

Making Everything Easier!™

HPE特別版

Composable Infrastructure

FOR
DUMMIES®

コンポーザブルインフラを 分かりやすく解説

- 現代の企業が直面する深刻なIT部門の課題を探る
- コンポーザブルインフラストラクチャを使用して新しい価値を素早く、継続的に作成し、提供する
- 御社のインフラストラクチャ環境にアイデアエコノミーの原則を適用する

提供：



**Hewlett Packard
Enterprise**



Scott D. Lowe



HPEの概要

Hewlett Packard Enterpriseは業界をリードするテクノロジー企業でお客様の成長と業務の高速化を実現します。業界で最も包括的なポートフォリオにより、クラウドをデータセンターや職場のアプリケーションまで広める同社のテクノロジーとサービスは、世界中のお客様のIT業務を効率化し、その生産性を高め、より安全なものにします。

Composable Infrastructure FOR **DUMMIES[®]**

HPE特別版

Composable Infrastructure

FOR
DUMMIES®

HPE特別版

by Scott D. Lowe

WILEY

Composable Infrastructure For Dummies®, HPE特別版

版元：

John Wiley & Sons, Inc.

111 River St.

Hoboken, NJ 07030 - 5774

www.wiley.com

著作権 © 2018 by John Wiley & Sons, Inc., Hoboken, New Jersey

1976年アメリカ合衆国著作権法の第107章、108章の下、版元の書面による事前の許可がある場合を除き、本書のいかなる部分も複製してはならず、情報検索システムへの登録や電子、機械、コピー、録音、スキャンなどの形式を含む、いかなる手段での配信も一切認められないものとします。版元に使用許諾を申請する場合は、郵送ではPermissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030宛て、電話では(201) 748-6011、ファックスでは(201) 748-6008まで、またはオンラインでは<http://www.wiley.com/go/permissions>でお問い合わせください。

商標：Wiley、For Dummies、Dummies Manのロゴ、The Dummies Way、Dummies.com、Making Everything Easier、および関連するトレードドレスはアメリカ合衆国、あるいはその他の国に所在するJohn Wiley & Sons, Inc.および/または関連会社の商標または登録商標であり、書面による許可がない限り使用することは認められません。その他の商標はすべて、各商標所有者の財産であり、John Wiley & Sons, Inc.と本書で言及される製品および業者の間には一切の関係がありません。

責任の制限/保証の免責： 版元および著者は、本書の内容に関して、その正確性、完全性、および特定の目的に対する適合性を含み、またこれに限らず、一切の責任を放棄し、保証も一切致しません。また、本書の販売や販促物により保証が適用されたり、その範囲が拡大されるようなことはありません。本書に記載されたアドバイスや戦略は、状況により適切でない場合があります。版元が法律、会計、その他の専門サービスについてアドバイスを提供する業務に従事していないことを読者が承諾していることを前提として本書は販売されています。専門家のアドバイスが必要な場合は、適切な専門家にご相談ください。版元および著者は、本書により生じた損害に対し、一切の責任を負いません。引用および/または詳細な情報源として本書に記載されている企業やウェブサイトに関しては、その企業やウェブサイトが提供または推奨する情報の正否を著者や版元が保証するものではありません。また、本書に記載のウェブサイトが、本書が書かれた時点から読まれるまでの間に、変更または削除される場合があることをご理解ください。

ISBN 978-1-119-47801-0 (pbk); ISBN 978-1-119-47810-2 (ebk)

製作：アメリカ合衆国

10 9 8 7 6 5 4 3 2 1

弊社の他の製品やサービスの一般情報、また御社の事業や部門に合わせた「For Dummies」シリーズの製作については、米国の事業開発部までお電話（877-409-4177）またはメール（info@dummies.biz）にてお問い合わせいただくか、www.wiley.com/go/custompubをご覧ください。「For Dummies」ブランドの商品またはサービスを提供するためのライセンス供与に関する情報は、BrandedRights&Licenses@Wiley.comまでお問い合わせください。

謝辞

本書の出版にあたりご協力いただきました皆様に心より御礼申し上げます。

ディベロップメントエディター：

Elizabeth Kuball

コピーエディター： Elizabeth Kuball

アキジションエディター： Katie Mohr

エディトリアルマネージャー：

Rev Mengle

事業開発担当： Karen Hattan

プロダクションエディター：

Kumar Chellappan

目次

はじめに 1

本書の概要.....	1
対象読者	1
本書で使用するアイコン	1
本書について	2
本書の読み方	2

第1章：アイデアエコノミーの導入 3

従来型インフラストラクチャの落とし穴.....	4
オペレーション主導	5
コストと効率にフォーカス	5
変化が遅い	6
明かりをつけておく	6
現代のビジネスとITドライバー	7
アプリ主導	7
俊敏性にフォーカス	8
利益を生み出す	9
俊敏性、速度とその現実.....	10
従来型ITとアイデアエコノミー	10
何でも稼働できる	12
動作が速い	12
効率よく稼働.....	12
価値を引き出す	13

第2章：ユートピアの達成 15

時間、場所、ワークロードを選ばず適切 なサービスレベルで	15
時間を選ばない	16
場所を選ばない	16
ワークロードを選ばない	17
適切なサービスレベル	18
ビジネスを加速するプログラマブルイン フラストラクチャ	18

第3章：インフラストラクチャの4つのカテゴリ .. 19

従来型インフラストラクチャ	19
コンバージドインフラストラクチャ	20
ハイパーコンバージド インフラストラクチャ	21
コンポーザブルインフラストラクチャ	23

第4章：流動的リソースプール.....25

コンポーザブルインフラストラクチャとは	25
流動的リソースプール.....	26
ワークロード対応.....	27
物理.....	28
仮想.....	29
コンテナベース	29
シンプルな両方向への拡張性.....	30
従来型でありながらクラウドに近い.....	30

第5章：ソフトウェア・デファインド・ インテリジェンス.....31

ソフトウェア・デファインド・データセンター	31
テンプレート主導のインフラストラクチャ	32
摩擦のないIT	33
ハードウェアとソフトウェアを一元化.....	34

第6章：ユニファイドアプリケーションブ ログラミングインターフェイス（API）.....35

API増殖の問題を見つける	35
接着剤としてのコンポーザブル	36
IT：コストセンターか収入源か.....	36
開発者にフォーカス.....	37
ITの変革	38
テストの高速化と自動化.....	38
データセンターにおける統合と自動化.....	39
API主導コンポーザブルの見通し	39

第7章：コンポーザブルインフラスト ラクチャを選ぶ10の理由.....41

はじめに

データセンターは変革の第一歩であり、いわばルネッサンスです。重要なビジネス目標の達成を遅らせるような効率が悪く、柔軟性のないデータセンターシステムに縛られることを厭わない企業などありません。

IT部門は既存のビジネスアプリケーション群を維持しつつ、新しいビジネスモデル「アイデアエコノミー」に対応するように迫られています。

本書の概要

世の中には見た目以上にデータセンターを使用するチャンスがたくさんあります。本書は48ページに渡り、使用できるすべての選択肢を比較し、理解できるようサポートします。本書を読めばコンポーザブルインフラストラクチャがどのように環境に収まるかを理解でき、ビジネスを次の世紀まで推進する上で役立ちます。

対象読者

本書の読者は、データセンターコンピューティング、仮想化、およびストレージの基本を理解していると想定しています。本書はデータセンターアーキテクチャについて学びたいIT業界の方々にピッタリです。また、管理職や役員だけでなく、技術者の皆様にもお楽しみいただけます。

本書で使用するアイコン

本書では、重要なコンセプトをより良く理解し、覚えてもらうために以下のような複数のアイコンを使用しています。



章を読み進める前に一度立ち止まって大切なコンセプトを思い出させるために使用します。



このアイコンを見つけたら、その情報を頭の片隅にしまっておき、後でデータセンター環境を分析する際に思い出せるようにしましょう。



本書で技術用語を詳しく解説することはありませんが、それでも技術的な情報がチラホラ出てきます。このアイコンはそういった情報に使用されます。



後で問題になり得ることに対してはさらなる警戒が必要になる場合があります。本書を通して注意が必要な個所にこのアイコンを使用しています。

本書について

1冊ではすべてを語り切れません。コンポーザブルインフラストラクチャに関する知識を深めたい方は本書を読み終えた後、HPEのウェブサイトでもコンポーザブルに特化した資料をご覧ください。詳しくはwww.hpe.com/jp/synergyをご確認ください。

本書の読み方

他のどの部分を忘れても、これだけは覚えておいてください。いくら費用を節約したところで、ユーザーが合理的な方法で仕事を片付けることができれば、IT部門は無能と見なされます。IT機能をビジネスの原動力に変える方法を見つけ出すのは、先見の目がある技術者とリーダーの役割です。

第1章

アイデアエコノミーの 導入

本章の内容

- ▶ 現代のIT部門が対応すべき2つのオペレーションモデルを確認
- ▶ アイデアエコノミーの達成に新しいIT構想が必要な理由
- ▶ 従来型IT方式が十分でなくなった理由を分析

HPEのCEO、Meg Whitmanは「私たちが住む世界は、競合他社よりも早くアイデアを利益に変える能力により成功が決まるアイデアエコノミーに変わった」と語っています。

IT業界はビジネスを行う上で根本的な変革を迫られています。

怖がらせてしまいましたか？

多くの人にとっては怖いでしょう。しかし、この変革にはちゃんとした理由があり、その多くはアプリケーションとサービスをこれまで以上に素早く提供するというニーズの高まりにまつわるものです。過去10年間にわたり、ITが失敗ばかりしてきたということではありません。むしろ逆で、IT部門はビジネスニーズに応えるため、必死で働いてきました。

ただ、時代が変わり企業も変わりました。私たちを取り巻く環境も変わりました。私たちは変化の時代を生きているのです。

アイデアエコノミーとは

CIOや業務執行役員は、境界を越えてつながった企業の競合優位性を活用することを目的とした大きな変革の最前線にいます。業界調査会社であるIDC社はモバイルデバイス、クラウドサービス、ソーシャルネットワーク、ビッグデータ分析のための新しいITを「第三のプラットフォーム」と呼んでいます。また、Gartner社はこれを「バイモダルIT」と呼んでいます。

HPEはこのインフラストラクチャの進化を「アイデアエコノミー」呼んでいます。なぜなら、これにより、IT部門は利益を出す商品やサービスや経験を素早く提供して、新しい事業機会を推進する能力を得られるからです。IT部門は社内の事業プロセ

スを自動化するテクノロジーを提供するだけでなく、以下を実現するソフトウェアベースのサービスを作り出すことで事業戦略や利益に直接働きかけることができるようになりました。

- ✓ 成長の促進
- ✓ 利益の強化
- ✓ 生産性の向上
- ✓ 革新の増強
- ✓ 組織における俊敏性の向上
- ✓ 顧客体験の改善
- ✓ 競合優位性の提供

この定義を念頭に、ITを進化させる上で直面する課題やチャンスについて説明していきます。

IT部門は現在のデータセンターを構築する上で多大な努力を重ねてきましたが、世界は一変しました。「クラウド」が普及し、あらゆる面で効率を上げています。仮想化も当たり前になりましたが、より大きなワークロードをサポートする段階に来て立往生しています。ビジネスは10年前よりも大幅にテクノロジーに精通しています。

これこそが新しいビジネスの流れ、通称「アイデアエコノミー」であり、これまでとは違う物の見方とインフラストラクチャが要求されます。本章では、その理由を説明します。

従来型インフラストラクチャの落とし穴

ここで残酷ながら動かせない事実を語りましょう。私たちが現在インフラストラクチャの管理に使用しているテクノロジーや

ツールが進化していく中で、オペレーションおよびリソースサイロの複雑さと効率の悪さは依然として残っています。企業はIT部門を最前線に置き、生産性の向上、コストの削減、およびイノベーションにより、全く新しいレベルへ効率性を引き上げ、競合優位性を維持するチャンスを作る必要があります。しかし、IT部門はダウンタイムに耐えられない既存のインフラストラクチャの維持に足を引っ張られています。新旧両方のインフラストラクチャを管理する上で困難が絶えない組織の多くが変化を考え始めるのも当然でしょう。



今日のITの大きな課題は、現代のIT部門という木に成った果実であり、その根は1970年代にまで遡ります。多くのIT部門にとって、使用するハードウェアやソフトウェアは変わりましたが、ビジネスそのものは変わっていないというのが現状です。

信じられませんか？その理由と乗り越える方法は読み進めれば分かります。

オペレーション主導

こんなジョークがあります。

放送係：「何が欲しいんだ？」

群衆：「タイムトラベル！」

放送係：「いつ欲しいんだ？」

群衆：「そんなもの関係ない！」

現代のビジネスが抱える未来にワープしたいというビジネスニーズに応えるのにIT部門は四苦八苦しています。現代のIT部門は、対応しているサービスの種類を厳密に調査し、漠然とした未来ではなく、今日にでも行動を起こせるようにしておかなければならないのです。大抵はデータセンターインフラストラクチャの問題が業務に不可欠なプロジェクトを遅らせてしまい、大いに影響を及ぼす問題を作り出してしまうのです。

コストと効率にフォーカス

簡単に言えば、多くの組織にとってIT部門はコストセンターと見なされ、サポート機能として配置されています。そして他のコストと同様、企業は「わずかなお金で多くのものを得る」をモットーにITコストを削減または完全に排除する機会をうかがっています。

このパラダイムが機能しない時が必ず来ることはすでに周知の事実です。システムから排除可能な非効率要素は限られているため、そのうち重要な要素をカットし始めることになり、サービスの提供が滞り、結果として収益まで下げてしまいます。

このようなケースではしばしば、限りなく少ないスペースにできるだけ押し込むという期待をこめ、データセンターの密度が引き合いに出されます。データセンターの拡張費用は運用予算の面でも資本予算の面でも高くついてしまうからです。仮想化を通して統合することで、より多くのアプリケーションを少ないハードウェア上でホストし、コストを削減するというアプローチが一般的です。

概して、従来のデータセンターとそれに携わるIT部門の目標はコストを排除することなのです。しかし、賢者は言っています。「好景気への道に近道はない」と。

変化が遅い

企業にとって最大の悩みの種となるのは恐らく、データセンターインフラストラクチャの変化の遅さでしょう。企業は改善された新しいサービスを求めており、インフラストラクチャが追いつくのを待っている余裕はありません。

ただし、良い方法があります。IT部門は安定性やデータ保護といった、インフラストラクチャにおける定期的なイノベーションや変化の速さにそぐわない目標掲げることによって予算を得ることもできます。



このシナリオが近々に変わることはないでしょう。何よりも盤石な安定性に特化したIT部門にも普遍的なニーズがあるのです。

明かりをつけておく

コストセンターはすべてをできるだけ安くするようにプレッシャーをかけられるため、IT部門は予算の範囲内で最低限のことだけをするように仕向けられることもよくあります。これを「明かりをつけておく」状況といいます。この状況下のIT部門の役割は、システムが基本的な業務ニーズに応えられるようにするというシンプルなメンテナンスに格下げされます。

IT部門はメンテナンスに集中し、新しいサービスを行うことを躊躇してしまいます。IT部門がこのような立場に立たされるのは残念なことです。必要なビジネスアプリケーションを稼働させることは非常に重要なのに、多くの企業でIT部門の成果は過小評価されており、基本的なオペレーションをできるだけ効率良く管理することだけを考えた結果、外注やクラウドへの移行が始まってしまいます。この種のサービスは様々な方法で商品化されています。

現代のビジネスとITドライバー

ITの力をバックオフィスやサポート機能としてだけでなく、ビジネス全体に活かせたらどうでしょう。また、IT部門や基盤となるインフラストラクチャを単に管理すべきコストとしてではなく、成長や新しいチャンスを促進するドライバーとして開発したらどうでしょうか。

アプリ主導

現代はアプリに支配されています。あらゆる用途のアプリがあふれており、お客様も従業員もリモートオフィスのPCやシンクライアント、モバイルなどを問わず、いろいろなデバイスからビジネスサービスにアクセスする必要があります。現代のアプリの多くはデータセンターに保管されている情報にアクセスしたり、アプリケーションに変更を加えるためのインターフェイスにすぎません。例えば、銀行を訪れることなく頻繁に使用するファイナンシャルサービスすべてにアクセスできる利便性が求められるモバイルバンキングが好例です。これらはローカルのデータセンターでも稼働しますし、クラウドサービスやSoftware-as-a-Service (SaaS) プラットフォームで稼働することもあります。

稼働する場所に拘わらず、事業部はお客様のニーズの変化や新しいサービスチャンスに応えるため、アプリケーションを継続的に更新または変更する必要があるため、IT部門はこのような変化を安定的に素早く実行する必要があります。

バイモーダルIT

現実的には、IT部門がオペレーションのサポート業務から退任する日は当分来ないでしょう。既存システムの多くは業務に必要な不可欠であるため、既存のシステムと新しいシステムをつなげることで、IT部門をバックオフィスアプリケーションの管理に縛るのではなく、ビジネスインシアチブのサポートにも時間を使えるようにすることが大切です。基本的な業務（ビジネスの「明かりをつけておく」業務）を正しく行えない場合、IT部門を他の業務で活用するチャンスはないでしょう。

ただし同時に、IT部門がコストや効率の管理支援を行うだけで

なく、利益を生み出せるようにしたいという差し迫ったニーズもあります。Gartner社はこの分岐したITオペレーションモデルを「バイモーダルIT」と呼んでいます。第一のモードは従来型の安定性に特化したモデルで安全性と正確性を重んじるものです。第二のモードはGartner社いわく、「試験的で非直線的なモデルで俊敏性と速度を重んじるもの」だそうです。

実際はどちらのモードも重要です。本書では、両ITモードの目標を一つのインフラストラクチャを使用して達成する上でコンポーザブルインフラストラクチャが役立つ理由を説明します。

俊敏性にフォーカス

ビジネスの俊敏性は日に日に重要性を増してきています。俊敏な企業の競争優位性は古いITサイロにとどまっている企業よりも高くなります。テクノロジーツールが適切であれば効果が出るということに事業部が気づく中、IT部門による承認を必要とし、対応するインフラをIT部門の管理により実装する古いパラダイムは許容できなくなってきました。

アイデアエコノミーは合理的な範囲で、シームレス且つユーザーに意識させない方法でプロビジョニングされたインフラストラクチャを使用してサービスを提供する能力を事業部に要求します。そう、インフラストラクチャそのものはIT部門の管轄のままですが、今やIT部門は新しいアプリケーションとサービスを高速で提供するための仲介役にすぎないのです。

このコンセプトはまた、IT部門に新しいサービスをよりシームレスに展開する能力を要求します。IT管理者は専門化チームに頼ることなく、一元化されたコンソールから素早く簡単に環境全体を立ち上げる必要があるのです。



これにより、IT部門は従来の役割から事業部目標、ひいては企業目標を達成するという新しい役割に移行しています。

利益を生み出す

現在の新種のアプリやサービスは組織全体の成長を促進し、ビジネスの存続を可能にしました。今ではIT部門がサポートするサービスでも利益を生み出すことができます。例えば銀行では、お客様がサービスの一環として必ず求めるであろうモバイルアプリケーションを新しく展開することを検討します。素早く簡単に拡張でき、管理が簡単で、プログラムにより実行可能なインフラストラクチャ（後ほど詳しく説明します）を備えることで、IT部門のサポートが必要であったとしても、これまで自動化できなかった部分を自動化した、新しいサービスの開発が大幅に簡単になります。

検討事項

ここで、モバイルバンキングアプリケーションについて少し考えてみましょう。ほとんどのお客様の給料日が月末であると想定すると、その時期にサービス利用のピークが来ると予測でき、ピークに対応するため、より多くのリソースを実装するスケジュールを立てるのも簡単になります。

ただし、ピークが分かりにくい場合はどうでしょうか。例えば、負荷の増加に対応するため、アプリケーションが自動的

に追加のウェブサーバーを起動できるとしたらどうでしょうか。また、この追加ウェブサーバーが人間の手を煩わせることなく、アプリケーションにより自動で完全に制御されるロードバランサーグループの一部になるとしたらどうでしょうか。

これこそアイデアエコノミーの課題の一つであり、コンポーザブルインフラストラクチャがIT部門の業務を変革する上で役立つポイントです。

俊敏性、速度とその現実

従来のIT部門は安定性重視でした。

しかし、現代のデータセンターでは俊敏性（新しいニーズを満たすためにITおよびそのサービスを調整する能力）と作業速度（新しいサービスや製品を市場に投入する速度）が大切です。この2つが速ければ速いほど組織は新しい収入源を手に入れたり、顧客満足度を改善することで、より速く利益を得られます。図1-1ではこの2つのバランスを示しています。

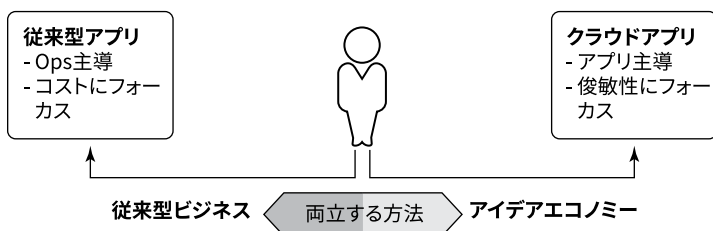


図1-1：2つの分岐するパラダイム間のバランスを慎重に計る現代のIT部門



これらの競合するニーズには厳しい現実が待っています。従来型環境ではIT部門がハードウェアのサイロの中で行き詰っており、クラウド環境では複数のクラウドベンダーによるオペレーションサイロを抱えています。

さて、IT部門はどうしたらいいのでしょうか。

従来型ITとアイデアエコノミー

本章の最初に、以下に挙げる2つの大きく異なるアプリケーション分類について説明しました。

- ✓ **従来型アプリケーション**は協働、データの処理や分析、サプライチェーン、ウェブインフラストラクチャなど、既存のビジネスプロセスをサポートし、自動化するように設計されています。この種のアプリケーションは企業内で安定したオペレーションを維持するために重要です。

✓ **新種のアプリケーションとサービス**はその携帯性やビッグデータ、クラウドネイティブテクノロジーを利用して利益や顧客満足度を向上させます。この種のアプリケーションはデジタル/アイデアエコノミーを促進する上で重要です。

IT部門はオペレーションコストを削減するために従来型環境をフル稼働すると同時に、オペレーションの速度を上げるために新しいアプリケーション環境をフル稼働させており、この状況に変化が必要なのは明らかです。

現代の課題に取り組むにはテクノロジーの力を利用して素早くクリエイティブなプロセスの高速化を行い、アイデアを実現することが大切です。IT部門の役割はビジネスを継続させることから、ビジネスにおける価値が生み出されるまでの時間を短縮することに進化する必要があります。この新しい宣言にIT部門が同意したとしても、従来型インフラストラクチャは速度ではなく、安定性、拡張性、およびパフォーマンスを提供するために最適化されており、現代の優先順位を実現するのは困難です。アイデアエコノミーでは、インフラストラクチャは価値創造のエンジンでなくてはならず、成功を邪魔するハードルになってはなりません。

どちらのワークロードも同じインフラストラクチャで稼働できるとしたらどうでしょう。また、IT部門が流動的で拡張可能な環境を提供しつつ、業務に不可欠で、明かりをつけておくことができ、安定性を期待できるアプリケーションを簡単に管理できたらどうでしょう。

コンポーザブルインフラストラクチャをお試しください。概要は図1-2で示す通りです。



図1-2：コンポーザブルインフラストラクチャの概要

何でも稼働できる

最も進んだ企業でも、業務に不可欠な従来型のアプリケーションを停止するわけにはいきません。それだけでなく、こういったアプリケーションも堅牢な環境で稼働させる必要があります。

同時に、新種のアプリケーションをサポートする環境も必要ですが、コンポーザブルインフラストラクチャなら選ぶ必要がありません。既存のアプリケーションをすべて稼働しつつ新しいアプリも稼働でき、さらにそれぞれを適切なサービスレベルに最適化できます。物理および仮想のコンピュート、ストレージ、ファブリックで構成される流動的なプールを備えた単一のインフラストラクチャで設備投資（CapEx）を削減しながらリソースを解放できます。

動作が速い

新しいアプリやサービスをネイティブでサポートする単一の包括的なインフラストラクチャを使用すれば、IT部門は素早くアプリやサービスの提供スピードを上げることができ、コストセンターから収入源に変身できます。コンポーザブルインフラストラクチャを使用すれば、ほぼ即時にインフラストラクチャを正確に構成する単一のインターフェイスを通してアプリケーションやサービスの提供を高速化できます。

悪くないですね。

効率よく稼働

様々な用途に分かれたインフラストラクチャのサイロを管理する試みは、みるみるうちに時間やお金を無駄にします。サイロを管理するには複数のツール、人材、およびプロセスのオーケストレーションが必要です。すべてのニーズに応える単一のインフラストラクチャを使用すれば、業務の手間やコストを削減することができ、イノベーションに集中できます。



コンポーザブルインフラストラクチャは物理環境のストリーミングだけでなく、テンプレート主体の摩擦のないオペレーションを行う社内のソフトウェア・デファインド・インテリジェンスを通して業務の手間やコストを削減します。

価値を引き出す

結局、IT部門も成果をあげなくてはならないのです。アイデアエコノミーでは、従来型アプリケーションをサポートしながら新しいアプリやサービスをサポートすることが成果になります。コンポーザブルインフラストラクチャを使用すれば、IT部門は一元化されたアプリケーションプログラムインターフェイス（API）を通してインフラストラクチャのオペレーションとアプリケーションを統合し、自動化することでデータセンター全体の生産性と管理性を高めることができます。

第2章

ユートピアの達成

本章の内容

- ▶ 時間、場所、ワークロードを選ばないインフラストラクチャの展開
- ▶ 旧型と新型をつなぐブリッジを築くこととブリッジがなくなる理由
- ▶ プログラマブルインフラストラクチャによるオペレーションの円滑化とビジネス価値の創造

IT部門は変わる必要があり、今こそ第1章で紹介したコンセプトを軸に、より良いシステムを築く時です。新体制を実現し、成功に導くには、何が可能かを理解し、プランを練る必要があります。

本章ではカギとなる3つの検討事項について説明します。これらのコンセプトはコンポーザブルインフラストラクチャの基盤で、本書の後半で詳しく説明します。

時間、場所、ワークロード を選ばず適切なサービスレ ベルで

ご存知かと思いますが、アイデアエコノミーに対応するにはITを見直す必要があります。成果に関して言えば、コンポーザブルインフラストラクチャの実装を通して新しいビジネス環境に対応する上で、どんな目標を達成したいのかが重要です。

時間を選ばない

インフラストラクチャの展開はそのユーザーによりオンデマンドで行えるようにすべきです。こういったデマンドベースの展開が役立つ例として最も一般的なのがDevOpsです。



アイデアエコノミーは、どんな会社でも急速に革新を遂げる製品やサービスのファブリックの一部となることをIT部門に要求します。IT部門はビジネスがその競争性を保てるよう持ちこたえるか、場合によってはリードしなければなりません。

このような状況が、現在IT業界を牛耳っているDevOpsアプローチの台頭につながりました。DevOpsの利用を検討している組織にとって連携は当たり前で、異なる指揮系統により開発と運用が分離されるような事態はもう起こりません。DevOps環境では、開発スタッフと運用スタッフが緊密に連携し、初期構想から製品サポートまで、ソフトウェアのライフサイクル全体をサポートします。

DevOpsグループ、およびその他の利害関係者に要求されるレベルのITインフラストラクチャを提供するため、IT部門はユーザーがリソースの構成や再構成をいつでも行えるようにスピード、俊敏性、柔軟性を高める必要があります。



コンポーザブルインフラストラクチャは、この時間を選ばない枠組みを実現します。

場所を選ばない

ハイパーバイザーのベンダーが望まなくとも、企業が抱える膨大なワークロードはなんと、物理サーバーで稼働しています！つまり、仮想化されていないアプリケーションがあり、それらは単純にサーバー上で稼働しているのです。

今日の最先端テクノロジーの数々により、その事実は忘れ去られているようです。例えば、多数のハイパーコンバージドソリューションを備えたハイパーコンバージドインフラストラクチャと同様の素晴らしい効果を上げるには、完全に仮想化されたワークロードを備えることが必須条件の一つになります。その場合、基本的に物理ワークロードはサポートされません。

物理および仮想ワークロードは、現代的なデータセンターでサポートする必要がある2つのアプリケーションの基盤にすぎないのです。現在、特定の種類のワークロードに対して大きな効果を与えるものとして注目されるつつある構成物がコンテナです。残念ながら、従来のインフラストラクチャのアプローチを使ってこの構成物をサポートするには、少なくとも物理、仮想、コンテナベースのワークロードをシームレスに処理できるインフラストラクチャが展開されるまで、カスタマイズされた専用のインフラストラクチャをIT部門で構築する必要があります。

このすべてのサポートを任せられ、それをサポートするために従来のインフラストラクチャを使用しなければならないIT部門の直面する複雑さを考えてみてください。「悪夢」、「高額」、「信じられない」といった言葉がピッタリな状況です。それをサポートするには、どの環境においても専用のハードウェアとソフトウェアが必要で、下手をすると追加の従業員まで必要となりかねません。ハードウェアやソフトウェアと同様に人件費は高額になってしまいます。



コンポーザブルインフラストラクチャは物理システムでも仮想システムでもワークロードを実行する能力をサポートする環境を提供します。

ワークロードを選ばない

コンポーザブルインフラストラクチャではワークロード専用の環境を用意する必要がなくなり、各アプリケーション特有のニーズを満たすために動的に組み合わされた流動的なリソースセットを提供します。

今後とも稼働させる必要があるオンプレミス型のExchange Serverをお持ちですか？今のところビジネスに役立っているものの、廃止に10年ほどかかりそうな企業リソース計画（ERP）システムをお持ちですか？また、開発努力の一環として動的なコンピューティング環境を使用し、開発者を後押しするDevOps構想をお持ちですか？



適切な種類のインフラストラクチャを使用すれば、これらすべてを同時に達成できます。コンポーザブルインフラストラクチャでは、どんなワークロードもソリューションの一環として処理できます。

適切なサービスレベル

アプリケーションやサービスが展開されている場所にかかわらず、特定のアプリケーションの優先度が低く設定されている場合でも、サービスレベルは重要です。コンポーザブルインフラストラクチャ環境でのサービスレベルは、コンピュート、ストレージ、ネットワーキングを含む可変的リソースプールをユーザーに提供することで達成できます。アプリケーションが違えば、異なる組み合わせのリソースが必要になります。高パフォーマンスのストレージが必要なものもあれば、低いレベルのストレージパフォーマンスに対応しているものもあります。どんなワークロードのオペレーションも可能なインフラストラクチャを提供することで、各ワークロードに対し、適切なサービスレベルを確立できます。これにより、IT部門はオンデマンドのリソース管理に移行でき、既存の環境の能力を上回る新しいアプリケーションが展開された場合など、予期できない事態が起こった場合のみ対処するだけで済みます。

ビジネスを加速するプログラマブルインフラストラクチャ

インフラストラクチャを素早く展開するため、開発者は*infrastructure as code*に対応することを検討するといいいでしょう。これにより、開発者は展開やバージョン管理、解体までを自身が開発したアプリケーションソフトウェアと同様に管理することができるようになります。この強力な共通ソフトウェアレイヤーはその後、データセンターのすべてのリソースを対象とした完全で、著しくプログラマブルな抽象化レイヤーになります。

抽象化は仮想化に近いため、ユビキタスコンピュートハイパーバイザーが重要な役割を担います。しかし、*infrastructure as code*を達成するには、抽象化をインフラストラクチャレベルで行う必要があり、すべてのリソース（コンピュート、ストレージ、およびネットワーキング）が重要な役割を持つようになります。すべてのリソースがソフトウェア要素として構成されるため、実行者はインフラストラクチャに対するコントロールを強化し、このコントロールを使用して数年前には成し得なかった方法でビジネスを高速化することができます。また、様々なインフラストラクチャ要素は単純にソフトウェアの延長となり、コードとして扱われます。

第3章

インフラストラクチャ の4つのカテゴリ

本章の内容

- ▶ 従来型インフラストラクチャが重要なビジネスニーズに応えられなくなった理由
- ▶ 台頭するデータセンターアーキテクチャを導入する機会
- ▶ 新しいアーキテクチャの欠点
- ▶ コンポーザブルインフラストラクチャがビジネスワークロードのニーズを満たす方法

IT部門のリーダーはビジネスニーズを満たしてくれるデータセンターアーキテクチャを何年も探しています。一般的に、データセンターアーキテクチャは以下の4つのカテゴリに分類されます。

従来型インフラストラクチャ

まず、多くの企業が未だに使用している従来型データセンターアーキテクチャです。現代の従来型インフラストラクチャソリューションは仮想化が現れる以前の日々を思い起こさせます。そう、データセンターのオペレーターが物理サーバーで一生懸命働くしかなかったあの頃を。オペレーターは事業部が新しい物理サーバーを2、3台から時には10台以上立ち上げなければできない新しいアプリケーションを求めるたびにやりきれない思いをしてきました。

従来型インフラストラクチャは長い間役立ってきましたが、弱点もいくつかあります。まず、様々なアーキテクチャ（物理、仮想、およびコンテナ）を展開するには様々なハードウェア

とソフトウェアが必要です。さらに、様々なスキルセットが必要なため、包括的で管理が簡単なものではなく、「ぎこちない」インフラストラクチャになってしまう可能性があります。



従来型インフラストラクチャは最終的にサイロだらけのデータセンターになってしまいます。ストレージを管理するチームとサーバーを管理するチームだけでなく、ネットワークレイヤーを管理するチームまで必要です。また、コンピュータ環境とストレージ環境を組み合わせた仮想化などの部門間協力を行う他のチームが必要になる場合もあります。

結果として、柔軟性のない静的な環境ができあがります。また、複数のチームとあらゆる非効率的な手間を巻き込んだ、程度の異なる手動プロセスを含む複雑なプロセスもあります。

結局、すべてを管理するのが大変になってしまいます。多大な調整と、手に入れることが困難で費用も高くなってしまい様々なスキルが管理に欠かせなくなってしまうからです。

コンバージドインフラストラクチャ

従来型インフラストラクチャの構成を個別に調達し、組み立てるのには多くの時間やお金、知識が必要です。ただし、綿密に計画を練っても予想もしなかった課題にぶち当たることがあります。すべてを自分で組み立てようとすると、構成が欠けたり、その構成が必要な様式に合わなかったりすることもあります。

また、様々な構成のそれぞれが異なるタイプの知識を必要とする別々のツールで管理されます。

コンバージドインフラストラクチャはこのような課題に対応するために作られました。コンバージドインフラストラクチャ環境では、事前に構成された物理ITアセットを購入することになります。コンピュータ、ストレージ、ネットワーク環境全体をセットで調達することで、それぞれを思った通りに連携させることができます。さらに、コンバージドインフラストラクチャにより、管理が簡単になることもあります。

ここで舞台裏を覗いてみましょう。このような環境でも、これまで買っていたハードウェアと同じようなものを買う必要があ

りますが、パッケージとして買うことができます。コンバージドインフラストラクチャシステムは新種のハードウェア上に構築されるわけではなく、互いに連携することが証明された既存の製品を組み合わせせてできています。



コンバージドインフラストラクチャには以下のような見逃せないメリットがあります。

- ✓ **従業員の生産性の向上**：様々なハードウェアのサイロが互いに連携するかを心配しなくていいので、すべてを正常に機能させることよりも環境を最大限使用することに集中できます。
- ✓ **人材やソフトウェアで複雑性に蓋をする**：多くのコンバージドインフラストラクチャソリューションは環境全体を管理する管理ツールとともに販売されるため、すべてを別々に管理するよりも大幅に管理しやすくなります。こういった面で、インフラストラクチャソリューションは従来型環境よりも、はるかに先を行っています。
- ✓ **大幅に簡略化された調達**：コンバージドインフラストラクチャはすべてを一括で買うことになるので調達が一瞬で済みます！

そうは言っても、コンバージドインフラストラクチャはソリューションを実行するハードウェアにより決まります。現代のビジネスにはびこる一番大きな課題を解決するのに必要な柔軟性はなく、相当絞り込んだワークロードタイプにのみ適しています。つまり、異なるワークロードタイプ向けのコンバージドインフラストラクチャで新しいサイロが作られる可能性があるのです。

ハイパーコンバージド インフラストラクチャ

ストレージエリアネットワーク（SAN）によって各サーバーの分散型ストレージの管理が不要になりました。簡単に拡張できますが、新しいリソースの管理と展開においては特有の課題があります。拡張性は大規模な実装においては理想的ですが小規模なサイトや企業では、大きすぎて人材やコストの管理ができなくなります。また、小さいインスタンスの配列が低いレベルのサービスを必要とする場合、コストの面でも管理オーバーヘッドの面でも行き過ぎになってしまいます。

もしこれを解消できるとしたらどうでしょう。

これがハイパーコンバージド インフラストラクチャが作られた一番大きな理由です。最近台頭してきたこのデータセンターアーキテクチャの形ではSANを排除することで複雑性をなくしています。一枚岩のストレージアレイを展開する代わりに、ハイパーコンバージド インフラストラクチャクラスタを構成する様々なノードにストレージが分散されます。

クラスタ内の各ノードはハイパーバイザーを実行しながらローカルストレージを管理します。今言ったことを忘れないでください。ハイパーコンバージド インフラストラクチャのノードはそれぞれ仮想ホストおよびストレージクラスタのノードとして稼働します。つまり、ハイパーコンバージド インフラストラクチャは物理ワークロードのサポートに向かないのです。物理サーバーがハイパーコンバージド環境に閉じ込められたストレージリソースにアクセスできないためです。恐らくこれこそハイパーコンバージド インフラストラクチャを稼働する上での最も大きな欠点でしょう。

ハイパーコンバージド インフラストラクチャはソフトウェアデファインドストレージを使用することで功を奏します。ソフトウェアデファインドストレージもまた、今日の市場をけん引するコンセプトの一つです。ソフトウェアデファインドストレージ環境にはローカルノード上のストレージ管理を担うと思われるソフトウェア（ハイパーバイザーカーネルに直接構築された仮想マシン装置で稼働することもある）があります。ハイパーコンバージド インフラストラクチャでは、これらストレージ管理ソフトウェアレイヤーそれぞれが他のノード上のピアすべてと連携し、一枚岩のSANを必要としない、完全に分散かつスケールアウトされたストレージ環境を簡単に作成します。

ハイパーコンバージド インフラストラクチャを検討する理由は以下のように様々です。

- ✓ **管理のしやすさ**：SANが過去のものになってしまったため、管理が必要な専用ストレージリソースが必要なく、多くの企業（特に人件費が高額なストレージ管理者を雇えない中小企業）が胸をなでおろしています。ストレージはすべてハイパーコンバージド インフラストラクチャソフトウェアベンダーにより提供されるハイパーバイザー管理ソフトウェアまたは統合管理ツールを通して管理されます。

- ✓ **シンプルな拡張**：クラスタに別のノードを追加するだけで容量（ストレージ、処理力、RAM、ネットワークアップリンクなど）を増やせます。とっても簡単ですよ！また、所属する会社の成長に合わせてだんだん増やすこともできます。同じ会社など1つもなく、どの会社もそれぞれのスピードで成長する必要があります。ハイパーコンバージドインフラストラクチャなら、流動的な仮想ITをサポートする環境への移行も視野に入れることができます。
- ✓ **コスト**：わずか2、3のノード（ハイパーコンバージドインフラストラクチャのベンダーによる）を展開する能力により、従来型環境を使うために事前に大量のストレージを購入するよりも上記のソリューションへ移行する方が安く済む場合があります。



先ほどチラッとほのめかした通り、ハイパーコンバージドインフラストラクチャを検討する際は、以下のような忘れてはいけない深刻な課題があります。

- ✓ **仮想のみ**：物理サーバーはハイパーコンバージドインフラストラクチャ環境で歓迎されません。多くの企業では未だに物理サーバーと仮想サーバーの両方を使用しているため、少なくとも2つの環境（物理サーバー環境とハイパーコンバージドインフラストラクチャ環境）の管理に時間を取られてしまうことになるでしょう。
- ✓ **物理インフラストラクチャに対応していない**：大切なことです。もう一度繰り返します。
 - Fibre Channel SANへの接続が必要なアプリケーションに対するSANサポートはありません。
 - HCシステムがコンピュータとストレージの両方を組み合わせるため、片方を追加するには、もう片方も追加する必要があります。

コンポーザブルインフラストラクチャ

従来型インフラストラクチャ、コンバージドインフラストラクチャ、およびハイパーコンバージドインフラストラクチャの長所と短所については大体理解できたかと思います。ここからは、そのようなアーキテクチャにより生じる課題を解決するためにコンポーザブルインフラストラクチャができることを大まかに説明します（表3-1も参考にしてください）。

- ✓ **ハードウェアとソフトウェアを一元化**：他の多くのデータセンターアーキテクチャと違い、コンポーザブルインフラストラクチャシステムは環境を管理するソフトウェアの側に構築されています。この緊密な連結により、アーキテクチャの両側が協力し合ってビジネスワークロードをサポートします。
- ✓ **流動性の高いIT**：適切なハードウェアにより、コンポーザブルインフラストラクチャはコンピュート、ストレージ、ファブリックなどのリソースの管理を非常に簡単にします。
- ✓ **すべてがソフトウェアデファインド**：一般的に、ハードウェアに縛られたアーキテクチャは、現代のビジネスに必要な不可欠なこの種の柔軟性や俊敏性を享受しません。しかし、コンポーザブルインフラストラクチャはすべてのエンタープライズインテリジェンスをソフトウェアにまとめ、抽象化されたリソースをいとも簡単にシステムで管理できるようにし、効果を最大限に引き出します。
- ✓ **物理、仮想、およびコンテナ化されたワークロード**：コンポーザブルインフラストラクチャでは、稼働させる必要があるワークロードについて妥協する必要がありません。どうぞお好きのように！

表3-1 データセンターアーキテクチャの範囲

	従来型	コンバージド	ハイパーコンバージド	コンポーザブル
複雑性	高	中	中	低
購入から実際に価値が生み出されるまでの時間	日、週、月単位	日～週単位	数時間、数分	数分、数秒
柔軟性	中だが非常に複雑	中	中	高
スケーリング	難	中	易	易
ワークロード対応	物理、仮想、コンテナ	物理、仮想、コンテナ	仮想、コンテナ	物理、仮想、コンテナ

第4章

流動的リソースプール

本章の内容

- ▶ コンポーザブルインフラストラクチャの背景
- ▶ 分割されたリソースの重要性
- ▶ コンポーザブルインフラストラクチャ環境でサポートされる様々なワークロードタイプ
- ▶ コンポーザブル環境の拡張がいかにより簡単か

続

く3つの章ではコンポーザブルインフラストラクチャを形作る3つの特長、流動的リソースプール、ソフトウェア・デファインド・インテリジェンス、およびユニファイドアプリケーションプログラミングインターフェイス（API）について解説します。その前にまず、アーキテクチャ全体を見てみましょう。

コンポーザブルインフラストラクチャとは

コンポーザブルインフラストラクチャについてはすでに少し触れましたが、本章ではより詳しく説明します。組織によるアイデアエコノミーの探求を従来型、コンバージド、およびハイパーコンバージドアーキテクチャが支援する上で生じる課題についてはすでに説明しましたが、これを補足するコンポーザブルインフラストラクチャアーキテクチャとはどういう仕組みなのでしょう。

コンポーザブルインフラストラクチャには現代の企業ニーズを満たすためにピッタリの3つの特長があります。図4-1ではこの3つの特長を簡単に説明しています。本章、第5章、および第6章では、この特長について詳しく見ていきます。

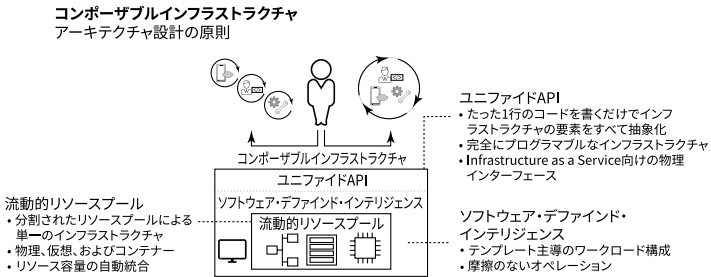


図4-1：コンポーザブルインフラストラクチャに対するHPEのビジョン

流動的リソースプール

第3章では、自由に使える様々なデータセンターアーキテクチャの選択肢について説明しました。どの選択肢にも異なる機能があり、とりわけ環境を拡張する際にリソース同士を調整する方法に関して、それが顕著に現れていました。



従来型、コンバード、およびハイパーコンバード インフラストラクチャの問題は、環境を作るリソースに関するものがメインです。様々な種類のワークロードをサポートするにはインフラストラクチャのサイロを構築しなければならないことも多く、それによりデータセンター環境のコストや複雑性が増してしまいます。

さらに、システムアーキテクチャの中には（特にコンバードとハイパーコンバード インフラストラクチャ）管理者がリソースをロックステップ方式で拡張しなければならないこともよくあります。例えば、ハイパーコンバード インフラストラクチャ環境を稼働しており、数テラバイトのストレージを追加する必要があるとしましょう。フルサーバーノード単位でリソースを追加する必要があるため、コンピュータとRAMも追加しなければなりません。

また、コンポーザブルインフラストラクチャでは、管理者が基盤となるコンピュー、ストレージ、およびファブリックリソースを切り出すという方法で環境を構築でき、こういった問題を効果的に解決できます。この場合、「切り出す」というのは、基本的に、個別のリソースのスケールを別々に拡張できるということを意味します。実際に必要なリソースを手に入れるために不要なリソースを追加する必要はありません。

さらに素晴らしいことに、コンポーザブルインフラストラクチャを使用すれば、基盤となるリソースが拡張のために分割されていても、単一の包括的なインフラストラクチャとして管理やワークロードに使用できます。ワークロードは必要に応じて単一のリソースプールからリソースを引き出せます。このリソース割り当てプロセスは管理者による手動操作も可能で、強力なアプリケーションプログラミングインターフェイス（API）に対応させ、開発者がデータセンターアーキテクチャをコード要素として扱い、アプリケーションの使用量に応じて自動的にリソースを割り当られるように開発することもできます。

さて、実用面では一体どういうことなのでしょう。簡単に言えば、コストを大幅に削減できます。従来型環境では、複数のインフラストラクチャのサイロ間で大きなリソースの重複があり、設備投資コスト（CapEX）も運用コスト（OpEX）も大きく増加してしまいかねない状況でした。単一の流動的なリソースプールに移行することで、サイロを事実上排除することができます。初めて仮想化を導入した時のような感じになるでしょう。ワークロードをサポートするために冗長的なリソースをたくさん立ち上げる必要はもうありません。しかし現代では、多くの組織が単一のサーバーを立ち上げるよりも、環境全体を展開して複数のアプリケーションをサポートし、驚くほど効率性を上げる可能性を選びます。

ワークロード対応

データセンターで稼働するワークロードの形やサイズは物理、仮想、コンテナベースなど、いつも様々です。従来のツールを使用してこれらのワークロードに共通のプラットフォームを提供することは困難で、費用もかさみがちです。コンポーザブルインフラストラクチャは、必要な時に、あらゆるタイプのワークロードを稼働できる単一のリソースプールを提供します。



コンポーザブルインフラストラクチャは仮想ワークロードとコンテナベースのワークロードをサポートする環境と全く同じ環境で物理ワークロードを管理する能力を備えています。

物理

物理サーバーが無能であるという思い込みは忘れましょう。物理サーバーは有能であり、今後もすぐに消えるようなことはありません。物理ワークロードは排除されたものと勘違いしないように、もう一度考えてみましょう。多くの企業が新しいワークロードに対して「仮想第一」のアプローチを取っているとしても、物理ワークロードは過去のものということにはならないのです。

実際、色々な理由で、物理ワークロードの方が好ましいことが多いのです。その理由は以下の通りです。

- ✓ **パフォーマンスニース**：他のアプリケーションは膨大な馬力を必要とし、サーバーが持つすべてを単純に消費できる時に最も効果を発揮します。
- ✓ **ライセンス付与**：アプリケーションの中には、仮想環境でのライセンス付与がリスクと法律用語の巣窟を作り、結果として、ライセンス付与のコストが高くなってしまいうものがあります。実際、厄介なライセンス付与制限に対して注意を払わない場合、そのようなライセンス付与に関して顧客を訴えるベンダーもあります。

典型的な物理インフラストラクチャと同じじゃないかと思う方もいるかもしれませんが。大きな違いはたった1つの単語、「ステートレス」に集約されます。中央制御のステートレスなインフラストラクチャ環境を提供することで、IT部門は必要に応じてワークロードをシームレスに新しい場所へと移動させることができます。

コンポーザブルインフラストラクチャは従来型環境をコンピュータ、ストレージ、ファブリックの流動的なプールと置換えます。これにより、サーバー、ストレージ、ネットワークリソースの要素が削られ、最低限に集約されます。この環境では、コンピュータ、ネットワーク、ストレージがステートレスでアノニマスなりソースとして稼働し、アプリケーションのニーズに即座に応えられるよう、簡単に組み合わせたり、構成したりできるようになります。このような機能により、仮想ワークロードを実装するのと同じ方法で物理ワークロードをプロビジョニングできます。

仮想

物理システムを稼働する大きなメリットもありますが、実際は新しいワークロードの80%が仮想環境で実行されます。これを考慮すると、どんなデータセンターアーキテクチャでも仮想化を包括的にサポートする必要があるのは火を見るより明らかです。

コンポーザブルインフラストラクチャなら、仮想化に対するサポートが「標準装備」されます。これらのワークロードは物理およびコンテナベースの環境と同じインフラストラクチャ環境に存在し、それらのプラットフォームと共に管理されます。

コンテナベース

コンテナはワークロードを実行する方法として日増しに普及してきており、特にDevOps主体の企業でよく使用されています（第3章を参照）。コンテナはオペレーションシステムレベルの抽象化をもたらし、主要なサービスを実行します。



コンテナはソフトウェアの一部を実行に必要なすべて（コード、ランタイム、システムツール、システムライブラリなど、サーバーにインストール可能なあらゆる要素）を備えたファイルシステム一式に含みます。これにより、実行する環境に関係なく常に正常に稼働できるようにします。これにより、プログラマーや開発チーム、オペレーションエンジニアは現代のアプリケーションにおける分散型の、ネットワークでつながった特

今ある仮想化管理 ツールを使い続ける

たくさんの人が仮想化ツールに精通するようになり、毎日使用しています。コンポーザブルインフラストラクチャは効率と活用が要であること、また、従来型ITインフラストラクチャの80%

以上が仮想化されていることを考えると、コンポーザブルインフラストラクチャを使用することにより、仮想リソースを使い慣れたツールで管理し続けることができます。

質を活かすのに必要な共通のツールボックスを手に入れることができます。

コンテナに含まれるアプリケーションでは構成と依存性がパッケージ化されており、マシン間でのアプリの共有が簡単です。これにより、開発チームとの協働や開発者およびテスト間の協働が簡素化されます。開発者はコンポーザブルインフラストラクチャ環境のセットアップと管理に気をかける必要がないため、新しい機能の作成やエラーの修正、ソフトウェアの納品に全精力をつぎ込めます。

シンプルな両方向への拡張性

インフラストラクチャを追加する際は、既存のインフラストラクチャと自動で統合されるため、何もしなくても勝手に容量プールの一部に加わってくれます。リソースをすぐさま簡単に組み合わせられるため、インフラストラクチャを極端にオーバープロビジョニングする必要がなくなり、インフラストラクチャを適切なサイズでプロビジョニングできる上に、必要に応じて拡張できるようになります。



リソースの分割により、IT部門は必要なリソースだけを実装できるようになります。例えば、ストレージを追加したからといってより多くのコンピュータリソースを実装する必要はありません。

どちらのシナリオにおいても、確実にコストカットできます。IT部門は必要に応じて段階的に拡張でき、無駄になるリソースを展開する必要はありません。

従来型でありながらクラウドに近い

コンポーザブル環境により、組織は第1章で触れたアプリケーションタイプの両方を実行できます。従来型のリソースを要求する従来型のアプリケーションも実行できるのです。しかも、ビジネスニーズに応じて、同じ環境で同時に、クラウドに近い現代的なアプリケーションを実行できます。

どちらのITモードもサポートできることで、IT部門はビジネスをコントロールできるようになります。

第5章

ソフトウェア・ デファインド・ インテリジェンス

本章の内容

- ▶ テンプレート主導のインフラストラクチャ環境がIT業務全体を円滑化する方法
- ▶ 摩擦のないITの重要性とアイデアエコノミーでのその役割
- ▶ コンポーザブルインフラストラクチャがハードウェアとソフトウェアをつなぎ合わせ、IT部門がビジネスをコントロールできるようにする方法

第

4章では、流動的リソースプールとそれがデータセンターに必要な様々なワークロードをサポートする方法について説明しました。本章では、コンポーザブルインフラストラクチャ環境全体をつなぎ合わせるソフトウェアをご紹介します。

ソフトウェア・デファインド・ データセンター

ソフトウェア・デファインド・データセンター（SDDC）に関する様々なベンダーのイニシアチブについては皆様もご存知かと思います。多くの場合、これらの試みは断片的であったため、複数のソフトウェアレイヤーやアプリケーションプログラミングインターフェイス（API）、管理上のパラダイムをもたらしました。開発、テスト、実稼働環境全体を通して必要なのは素早く、一貫した、ポリシーベースのアプリケーションおよびインフラストラクチャの自動化です。

インフラストラクチャを素早く展開するため、インフラストラクチャをソフトウェアとして扱いたい組織が増え続けています。これにより、開発者は実装やバージョン管理、解体までを自身が開発したアプリケーションソフトウェアと同様に管理することができますようになります。この強力な共通ソフトウェアレイヤーはその後データセンターのすべてのリソースを対象とした完全で、構成可能かつプログラマブルな抽象化レイヤーになります。

すべてのリソースがソフトウェア要素として仮想でコントロールされるため、実行者はインフラストラクチャをコントロールし、数年前には成し得なかった方法でビジネスを高速化することができます。また、様々なインフラストラクチャ要素は単純にソフトウェアレイヤーの延長となります。

テンプレート主導のインフラストラクチャ

新しい仮想マシンを実装する必要がある場合にテンプレートを作成することが当たり前になっている環境もあります。ワークロードが必要とするプラットフォーム基盤に関係なく、お使いの環境にあるすべてのワークロードでこれを実現できるとしたらどうでしょう。

HPEコンポーザブルインフラストラクチャAPI（第6章を参照）を使用すれば、ITの管理者は自動テンプレートを通して物理、仮想、クラウドインフラストラクチャを基本的にオンデマンドで構成、管理できるようになり、リソースを素早く、効率よく、エラーの少ない方法で要求、変更、更新、修正できる能力を得られます。また、インターフェイスをハイレベルのオーケストレーションに使用することができ、複数のAPIにワークフローを無理やり合わせるのではなく、DevOpsワークフローと簡単に調整できます。これにより、管理者とソフトウェア開発者の生産性が上がります。コンポーザブルリソースは繰り返し使用できるテンプレートを使用して、ステート（BIOS設定、ファームウェア、ドライバー、プロトコルなど）やオペレーティングシステムイメージとともにプロビジョニングされます。何週間、何カ月とプロジェクトを遅らせるオペレーションサイクル全体で時間がかかるプロビジョニングプロセスを排除できるため、従来型ITにとってもDevOpsアプローチにとっても理想的です。ライフサイクルオペレーションの自動化はコストを削減し、時間を節約し、購入から実際に価値が生み出されるまでの時間を短縮します。

テンプレートは初期の展開だけにとどまらず、システムアップデートを簡素化し、インフラストラクチャの安定性を確保するためにコンプライアンスを実施することができます。実装計画を管理し、キャプチャやクローン、ゴールデンイメージのカスタマイズにより、ブート可能なイメージを作成できます。さらに、テンプレートを使用してイメージのプロビジョニング、アップデート、ロールバックを素早く行い、メンテナンス期間を最小化することでコンプライアンスを実施できます。

このソフトウェア主導のデータセンターアーキテクチャモデルはIT部門に従来のアーキテクチャでは不可能だった、卓越した柔軟性をもたらします。事実、この種の機能をパブリッククラウド環境に存在する機能と同等にすることは簡単なのです。このようなサービスを使用すれば、ワークロードを実行するために多数の物理サーバーやストレージを展開する必要はなくなります。クラウドプロバイダーに提供されたリソースを使用するだけでいいのです。



コンポーザブルインフラストラクチャ環境でも当然IT部門がハードウェアを展開する必要がありますが、ユーザーにとっては、IT部門がクラウドプロバイダーとなり、オンデマンドでテンプレート主導のリソースを使用可能にします。

摩擦のないIT

従来のコンピューティング環境では、IT部門の内部にも多くの摩擦と不安が存在していました。異なる管理サイロが異なる環境の側面を担っており、かみ合わないことも多く、縄張り争いに発展していました。この種の問題はユーザーのリクエストに遅れを生じさせるばかりです。

コンポーザブルインフラストラクチャは摩擦のないオペレーションという機能も提供します。様々なデバイスタイプにまたがる様々なファームウェアシステムを手作業で管理するという業務をIT部門の従業員に任せる必要がなくなるのです。ファームウェアのアップデートやサービスへのストレージの追加、ネットワークの接続性の修正といった変更作業はテンプレートにより自動で実行され、手動操作と人的エラーを大幅に削減します。これにより、IT部門は開発、テスト、実稼働環境向けにインフラストラクチャ全体を単一のインターフェイスで、誤差なく、正確に、素早く、ワンステップで構成できるようになります。

コンポーザブルインフラストラクチャはより万能なアプローチでリソースを管理できるようにすることで、ITの均等化による摩擦の排除に役立ちます。組織はリソースに特化した、その道の専門

家チームを求めるのではなく、データセンターに関する幅広い知識を持っている人材やビジネス目標に合わせてITインフラストラクチャを調整するスキルを持った人材を個別に雇えます。

ハードウェアとソフトウェアを一元化

コンポーザブルインフラストラクチャは柔軟なコンピュート、ストレージ、ファブリックプールとテンプレートベースのアプローチを使用して継続的デリバリーへの移行を促進します。オープンで一元化されたRESTful API（第6章を参照）を繰り返し使用可能なテンプレートと組み合わせて使用することで、コンポーザブルインフラストラクチャはハイレベルのオーケストレーションツール向けの、プログラムにより実行可能なインターフェイスを提供し、アイデアエコノミーへの道を開きます。この継続的デリバリーアプローチはスピードや俊敏性、競合優位性を企業にもたらしめます。



コンポーザブルインフラストラクチャでは、物理でも仮想でも、インフラストラクチャの要素が細かいリソース要素の構成や停止、および再構築を可能にする流動的なリソースプールで構成されます。リソース要素にはコンピュート、ストレージ、ネットワークが含まれます。

コンポーザブルインフラストラクチャ環境には、インフラストラクチャとワークロード環境の要素それぞれをすべて一元管理する場所があります。ソフトウェアインテリジェンスと単一のリソースプールは以下のような成果を実現します。

- ✓ **設備投資（CapEx）の削減：**様々なワークロードを実行する際にワークロード固有のリソースを展開する必要がないため、リソースが「閉じ込められる」こともなければ、アクセスできないサイロに分かれることもありません。
- ✓ **運用コスト（OpEx）の削減：**ワークロード中心のアーキテクチャに移動させ、複数の管理ツールを一つのツールにまとめることで操作の手間と時間が減り、リソースを他のイニシアチブに使えるようになります。
- ✓ **俊敏性の改善：**コンポーザブルインフラストラクチャのおかげで実現するもう一つの成果、俊敏性の改善は設備投資や運用コストの削減よりも重要かもしれません。理由は一つ。コストはある程度までしか管理できませんが俊敏性の向上は相乗効果を限りなく生み出すことができるからです。

第6章

ユニファイドアプリケーションプログラミングインターフェイス (API)

本章の内容

- ▶ APIは役立つもののなのに、その増殖が問題である理由
- ▶ コンポーザブルデータセンターですべてを一つに
- ▶ IT部門を単なるコストセンターではなく収入源として認識させる方法
- ▶ コンポーザブルインフラストラクチャでソフトウェア開発方法を徹底的に見直す

ア プリケーションプログラミングインターフェイス (API) については最後の章までとっておきましたが、流動的なリソースプールやソフトウェア・デファインド・インテリジェンスと同じくらい重要なコンポーザブルインフラストラクチャにおける革新的要素です。APIはコンポーザブルインフラストラクチャに元々備わっており、これによりインフラストラクチャ管理者や開発者はインフラストラクチャ全体をコードとして扱うことが可能になります。

API増殖の問題を見つける

ここ数年間で、ハードウェアやソフトウェアのプロバイダーのほとんどがAPIを実装し、プログラムによる製品へのアクセスを可能にしました。このようなAPIを使用すると、インフラストラクチャの管理者や開発者はコードを書き込むことで環境を操れるようになります。新しい仮想マシンは必要でしょうか。いいえ、全く必要ありません！コードを数行書くだけで全く新しい仮想マシンを構築できます。

様々なプロバイダーが手掛けた複数、時には何十もの異なるAPIそれぞれにコードを書く苦労を想像してみてください。複雑さとエラーが発生する可能性がグンと高くなるでしょう。では、1つの構成物を他のベンダーが手掛けた同様の構成物に変更したらどうなるでしょう。新しいベンダーのAPIを使用するためにコードを書き直さなくてはならなくなってしまいます。



この現実には持続可能なものではありません。APIの増殖は大きな問題なのです。

接着剤としてのコンポーザブル

コンポーザブルインフラストラクチャはソフトウェアデファインドテンプレートに基づいており、コンピュータからストレージ、ネットワーク、さらには個々のワークロードまで、環境のどの部分も操作できる一元化されたユニファイドAPIを管理者に提供します。

ユニファイドAPIはインフラストラクチャのオペレーションとアプリケーションを統合し、自動化することによりデータセンター全体の生産性とコントロールを向上させます。また、コンポーザブルインフラストラクチャの検出、検索、構成情報管理、設定、プロビジョニング、更新、診断を行う一元化されたインターフェイスを提供します。さらに、たった1行のコードでアプリケーションに必要なインフラストラクチャを表現してプロビジョニングでき、時間のかかる低レベルツールおよびインターフェイスの何百もの呼び出しのスク립ティングが不要になります。

ユニファイドAPIには以下のような多数のメリットがあります。

- ✓ たった1行のコードを書くだけでインフラストラクチャの要素をすべて抽象化できます。
- ✓ インフラストラクチャが完全にプログラマブルになります。
- ✓ Infrastructure-as-a-Service (IaaS) のような効果が得られる物理インターフェイスを得られます。

IT：コストセンターか収入源か

現実的に、IT部門は常にある程度のコストセンターであり続けます。つまるところ、これらの汎用ビジネスアプリケーション

が自分で自分を管理することはないのです！ただし、ITのモードには2種類あり、どちらもコンポーザブルインフラストラクチャにサポートされていることを忘れないでください。



アプリケーションの中には、ほぼ強制的にIT部門をコストセンターにしてしまうものもありますが、適切なインフラストラクチャを提供することにより、間接的ではありますが、IT部門は収入源にもなれるのです。

コンポーザブルインフラストラクチャのAPIはコインの表裏、どちらもサポートします。従来型環境では、IT部門がIT業務を自動化できるようになり、企業のニーズに従ってワークフローを設計できるようになりました。新種のアプリケーションの場合は、インフラストラクチャの実装、拡張、更新を通してDevOpsによるアプリケーションの自動化が可能になりました。ユニファイドAPIは仮想およびパブリッククラウドリソースと同様に物理リソースを収集するため、開発者は基盤となる物理要素を細かく理解していなくてもコーディングできます。

開発者にフォーカス

新しいコードを書き、テストを行った後に手動で環境を再構築する必要性がないため、開発時間を大幅に短縮できます。実際、コストを削減し、俊敏性を実現するため、IT分野の大部分において単純化が大きなトレンドになってきています。

例えば、コンポーザブルインフラストラクチャを使用すると、開発者が昼食時に新しいアプリケーションを思いついた場合、新しい開発環境（希望に応じて実働環境をクローニングすることも可能）が机に戻った時にはすでに稼働しているようにできます。新しいアプリケーションの開発にすぐさま取り組めるのです。

必要に応じて、複数のティアに分かれた複雑な環境をたった1行のコードを実行するだけで作成することも可能です。このインフラストラクチャ環境は仮想マシンだけではなく、必要に応じて構成、停止、異なるプールへの再構成を素早くできる流動的なプールを開発者が利用できるようにするためのリアルサーバー、ネットワーク、およびストレージハードウェアで構成されます。インフラストラクチャのコンポーザブルプールはビジネスにおける優先事項に基づいて、必要に応じて何度でも構成、再構成が可能です。

ITの変革

IT部門をサポートサービスの提供からTime-to-valueに特化した新しい基幹業務に変身させ、利益を生み出せるようにするにはどうしたらいいのでしょうか。

コンポーザブルインフラストラクチャは間違いなく、この目標を達成する方法の1つです。ご存知の通り、テクノロジーによるイノベーションは企業の至る所で生まれます。API主導のインフラストラクチャでプログラミングが可能になるので、IT部門は組織全体の開発者によるインフラストラクチャへのアクセスを可能にできます。

開発者がテスト環境を構築できるようにするだけでなく、もっと大局的に「infrastructure as code」の真の意味を考えてみましょう。このシナリオでは、たくさんのソフトウェアを構築する組織を想定します。



テストを有効なものにするには、開発部門が継続的に構築を行い、設定に一貫性を持たせる必要があります。企業で開発を行うには、基盤となるオペレーション環境を開発、テスト、実働環境として継続的に構築および再構築する必要があります。さらに、ほとんどのアプリケーションは正常に機能するために複数の基盤インフラ要素を必要とします。これらの要素にはデータベースサーバー、ネットワークデバイス、ロードバランサーなどが含まれる場合があります。HPEのコンポーザブルインフラストラクチャを使用することで、DevOpsチームはこれらの要素の作成を完全に自動化でき、インフラストラクチャではなく、コードに集中できます。



Infrastructure-as-codeのコンセプトをまとめると、仮想マシンと同様に、たった1行のコードで物理インフラストラクチャをプロビジョニングできるということになります。

コンポーザブルインフラストラクチャを使用すれば、動的で適応能力の高い基盤インフラにより、コストの高い、複雑な開発サイクルを減らすことが可能になり、インフラストラクチャは開発進展やイノベーションにおける障害でなくなります。

テストの高速化と自動化

品質管理チームもテストチームも最高の効果を引き出すためには、常に最新のコードを使用して業務を行う必要があります。

コンポーザブルインフラストラクチャを使用すれば、実働環境と同じインフラストラクチャでテストを行えます。また、テストの際には、そのインフラストラクチャを動的に割り当てられ、テストが終わったと同時に割り当てを解除できます。

データセンターにおける統合と自動化

Infrastructure-as-codeはHPEのコンポーザブルインフラストラクチャ戦略における要です。

完全にプログラマブルなインターフェイスは、データセンターの管理者が業務を行う際に使用する管理ツールに統合されます。また、オープンソースの自動化ツールやDevOpsツールに統合すれば今後にも備えることもできます。この統合により、インフラストラクチャの効果をデータセンターを使用する全員が享受できます。

- ✓ 仮想化の管理者はハイパーバイザークラスターを自動でプロビジョニングでき、仮想マシンを管理するのと同じインターフェイスを使用して、中断することなくインフラストラクチャを更新できます。
- ✓ 施設の運営者はData Center Infrastructure Management (データセンターインフラ管理：DCIM)インターフェイスを通して消費電力と熱動インフラを可視化でき、最適なワークロードの配備を提案します。
- ✓ IT管理者はクラウドインフラストラクチャを簡単に構築でき、ビジネスにおけるサービスプロバイダーになれます。
- ✓ アプリケーション開発者はDevOps方式を使用することで、ユニファイドAPIを通してインフラストラクチャをコード化でき、インフラストラクチャとアプリケーションを同じ方法で一緒に素早くプロビジョニングできます。

API主導コンポーザブルの見通し

コンポーザブルインフラストラクチャは柔軟なコンピューティング、ストレージ、ファブリックプールとテンプレートベースのアプローチを使用して継続的デリバリーへの移行を促進します。オ

オープンで一元化されたRESTful APIを繰り返し使用可能なテンプレートと組み合わせて使用することで、コンポーザブルインフラストラクチャはハイレベルのオーケストレーションツール向けのプログラムにより実行可能なインターフェイスを提供し、DevOpsのために道を開きます。

このアプローチの核となるのはユニファイドAPIであり、たった1行のコードでインフラストラクチャ要素を抽象化する能力を提供します。適切なインフラストラクチャと組み合わせることで、ユニファイドAPIは物理または仮想リソースの抽象化と自動化を実現します。この継続的デリバリーモデルはスピードや俊敏性、競合優位性を企業にもたらしめます。

第7章

コンポーザブルインフラストラクチャを選ぶ 10の理由

白

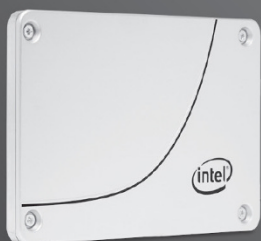
前の戦略を練るにあたって、プロバイダーに期待すべき効果がいくつかあった方がいいでしょう。HPEが考える、コンポーザブルインフラストラクチャの使用により得られる10の効果は以下の通りです。

- ✓ **あらゆるワークロードをサポート**：コンポーザブルなことで、仮想マシンでも、物理的な展開でも、コンテナでも、クラウドネイティブなアプリケーションでも、すべてのワークロードタイプをサポートするインフラストラクチャとあらゆるワークロードをプロビジョニングして稼働する能力を提供するべきです。
- ✓ **すべてのリソースを使用**：コンポーザブルインフラストラクチャはコンピュー、ストレージエリアネットワーク（SAN）、ローカルストレージ、ネットワークファブリックリソースの流動的なプールを提供し、アプリケーションのニーズに基づいて継続的に集積、分割、構成可能であるべきです。
- ✓ **単一のユニファイドアプリケーションプログラミングインターフェイス（API）**：完全にコンポーザブルであることで、シンプルでオープンな単一のRESTful APIを提供し、インフラストラクチャのすべての要素（コンピュー、ストレージ、およびファブリック）に対する抽象化と管理が可能になり、他のプログラミングの要素と簡単に連携させることができます。

- ✓ **ソフトウェア・デファインド・インテリジェンス**：コンポーザブルであることで、単一の管理プラットフォームに含まれるプログラマブルなテンプレート主導のソフトウェアインテリジェンスを提供できるため、インフラストラクチャのプロビジョニングやメンテナンスにおける複雑性はなくなるはずです。また、インフラストラクチャやアプリケーションと連携しながら自己検出や自動統合を行う機能も備えていなくてはなりません。
- ✓ **あらゆるリソースを使用**：コンポーザブルインフラストラクチャはAnsibleやChef、Docker、OpenStack、Puppet、VMwareなどのツールと最初から統合されているはずです。
- ✓ **完璧なInfrastructure-as-code**：コンポーザブルインフラストラクチャは物理環境でプライベートクラウドを展開でき、アプリケーションの物理リソースのプロビジョニングや管理が可能です。
- ✓ **コンポーザブルなアーキテクチャの製品**：ソリューションは最初からコンポーザブルであるように設計されるべきです。コンピュート、ストレージ、ファブリックおよび管理は連携し、ワークロードを稼働する際に適切なタイミングで適切なリソースをプロビジョニングする必要があります。
- ✓ **投資対効果を得る**：コンポーザブルであれば、オーバープロビジョニングやリソースの取り残しを排除することができます。ワークロードはそれぞれ必要なリソースのみを使用し、リソースがなくなった場合はリソースプールに返さなければなりません。これにより、単一のデリバリーモデルに関するプロセスやサービスの統合がとれ、複雑性やコストを削減できます。
- ✓ **今後のビジネスの備え**：拡張性や柔軟性に制限があっては いけません。コンポーザブルインフラストラクチャであれば、垣根を越え、あらゆるデータセンターと連携できるはず です。
- ✓ **コンポーザブル環境への移行を今すぐ開始**：次世代インフラストラクチャの展開を段階的に行えるソリューションであるべきで、必要なビジネスアプリケーションを中断することなく、コンポーザブルであることのメリットが最も必要な今だからこそ、それを提供すべきです。



REMOVE YOUR STORAGE BOTTLENECK



Feed your I/O starved applications with outstanding performance, extended endurance, and low latency from the Intel® SSD Data Center Family for SATA.

Intel.com/ssd

Copyright © 2016 Intel Corporation. All rights reserved. Intel, the Intel Experience What's Inside logo, and Intel Experience What's Inside are trademarks of Intel Corporation in the U.S. and other countries.



これらの内容は当社が著作権を保有しています。© 2018 John Wiley & Sons, Inc. いかなる公開、配布、および不正使用も厳禁します。

コンポーザブルインフラで データセンターを収入源に

ITと業務の整合性は何年もの間、耳の痛い表現でしたが、整合性がとれていないことは誰のせいでもありませんでした。しっかりと安定した、信頼のおけるインフラストラクチャを最善を尽くして提供することはIT部門の役割です。しかし、企業はIT部門やITインフラストラクチャに柔軟性や俊敏性の向上を求めています。このような矛盾した要求に応えるため、IT部門はアイデアエコノミーのコンセプトに基づいた新しいビジネススタイルを実践でき、柔軟性やスピードの高さを保ちながら、非常にしっかりと信頼できるインフラストラクチャ環境を実装する必要があります。

- ・ **データセンターアーキテクチャの範囲** — コンポーザブルインフラストラクチャがデータセンターのどこに適合するかを特定します。
- ・ **アイデアエコノミー** — IT部門がアイデアから新しいアプリを実現するために、システムの柔軟性と安定性が必要な理由を探ります。
- ・ **ITの2つのモード** — ITが新しいアプリやサービスの開発を促進しつつ従来型のアプリもサポートできる単一のインフラストラクチャを必要とする理由を学びます。
- ・ **コンポーザブルインフラストラクチャ** — コンポーザブルインフラストラクチャが企業に究極の柔軟性と俊敏性を提供する方法を解説します。

スコット D. ロウは1944年以来、IT業界で活躍しています。しばらく第一線で働いた後、10年間CIOを務めました。現在はActualTech Mediaのパートナーとなり、高等教育機関に分析やソリューションを提案するコンサルタントとしても活躍しています。



本書から学べること：

- ・ 今に至った経緯
- ・ アイデアエコノミーがITに
適応する方法
- ・ コンポーザブルインフラ
ストラクチャがアイデア
エコノミーを実現する
方法
- ・ コンポーザブルインフラ
ストラクチャの柱
- ・ コンポーザブルインフラ
ストラクチャーつで得ら
れる成果

その他の情報については
Dummies.com[®]
をご覧ください！

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook EULA.