備えているか？
- 単にアプリケーションを提供したりデータを保存したりするだけでなく、それ以上のことを自社のインフラで行えるようにするにはどうすればよい？ また、DevOps にとっての価値、ひいてはビジネス全体にとっての価値を自社のインフラからより多く引き出すにはどうすればよい？
- 現在のインフラは、社内で使用されているツールやフレームワークとどの程度まで緊密に統合されているか？
- ソフトウェア コードをベースとした手法（API、プラグイン、ネイティブ統合など）を活用することで、インフラ コンポーネントの呼び出しや使用をどの程度まで簡易化できるか？
- 現在のインフラはニーズや状況の変化にどの程度すばやく対応できるか？

これらの質問に答えることによって、リソースのプロビジョニングに数週間かかる場合と数分で完了できる場合の違い、自動化された直感的なワークフローを導入した場合と複雑で手間のかかる手動処理を継続した場合の違い、さらには、導入が短時間でスムーズに完了する場合と煩雑な引き渡しや導入の遅延を伴う場合の違いが見えてきます。また、単にアプリケーションのニーズを満たすことと、ビジネスに真の価値をもたらすことの違いも明確になります。

南アフリカに拠点を置くプロバイダ Internet Solutions は NetApp® SolidFire® ストレージ インフラを導入したことで、OpenStack および VMware との統合を容易に実現しました。このインフラを通じて、継続的統合 / 継続的導入（CI / CD）ワークフローによる完全自動化が可能になりました。また、従来のアーキテクチャよりも最大 15 倍早く DevOps サービスや仮想マシンをコンテナとともに展開できるようになりました。

「SolidFire と OpenStack の統合は驚異的です。統合と拡張性の2つの点から、この分野で絶大な価値を提供しています」<sup>8</sup>

—Internet Solutions テクノロジー担当ディレクター Kervin Pillay 氏

#### DevOps：インフラに関する既知のアイデアや機能に新たな視点

データセンター インフラの管理に携わる企業は「システムの常時稼働を維持」するために必要な作業には慣れています。そのため、データセンター アプリケーションをサポートするコンピューティング リソース、ネットワーク リソース、ストレージ リソースの管理にあたっては、リスクを避けつつシステムの可用性、パフォーマンス、信頼性を確保することに主に重点を置いています。ストレージ システムに関しては、データ損失やダウンタイムを最小限に抑えるとともに、復旧までの時間を短縮できるようすることも重要なポイントになります。ポイントインタイム（PiT）コピーと呼ばれるデータセットのスナップショットを瞬時に作成することは、このようなユースケースの好例です。

このようなスナップショットはすばやく復元できるため、IT ディザスタリカバリ プランにおける重要な要素として多くの企業で採用されています。

その一方で、DevOps の普及と、それにより加速されるコードを介したインフラ利用に対するニーズの高まりによって、スナップショットやクローニングといったインフラ機能が別の観点から改めて脚光を浴び始めています。これらのテクノロジーや機能は、次世代データセンターにおいて上記以外にどのような価値を企業にもたらすのでしょうか。





開発者やテスト担当者、QA 担当者がスナップショットやクローンなどを直接呼び出せるようになったら、DevOps にはどのようなメリットがもたらされるでしょうか。これらの機能を使用することで、本番環境と同じ品質のデータにすばやくアクセス、更新したり、必要に応じて新しいボリュームを自分で追加したりできるようになれば、どのような利点が生まれるのでしょうか。

これらの機能は、何カ月もの間、多数のチーム リソースを費やして環境を構築しなくても、既存の DevOps ツールやフレームワークに含まれるインフラ統合機能を通じて直接利用できます。また、プロビジョニングのリード タイムが長いために古くなってしまったデータセットを使用して作業する代わりに、リアルタイムかつオンデマンドでデータセットを入手して、コードの品質チェックをプロセスの早い段階に移すことが可能になります。そのために必要となるのは、適切な種類の拡張可能なインフラと、それを活用するためのノウハウのみです。

インフラを必要に応じてシームレスに利用できるようになれば、開発、テスト、導入の各段階で時間を大幅に短縮でき、チーム間での手作業による煩雑な引き渡しを減らすことができます。

しかし、これは次世代データセンターがもたらす恩恵のごく一部にすぎません。

「複製した銀行環境を  
いつでも 10 分以内に  
誰にでも提供できます。  
開発者はボタンを  
クリックするだけで、  
必要なデータ、サーバ、  
アプリケーション、構成、  
テストなど、環境に  
付随するものをすべて  
入手できます。どれも  
検証済みで、すぐに運用を  
開始できるよう完全に  
パッケージ化されています。  
これらが揃っていれば、

どのようなテストでも  
思いどおりに実行できます。  
見事としか言いようが  
ありません」<sup>9</sup>

—ING DIRECT Australia IT 戦略担当部長 Ben Issa 氏

#### DevOps 手法とインフラ利用の成熟

本書の冒頭で触れたような新たな現実のもとでは、DevOps の手法を最大限に実践することで、イノベーションを加速できます。継続的開発と継続的導入を促進するスムーズなワークフローの中で、インフラをシームレスに変革することができますはずです。

しかし、DevOps の成功の度合いにおいても、新たな DevOps ツールの検証の進み具合においても、動的なインフラを自動で利用できる環境の整備においても、まだその段階までたどり着いていない企業が多数あるのが現状です。また、ほとんどの企業は進化の過程で、さまざまな領域に異なるタイミングで重点的に取り組まなければならない現実にも直面します。

DevOps に関して言えば、ビジネス上の優先事項や強みと弱みによって企業ごとに重点の置き方が変化します。たとえば、コードの品質改善やテストの高速化に重点を置く企業もあれば、システムの可視性を高めて環境内の状況を把握することに重点を置く企業もあるでしょう。いち早くバグを特定して必要な修正作業を見極めることを重視する企業や、導入回数をせめて初期の段階でだけでも減らすことができれば満足と感じる企業、1 日に何度も新機能を導入したい企業もあるでしょう。また、企業の優先事項はさまざまな制約やニーズの変化に応じて、時間とともに変化していく傾向にあります。しかし、DevOps のフレームワークを支えるために必要な 6 つのコア機能は変わりません。そのコア機能とは、コンテナ、構成管理、コード / バイナリ管理、継続的統合 / 継続的導入 (CI / CD)、クラウド / プラットフォーム サービス (PaaS)、分析です。

インフラ チームや運用チームは、企業ごとの重点の置き方や目標の違いに影響を受けずに、目指す段階へ進むことができます。その取り組みの中で、多くの企業はインフラの成熟に向けていくつかの主要な段階を経ていくことになります (図 2 を参照)。

多種多様なエンタープライズ環境や最先端の SaaS 企業との連携を通じてネットアップが得た知見によれば、次世代データセンターへの移行を進めるにあたって、大半の企業はインフラの成熟に向けて 4 つの段階を一樣にたどることが明らかになっています。

大半のインフラでは統合や仮想化といったいくつかの初期段階がすでに完了していて、中にはスクリプト化したタスクを使用してプロビジョニングやレポート作成などの基本的インフラ タスクを自動化し始めている事例も見られます。

自動化が進むにつれ、企業はインフラを通じてどのようなことを実現したいと考えるようになるのでしょうか。4 つの段階について説明していきます。

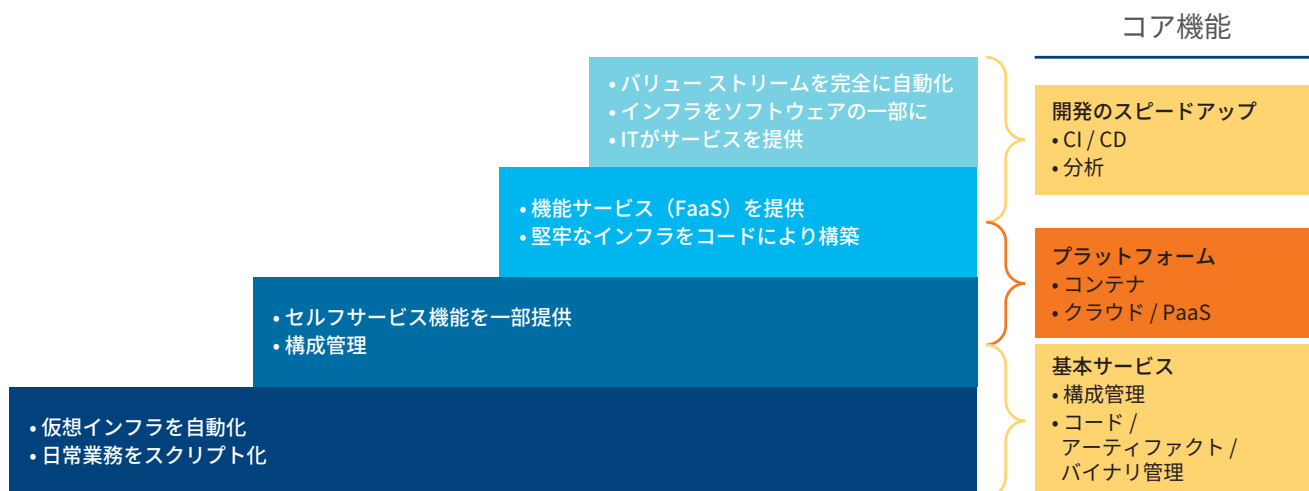


図2) 成熟への道

### 第1段階：インフラ ベースの API 統合に向けた取り組みの開始

この段階では、コードを介して利用できるインフラ機能で達成したい目的が認識されます。ここでは、インフラの統合を重視している DevOps チームのメンバーが、拡張可能なインフラの特徴や機能を見定め、自社の環境や優先課題にとって有益な API 統合の可能性を検討し始めます。その際、通常は、開発者が好んで利用しているツールセットとの統合がどの程度まで可能かをインフラ ベンダーに問い合わせることになります。この後、チームは DevOps を推し進めていく中で、そのようなコンポーネントの導入に期待を寄せるようになります。

例としては、インフラのコンソールや GUI ではなく、コードを通じてインフラの機能を利用できるようにするケースなどが挙げられます。その際には、一般的なツールセットとのネイティブな統合を活用するのが理想的です。ネイティブ統合を利用すると、必要なコードの量を最小限に抑えることができます。また、別の例として、REST プロトコルを使用してインフラの機能にアクセスするケースが挙げられます。REST プロトコルを使用できれば、ネイティブ統合機能が用意されていない場合でも、任意の DevOps ツールやフレームワークにインフラの機能を統合できます。

Premier Eye Care では開発環境をわずか数分で更新できるようになっただけでなく、前日夜のバックアップを使用する代わりに NetApp SolidFire アレイに格納された最新データを使用して開発環境を更新できるようになりました。この機能を実現するため、ネットアップは Premier とサービス パートナー Wirestorm との間を仲介して Windows PowerShell スクリプトを作成しました。Premier では、このスクリプトを使用して SolidFire アレイの Snapshot™ コピーを自動化し、それらのバックアップを開発環境で利用しています。

「私たちの働き方は完全に変わりました。ボタンを押すだけで、その瞬間まで本番環境で使用されていたデータを SolidFire から取得できるようになったからです。社内ユーザから金曜日の夕方 5 時に開発環境の更新を依頼されたとしても、今ではわずか数分で更新を完了して帰宅できます。しかも、ユーザが利用できるデータは前日夜のものでありません」<sup>11</sup>

—Premier Eye Care データベース管理者 Rob Connock 氏

## 第2段階：引き渡し業務の削減とさらなる自動化の推進

この段階で、ある程度の自動化は完了していますが、インフラの成熟に向かう第2段階では、ソフトウェアの開発 / テスト / 導入サイクルから人員や手作業による引き渡しをさらに減らせるように開発者やインフラ運用チームが取り組んでいく必要があります。チケットの提出や別チームへの要求が伴いがちなワークフローを自動化することが求められます。単純な手作業は自動化して、たとえばポリシー ガイドラインを自動化したり、インフラと開発者向けツールをプラグインで接続したりするほうがよいでしょう。

例としては、アプリケーション所有者や開発者が、定義済みの運用ガイドラインに基づき、Docker Volume Plug-In などを使用して、ストレージ ボリュームを作成し、オペレーティング システムにマウントできるようにするといった方法などが考えられます。

「ストレージ管理を  
合理化して、SolidFire の  
簡易性がもたらす利点を  
DevOps チームが  
活用できるようにすることで、  
ストレージ エンジニアは  
今までよりも有意義な  
活動により多くの時間を  
費やせるようになって  
います。現在、私たちは  
市場での優位性を  
維持できるように  
インフラの整備を  
進めています」<sup>12</sup>

—FICO プラットフォーム運用 / エンジニアリング担当シニア マネージャー  
Donald Talton 氏

## 第3段階：インフラを通じたセルフサービスの構築

この段階では、運用チームが自らの関与の度合いをさらに下げること、利用者に対するセルフサービス型インフラの提供を促進する方法について具体的に検討します。開発 / テスト チームは、この段階で、運用チームやインフラ チームと直接やり取りを行わなくても、必要なデータを必要なときに自分で入手できるようになります。これを実現するには、開発者が日常的に使用するツールが Ansible、Jenkins、Kubernetes、プライベート クラウド システムのいずれであっても、テクノロジーが緊密に統合されているセルフサービス型インフラ システムの構築が必要になることがあります。

そのためには、構成管理ツール（Ansible など）の機能を拡張して、構成管理ツールセット内からストレージなどのインフラ コンポーネントを直接追加したり変更したりできるようにすることが必要になります。

この段階では、管理者、運用チーム、ストレージ チームの役割が変わります。インフラのタスクをすべて自分たちで行う必要はありません。利用者が直接起動したコードによって、タスクは自動で実行されます。運用担当者は、全体の把握、可視化、インフラの管理に集中できます。別のチームからの入力を待ったり、手作業で対応したりする必要はありません。この時点で、セキュリティ、信頼性、可用性を犠牲にすることなく、ポリシーベースの管理や、一般的な DevOps ツールとインフラとのコード ベースの統合を通じて、シームレスなセルフサービスが実現します。

この段階の例として、テスト チームが Jenkins を使用してテスト フレームワークを定義することもできます。そのようなフレームワークでは、1 つのインフラに対して複数のテスト ベッド環境を運用しなければならない場合があります。この形式のセルフサービスに対応するために拡張可能なインフラを使用すれば、テスト環境のプロビジョニングや更新をすばやく行えるため、比較データよりも短時間でテストを完了できます。テストの実行にあたって、インフラ チームへの引き渡しを待つ必要はありません。

その他のセルフサービス型インフラの例として、特定のアプリケーションに関連付けられた永続的ストレージのプロビジョニングや管理を開発者が直接行えるように、Kubernetes Volume Plug-In を使用してアクセス権を付与するケースが挙げられます。これによって、開発者は他チームの支援を受けることなく、VMware vSphere Virtual Volume や Ansible などのテクノロジーを使用して基本インフラ環境を直接変更できるようになります。

「私のチームでは、  
導入システムの数  
1/10 に減らし、  
合計導入時間を  
20 時間から  
1 分以下にまで  
短縮できました」<sup>13</sup>

—Wirestorm エンジニアリング担当バイス プレジデント Jeremy Goodrum 氏

#### 第4段階：ソフトウェア定義インフラ（SDI）フレームワークの導入

開発チームとテストチームが、必要なものを必要なタイミングで利用できるようになり、運用する側やインフラ チームが手作業をする必要がなくなったら、本当のソフトウェア定義インフラを実現する準備が整った状態です。

この段階で問題になるのは、単に特定のハードウェアにソフトウェアをインストールできるかどうかではありません。インフラの安定性、信頼性、可用性、可視性を維持しつつ、必要なときにオンデマンドでインフラを変更して、絶えず変化するアプリケーションの要件に対応するためのポリシーとフレームワークを設定できることが重要になります。

この段階では、開発者が基本インフラ環境の変更を行う場合があります。具体的には、特定のインフラ動作やパフォーマンス特性（特定のアプリケーション ボリュームに関連付けられたパフォーマンス特性など）を実現できるようにポリシーを実装する作業などがあります。VMware vSphere Virtual Volumes や Ansible を使い、必要な変更をこうしたポリシーで導入することもできます。ポリシーベースのフレームワークを採用すれば、柔軟性に優れ、管理と利用が容易なインフラが実現します。

#### 移行の支援

DevOps や次世代データセンターへの移行は、一晩で完了するプロセスではありません。また、そのために必要なスキル、知識、経験は、試行錯誤を繰り返して企業全体で獲得していかなければならないものです。

このホワイトペーパーで説明されている動向、段階、および推奨されるマイルストーンは、企業が進んでいくための指標となるでしょう。DevOps の実現に向けた重要なマイルストーンを達成するうえで、拡張性のあるインフラが重要な役割を果たすということを、早い段階で認識することになるはずです。

DevOps の導入を成功させ、IT をイノベーションと利益創出の原動力とするためには、根本的な流動性を備えたインフラを取り入れ、開発者やテスト チーム、QA チームのニーズの変化に合わせて、必要に応じた拡大縮小や適応ができなくてはなりません。

インフラチームと運用チームが、この移行を成功に導くための鍵です。今後、これらのチームの役割は変わっていく一方で、拡張可能なインフラ コンポーネントの評価、選択、実装、監視、管理を適切に行い、目の前の課題に対処するには、彼らの専門知識が必要不可欠です。

これらのチームは、プラグイン、API、SDK、その他の統合機能、ポリシー自動化機能といった、拡張可能な特定のインフラ機能の利用可否を詳しく調べていくことで、大きな自信を獲得することができま。また、社内のチームやインフラ ベンダーなどと連携し、コードによる呼び出しや利用が可能なセルフサービスとソフトウェア定義インフラへのシームレスな移行について支援を受けることも、慎重に検討する必要があるでしょう。

#### その他のリソース

ソフトウェア開発ライフサイクルおよびネットアップ ソリューションとの統合について詳しくは、[こちら](#)をご覧ください。



## 参考資料

1. 『Evolution Not Revolution: DevOps in the Wild』 Boyd E.Hemphill (2015 年 2 月 4 日)、DevOps.com、<https://devops.com/evolution-not-revolution-devops-wild/>、最終確認日：2017 年 5 月 3 日
2. 『Extensible』 の定義、Techopedia、<https://www.techopedia.com/definition/7107/extensible>、最終確認日：2017 年 4 月 19 日
3. 『The Short History of DevOps』 Damon Edwards、<http://itrevolution.com/the-history-of-devops/>、最終確認日：2017 年 4 月 19 日
4. 『Walmart Boasts 213,000 Cores on OpenStack』 Linda Hardesty、2017 年 2 月 24 日、SDxCentral  
<https://www.sdxcentral.com/articles/news/walmart-boasts-213000-cores-openstack/2017/02/>、最終確認日：2017 年 4 月 19 日
5. 『2018 State of DevOps Report』 Puppet、<https://puppet.com/resources/whitepaper/state-of-devops-report>、最終確認日：2018 年 12 月 7 日
6. 『Better Agility for DevOps Helps Polaris Alpha Accelerate Time to Market』 ネットアップ、<http://www.netapp.com/jp/media/cs-polaris-alpha.pdf>、最終確認日：2018 年 12 月 21 日
7. 『Infrastructure as Code, the Missing Element in the I&O Agenda』 Robert Stroud、2016 年 2 月 9 日、Forrester、  
[http://blogs.forrester.com/robert\\_stroud/16-02-09-infrastructure\\_as\\_code\\_the\\_missing\\_element\\_in\\_the\\_io\\_agenda\\_0](http://blogs.forrester.com/robert_stroud/16-02-09-infrastructure_as_code_the_missing_element_in_the_io_agenda_0)、最終確認日：2017 年 4 月 19 日
8. 『We' ll Never Need to Do a Forklift Upgrade Again』 ネットアップ、<http://www.netapp.com/us/media/cs-internet-solutions.pdf>
9. 『ING DIRECT Innovates Faster with NetApp, Cisco, and Microsoft Private Cloud』 ネットアップ、<https://www.netapp.com/jp/media/ing-direct.pdf>、最終確認日：2018 年 12 月 21 日
10. 『DARZ Docker & Container-as-a-Service Drives Digital Transformation Through DevOps』 お客様の導入事例：サービス プロバイダ DARZ、ネットアップ、  
<http://www.netapp.com/us/media/cs-internet-solutions.pdf>
11. 『Healthcare Company Cures Storage Inefficiencies with SolidFire』 お客様の導入事例：Premier Eye Care、ネットアップ、<http://www.netapp.com/jp/media/cs-premier-eye-care.pdf>
12. 『Accelerated Time-to-Market Through Cloud Services Powered by Flash』 お客様の導入事例：ソフトウェア サービス プロバイダ FICO、<http://www.netapp.com/jp/media/cs-fico.pdf>
13. 『The New Norm for Wirestorm' s DevOps Is Fast, Ultrafast』 お客様の導入事例：サービス プロバイダ Wirestorm、ネットアップ、<http://www.netapp.com/jp/media/cs-wirestorm.pdf>

本ドキュメントに記載されている製品や機能のバージョンがお客様の環境でサポートされるかどうかについては、ネットアップ サポート サイトで [Interoperability Matrix Tool \(IMT\)](#) を参照してください。NetApp IMT には、ネットアップがサポートする構成を構築するために使用できる製品コンポーネントやバージョンが定義されています。サポートの可否は、お客様の実際のインストール環境が公表されている仕様に従っているかどうかによって異なります。

## 著作権に関する情報

Copyright © 2019 NetApp, Inc. All rights reserved. Printed in the U.S. このドキュメントは著作権によって保護されています。著作権所有者の書面による事前承諾がある場合を除き、画像媒体、電子媒体、および写真複写、記録媒体、テープ媒体、電子検索システムへの組み込みを含む機械媒体など、いかなる形式および方法による複製も禁止します。

ネットアップの著作物から派生したソフトウェアは、次に示す使用許諾条項および免責条項の対象となります。

このソフトウェアは、ネットアップによって「現状のまま」提供されています。ネットアップは明示的な保証、または商品性および特定目的に対する適合性の暗示的な保証を含み、かつこれに限定されないいかなる暗示的な保証も行いません。ネットアップは、代替品または代替サービスの調達、使用不能、データ損失、利益損失、業務中断を含み、かつこれに限定されない、このソフトウェアの使用により生じたすべての直接的損害、間接的損害、偶発的損害、特別損害、懲罰的損害、必然的損害の発生に対して、損失の発生の可能性が通知されていたとしても、その発牛理由、根拠とする責任論、契約の有無、厳格責任、不法行為（過失またはそうでない場合を含む）にかかわらず、一切の責任を負いません。

ネットアップは、ここに記載されているすべての製品に対する変更を随時、予告なく行う権利を保有します。ネットアップによる明示的な書面による合意がある場合を除き、ここに記載されている製品の使用により生じる責任および義務に対して、ネットアップは責任を負いません。この製品の使用または購入は、ネットアップの特許権、商標権、または他の知的所有権に基づくライセンスの供与とはみなされません。

このマニュアルに記載されている製品は、1 つ以上の米国特許、その他の国の特許、および出願中の特許によって保護されている場合があります。

本書に含まれるデータは市販品（FAR 2.101 の定義に基づく）に関係し、データの所有権は NetApp, Inc. にあります。米国政府は本データに対し、非独占的かつ移転およびサブライセンス不可で、全世界を対象とする取り消し不能の制限付き使用権を有し、本データの提供の根拠となった米国政府契約に関連し、当該契約の裏付けとする場合にのみ本データを使用できます。前述の場合を除き、NetApp, Inc. の書面による許可を事前 to 得ることなく、本データを使用、開示、転載、改変するほか、上演または展示することはできません。国防総省にかかる米国政府のデータ使用権については、DFARS 252.227-7015(b) 項で定められた権利のみが認められます。

## 商標に関する情報

NetApp、NetApp のロゴ、<http://www.netapp.com/jp/legal/netapptmlist.aspx> に記載されているマークは、NetApp, Inc. の商標です。その他の会社名と製品名は、それを所有する各社の商標である場合があります。

WP-7292-0219-jaJP