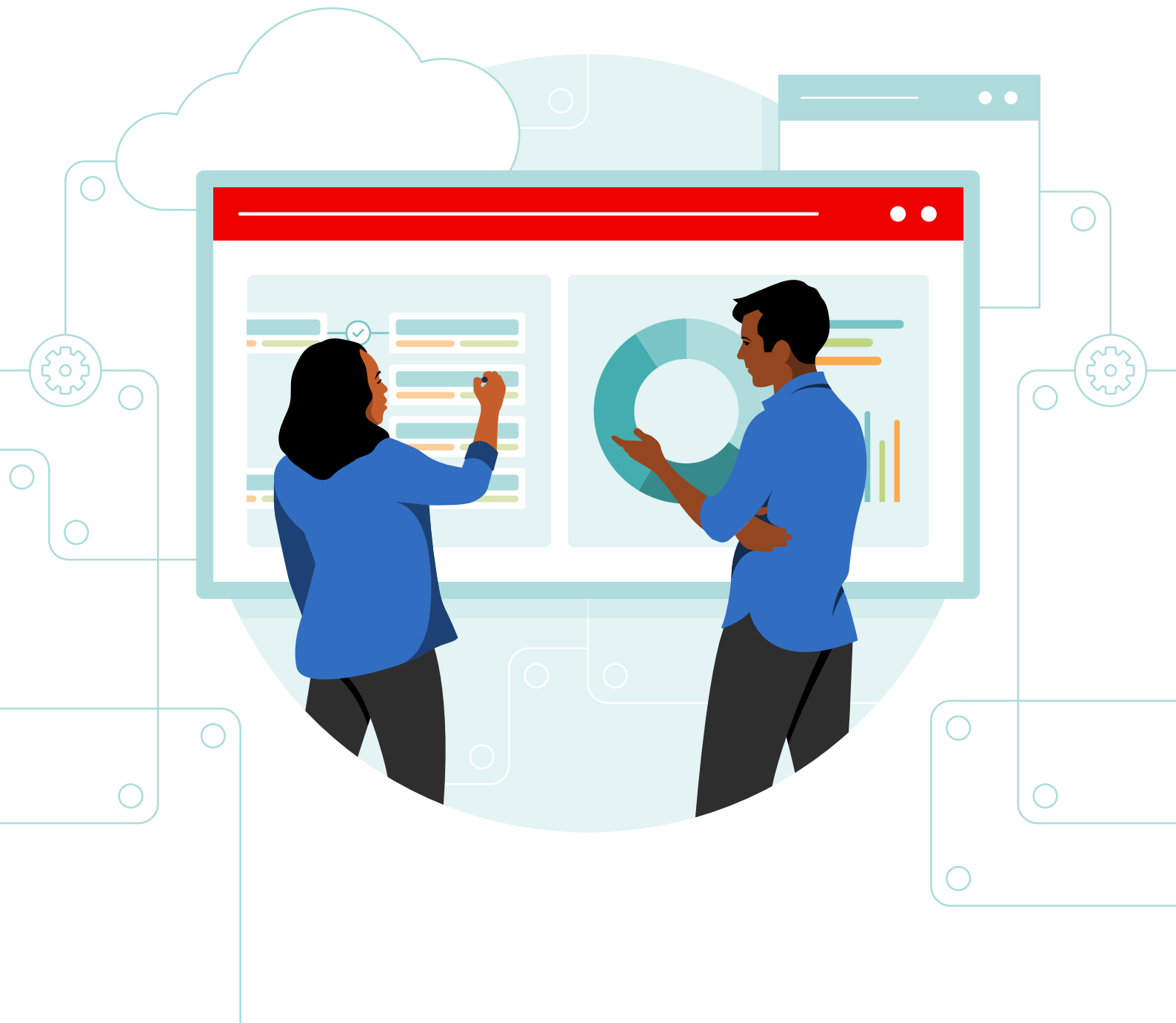


現場からの声：
システム管理者のための
IT 自動化ガイド



3 はじめに

4 第1章：IT 自動化にまつわる一般的な誤解を払拭する

- 5 私が IT 自動化を恐れていた理由
- 8 IT 自動化にまつわる 4 つの誤解を払拭
- 11 ルーチンタスクを自動化してミスを防ぎ、時間を節約する方法

14 第2章：IT 自動化のメリット

- 15 数字で見る：チームを IT 自動化に参加させる
- 18 自動化を使用して、増加したワークロードに対処する
- 21 業務を制御する：新しい働き方の創出
- 24 事例：IT 自動化で時間を節約する

27 第3章：IT 自動化のエキスパートになる方法

- 28 自動化アーキテクトの登場
- 32 IT 自動化を成功させるために必要な 8 つのスキル
- 37 Linux システム管理者の賃金プロフィールと可能性を向上させる 6 つの方法
- 41 チームを自動化ファーストの考え方に転換する 5 つの方法

44 まとめ：IT 自動化を開始する方法

- 45 IT 自動化を取り入れる：キャリアにとってプラスになる理由と、開始する方法
- 48 今すぐ始める



はじめに

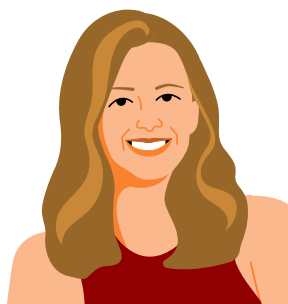
著者：Red Hat シニアプログラムマーケティングマネージャー Emma Van Sant (エマ・ヴァン・サント)

「現場からの声：システム管理者のための IT 自動化ガイド」による。本書は、IT 自動化を取り入れる際の興奮、不満、成功、課題に関する短い事例を世界中の組織とチームから集め、まとめたものです。

IT 自動化ほど、組織変革に大きな影響を及ぼすテクノロジーはなかなかありません。IT 自動化は、セキュリティとコンプライアンスを向上させ、技術的な知識が乏しいスタッフのために複雑なタスクを抽象化し、組織全体で標準化を推進し、組織の拡張を支援し、継続的デリバリーを向上させ、運用の複雑さとコストを削減します。しかし全体的な IT 自動化導入のマインドセットとアプローチを取り入れるには、このようなビジネス上のメリットと同等の課題が発生します。

根底にあるのは、自動化の中心は人だということです。Red Hat® チーム内では、「テクノロジーは簡単だが、人間は難しい」とよく言われています。新しいテクノロジーの導入に伴う技術上の困難が隠されてしまっていますが、この表現が真実の一面をうまくとらえているのは明らかです。最近の調査結果では、トレーニング不足、仕事を奪われる恐怖、組織文化が、自動化導入の普及に対する主な障壁となっていることが分かりました。この調査で挙げられた 10 個の障壁のうちの 6 つが人と文化に関する障壁で、技術的なものではありません。¹ テクノロジーの実装と導入を成功させるには、どのような場合でも人が重要な鍵となります。賛同、トレーニング、サポート、計画、安心感がなければ、自動化導入の取り組みは失敗に終わります。

本書は、このような課題に関して、組織内の全体的な自動化導入に対する文化面、感情面、実用面での障壁への対処に奮闘する Red Hat の著者による事例をまとめたものです。これらの著者は、組織内での自動化の実装に数年、またはその他のチーム内での実装と導入に関するコンサルティングに数十年の経験を積んでいます。組織の IT 自動化エキスパートとなるため最初の一步として、またそのような立場に立つことがご自分とキャリアにとってどのような意味を持つのかを考える手段として、本書が役立てば幸いです。



Emma Van Sant

Van Sant は Red Hat でグローバル IT 自動化を担当するシニアプログラムマーケティングマネージャーです。過去 7 年間はチェコ共和国、ドイツ、米国の IT 組織で働いていました。極めて困難な IT 問題に対処するためにオープンソース・ソリューションを使用する価値を広めようと熱意をもってあたっています。

¹ IT 自動化と管理に関するアンケート、2020 年 12 月、Qualtrics および Red Hat、n=838。

第1章

IT 自動化にまつわる 一般的な誤解を払拭する

- 5 私が IT 自動化を恐れていた理由
- 8 IT 自動化にまつわる 4 つの誤解を払拭
- 11 ルーチンタスクを自動化してミスを防ぎ、時間を節約する方法



私が IT 自動化を恐れていた理由

3 つの IT ロールの視点と IT 自動化に関する一般的な懸念を知る

著者: Red Hat コンサルティング シニアアーキテクト Allen Eastwood (アレン・イーストウッド)

新しいテクノロジーは往々にして多数の利点が見込まれますが、熟練した IT プロフェッショナルはこうした見込みに対して慎重な見方をすることが少なくありません。ベンダーのセールsteamがソリューションとして売り込む内容と、実際に解決しようとしている問題や課題には食い違いが生じがちです。大きな変化に対して不安が起きることが予想されます。お客様との業務経験から、特定のロールが IT 自動化に関して共通する懸念を持つことが分かりました。

エグゼクティブ

予算決定の直接的な権限を持ち、経費の承認に最も関連性が高い人物として、IT エグゼクティブが最も大きな関心を持つことが 2 つあります。まず、コストに対して何が手に入るのかを知りたいと考えています。次に、問題がどのように解決されるかを理解したいと考えます。詳細な技術レベルではなく、実装後に、プロセスが改善されたか問題が修復されたかを示すことを求めています。

幸い、IT 自動化の性質上、他のソフトウェア・ソリューションよりもこれらの懸念への対応は容易です。適切な IT 自動化プロジェクトでは、時間がかかる手動プロセスを自動化の対象とします。コストや時間は、タスクやその複雑性に依拠して、正確に見積もられるか、計算することができます。プロセスを構成するさまざまなタスクをすばやく特定し、これらを「構成要素」として自動化することで、新興に応じて改善対策を徐々に決定し、プロセスが自動化されたらより大きな利益を得られます。

IT マネージャー

予算の細部に関わることの多い IT エグゼクティブとは違い、IT マネージャーは購入決定の詳細に関わるが多いと思われます。マネージャーは作業とワークロードを管理しますが、従業員がどのように働いているかを完全に理解するための重要な知見や統計を把握していないことがあります。その結果、自動化のユースケースを見ると、ウォーターフォール型の長期計画を立てがちです。



しかし多くのユースケースでは、特にコンプライアンスやパッチなどのより複雑なものの場合、あまり知られていない多数の依存関係がプロジェクトの後半になってから発覚し、ウォーターフォール型計画の正確性と効果が失われてしまいます。効果的に計画できれば確実に不安を招きます。また、手動タスクを徐々に自動化プロセスに置き換えていく中で未知の事柄が発覚してくることが予想されるプロジェクトを開始することも、やはり不安の種となります。

以前に実施した自動化に関するコンサルティング・プロジェクトでは、マネージャーは、根本原因について知識のギャップがあると認めざるを得ないと感じていました。しかし多くの場合、このようなギャップの原因となっているのは自分たちの日々の業務でした。多くのマネージャーが、自分たちはいつも緊急事態に対応していると感じており、常にスタッフのやりくりを追われているため、最終的には人手不足でありリソース不足であるとの思いに追いやられていきます。

チームやチームの働き方について知識を増やすと、自動化プロジェクトを開始する準備が整ったと感じられます。

IT マネージャーは、サービスチケットの発行の原因となっているタスクの一部を自動化することで、こうした受動的で連続性のない管理手法の一部を削減できます。これにより、マネージャーの重点は、チームに手動プロセスのスピードアップをせかすことから、自動化を使用してビジネスニーズにより効果的に対応させることにシフトしました。

自動化に関する懸念に対処するもう 1 つの方法として、Git および Jira ツールを徹底的に活用して、誰が何をしていた、どのように貢献しているかについて、他では得られないような可視性と統計を取得することができます。この情報は、前述した未知の出来事のいくつかを発見する際に役立ち、自動化の問題の解決にどの程度の時間がかかるかを理解し、予測できるようになります。チームやチームの働き方について知識を増やすと、自動化プロジェクトを開始する準備が整ったと感じられます。

システム管理者とエンジニア

先に述べたロールが自動化に対して持つ恐怖が漠然としたものであるのに対して、システム管理者とエンジニアが自動化の導入に感じる恐怖はより本能的で、個人的であるのが一般的です。彼らの日常業務は手動のタスクとプロセスが中心で、シンプルなものもあれば複雑なものもあり、時間がかかるものもそうではないものもあります。こうした業務を自動化すると、この役目を担う人員を確保しておく必要性がなくなるのではないかと考えるのは当然の帰結です (この点についてはこの e ブックで後ほど詳しく見ていきます)。これはコンサルティングへのアウトソースにつながっていくのだろうか? 今の仕事に留まることができても、それを維持するにはどうすれば? それについて誰からトレーニングを受けられるのか? 自動化から価値を導く新しい方法をどのようにして見つければいいのか?



既知のプロセスから脱却することは、それがどれほど面倒なものであっても、実際にストレスが生じるものです。このプロセスから脱却すると自分の仕事なくなるかもしれないという不安があるなら、なおさらです。実のところ、個人的なリフレーミングが必要となることも、時にはあります。これまでインフラストラクチャ・エンジニアであったなら、今が、何か新しい職種、たとえばインフラストラクチャ開発者へと進化するチャンスです。これまでマネージャーから手動タスクをただ手早く処理するように求められていたなら、自動化のイノベーションを起こして、ビジネス価値を生み出すより直接的で目に見える役割を担当できます。

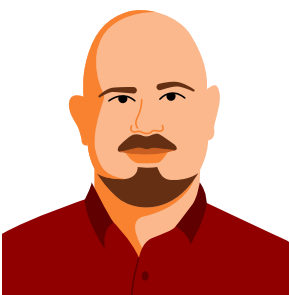
このようなタイプの変化を起こすための労力、知識、ストレスを軽視してはいけません。開発プラクティスに新しい関心を持った多くの人々にとって、実証された満足のいくパスです。それでは、どこから始めればよいのでしょうか？ **Red Hat トレーニングと認定**で、確実なスタートを切れます。スキルを構築しながら、マネージャーや他の人々に、自動化を通じてビジネス価値をもたらす能力があることを実証できます。

さらに、自動化は柔軟なソリューションで、新しいビジネスニーズへの迅速な適応を支援します。ただし、柔軟であるがゆえに、自分が1年後に何をどこで自動化しているかを予測するのは困難です。この柔軟性に適応し、コラボレーティブな自動化のチャンスを見つけ出し、スタックの理解に努め、Playbook のガバナンスシステムを確立すると、部門での自身の作業内容を充実させ、エンタープライズの自動化に向けたパスを作成できるようになります。

まとめ

自動化に関する恐怖を抑えるのに必要なのは、文化面での変化です。オープンなマインドセットを持てば、チーム間に存在する分断された敵対構造に対抗して、自動化の取り組みを迅速化して価値実現までの時間を短縮できます。知識のギャップに関する透明性を持ち、早期の失敗を受け入れることで、最終的にはチームは迅速かつ確実に価値をもたらすことができます。

→ eブック「**ビジネス自動化のための5つのステップ**」を読む



Allen Eastwood

Eastwood は Red Hat コンサルティングのシニアアーキテクトで、20 年以上の経験を積んでいます。顧客の IT システムの自動化を支援するかたわら、アイスホッケーを楽しんでいます。

IT 自動化にまつわる 4つの誤解を払拭

IT 自動化にまつわる一般的な誤解を特定して解消

著者: Red Hat OpenShift ネットワーク シニアソフトウェアエンジニア Kedar Vijay Kulkarni (ケダル・ヴィジャイ・クルカルニ)

DevOps エンジニアとしてのキャリアの中で、さまざまな日常業務の自動化に数千時間を費やしてきました。自動化を使い始めたばかりでも、ある程度の経験があっても、IT 自動化の一般的な誤解から生じる反対意見に遭遇することがあります。私の経験に基づいて、こうした誤解を解決したいと思います。

誤解 1: タスクの自動化がもたらす価値は、時間と労力を考えると割に合わない

「タスクを自動化するより、手動でやったほうが速いなら、自動化する価値がない」

時間短縮の点から、同僚や経営陣からタスク自動化について反論されることがあります。実際には、エンジニアとして行うどのジョブも自動化する価値があるのですが、時間と納品物について注意しておく必要があります。あるタスクが自動化の価値がないと思われるとき、本当は、その時点でそのタスクを自動化することができないだけだということがよくあります。ただしそのタスクの自動化は、将来的な目標として持っておきます。この見方を維持すれば、チームからの抵抗感は減少するでしょう。喫緊の目標を満たし、将来の効果を向上させるという形で自動化を提案しましょう。

誤解 2: 単発のタスクを自動化する必要はない

「1回だけでいいのに、なぜ手間をかけて自動化しなければいけないのか」

これはおそらく、私がこれまでのキャリアの中で出会ってきた最大の誤解です。

実経験を紹介すると、ある製品の顧客が固有の問題を挙げ、それには特に複雑な準備が必要でした。そのタスクの担当者が、私の同僚の DevOps チームメンバーと私に助けを求めてきました。依頼されたタスクの自動化に最善を尽くしました。タスクオーナーは検証を実施して、しばらくの間、稼働を維持できるかを尋ねてきました。その「しばらく」が数日から数週間になり、瞬く間に数カ月が経過しました。

その後、仮想マシン (VM) インスタンスを修正するという別件を受け、さらにはスナップショットの作成、最後には VM のクローンの作成というフォローアップのリクエストを受けました。VM を自動化でクリーンアップした後、最初からセットアップし直す必要がありました。



新しいリクエストが発生するたびに、私たちは担当者に自動化するように求めましたが、担当者や経営陣からの抵抗に遭いました。彼らはそれを単発のタスクだと考えていたからです。その部分だけを見れば確かに単発でしたが、実際には見えている以上に多くの労力がかかっていました。自動化にかける時間を最初に承認してくれていれば、顧客や私たちの時間をもっと節約できたでしょう。

変更の必要があるたびに、自動化のアップデートの一部として記録しておけば、全員の仕事が楽になったはずですが。これは1つの例に過ぎません。プロジェクト開始時に問題の解決にどの程度の作業が必要になるかを知ることは時に困難なのですが、適切な時点で振り返って、間に合わなくなる前に自動化に投資する必要があるかどうか評価することが大切です。

誤解 3: 自動化は機能しなくなるので、やるだけ時間の無駄

「自動化はよく機能しなくなるので、わざわざ時間をかける価値はない」

自動化が間欠的に機能しなくなるのは事実で、これは使っているうちにスクリプト言語が変化したり、自動化を通じて操作するシステムの機能が廃止されたり新たに導入されたりすることが原因です。

しかし、この理由で自動化の作業は無駄になるのでしょうか？必ずしもそうではありません。こうした事態は、**Red Hat Virtualization** 環境の構築に自動化を使用したときに幾度となく目の当たりにしてきました。自動化タスクは当初、バージョン 4.2 向けに設計されましたが、すぐにアップグレードしてバージョン 4.3 および 4.4 で構築することになりました。

自動化をメンテナンス
することで、自信が
ついてきます。

自動化のチューニングと調整を続け、さまざまなシナリオや各バージョンの癖に対応しました。その結果が **Red Hat Virtualization** デプロイメントの自動化の汎用的な組み合わせで、各バージョンの適切な入力を受け取るとエンジンとして機能し、完全にデプロイおよび構成された **Red Hat Virtualization** 環境が作成されました。

自動化のデバッグや修正に時間をかけたかという点、確かにかけました。しかし **Red Hat Virtualization** 環境の再構築が必要になったときはいつも、その作業は価値があったと断言できます。6 個ないし 7 個の環境があり、それぞれ固有のバージョン、サイズ、その他の特性がありました。新しいビルドが利用可能になったり、テスト中の乱用のために環境が陳腐化してきたりしたときは、ボタン 1 つをクリックするだけでリソースを再プロビジョニングできました。

自動化をメンテナンスすることで、自信がついてきます。自動化のメンテナンスに時間がかかるのは事実ですが、私の経験から言うと、最初に思っていたよりも効果がありました。結局のところ、メンテナンスを伴わない IT 業務はないのです。



誤解 4: このタスクを自動化することは不可能

「これを自動化するのは困難で、単純に無理だ」

時には、思っていたよりも自動化が困難なプロセスに遭遇することがあります。あるタスクの自動化がとても困難で、誰もやったことがなく、おそらく自動化できないという話を同僚から聞く (またはインターネット上で記事を読む) ことは珍しくありません。

私がキャリアを始めてまもなく、インフラストラクチャのタスクを自動化しているときに、こうした状況に突き当たりました。私には経験がなく、他の人からは無理だと言われました。当時の上司は、それでも私に挑戦して欲しいと考えました。困難だとわかっていながら、問題に取り組む時間を私に与えようとしてくれたことで、私のモチベーションは高まりました。

各種の選択肢を調べてさまざまな手法を試して PoC を作ろうと 1 カ月が過ぎた頃、その状況で一定の制約を設ければ、タスクを「半自動化」できると分かりました。つまり、手動作業が少し含まれることになります。困難だったのは事実ですが、不可能ではなかったのです。取り組んだ自動化の問題を追求するには、判断力と勇気が必要です。

もう 1 つ別の見方を紹介します。今まで誰もやったことがないのならば、自分で何か新しいものを考え出せばよいのです。それが、私がしたことです。経験を積んだ人たちの意見は間違っていないでした。自動化することはできなかったからです。ただし、彼らが言っていたのは、完全な自動化はできないということです。問題をクリエイティブな方法で解決することこそが、エンジニアの仕事なのです。

まとめ

人は自動化にまつわる誤解を信じがちです。自動化には時間がかかりますが、あなたやチームにとって価値がある可能性が高いと理解することが重要です。自動化に障害が発生して修正する必要があることがあるでしょうが、時間の節約ともたらされるメリットの増強という見返りが得られます。

タスクの自動化が困難な場合もありますが、これをイノベーションの機会と捉えてこうした課題に取り組み、獲得した知識を共有しましょう。最後に、1 回実行したタスクは、ほぼ間違いなくもう一度実行しなければなりません。自動化すれば、成功に向けて進んだことになります。



Kedar Vijay Kulkarni

Kulkarni は Red Hat のシニアソフトウェアエンジニアで、Red Hat OpenShift® ネットワーキングの仕事をしています。ソフトウェア・デファインド・ネットワークの機能、パフォーマンス、スケーリングを中心に取り組んでいます。以前は、Red Hat Ansible® Automation Platform、Red Hat Satellite、Red Hat CloudForms® (アップストリーム ManageIQ) プロジェクトの専任で、DevOps 自動化エンジニアとして、内部インフラストラクチャのデプロイメントと管理を主に担当していました。

ルーチンタスクを自動化して ミスを防ぎ、時間を節約する方法

著者: Red Hat ソリューションエンジニア Shashank Nandishwar Hegde (シャシャンク・ナンディシュワル・ヘグデ)、Red Hat OpenShift ネットワーク シニアソフトウェアエンジニア Kedar Vijay Kulkarni

エンジニア、システム管理者、DevOps のロールの職務には、大量の E メール返信、数百台のサーバーのアップグレード、データセンターやお気に入りのクラウド内のすべてのシステムのパッチ処理が含まれます。こうしたタスクのすべては、完了するまで多大な時間がかかり、時に退屈ではあっても、重要であることは常に変わりません。些細なミスであっても、システム全体のセキュリティへの脅威が生じる危険があります。たとえば、セキュリティパッチの適用に失敗すると、システムが脆弱になります。そして現実として、人間はミスを犯すものです。このような日常的なタスクを自動化して、いくらかの時間を節約し、チームのタスクを単純化できるとしたら、どうでしょうか。

自動化すべきタスクとは

反復可能なルーチンタスク

これらは、定期的に行う必要があるタスクです。たとえば、アップストリーム・リポジトリからの最新の変更を含めて、GitHub のフォークをアップデートするなどです。その他にも以下のようなタスクがあります。

- システムのバックアップログの収集
- 毎週のシステムアップグレードの E メール送信
- アップグレードの実行

シンプルながら時間を要するタスク

ラボでのシステム使用率を収集したり、システム可用性に関する E メールを毎日送信したりするなどです。継続的インテグレーション/継続的デリバリー (CI/CD) パイプラインや特定の設備など、システムで完了するまでに長時間を要するタスクがあります。このような場合は常に、何らかのアラートを設定しておくといでしょう。私たちは E メールアラートを使用して、Jenkins パイプラインが完了するたびに送信されるようにしています。このようにすると進行状況を追跡でき、その時間を他のタスクの処理に使うこともできます。

複雑なタスク

オペレーティングシステム (OS) のインストール、最新パッケージのインストール、毎週の CI/CD パイプラインがシステムで稼働することの確認など、複数のジョブの組み合わせは複雑ではなくても、データセンターやクラウド環境内の数百のシステムでこれらのステップを繰り返すことを考えてみましょう。他の例として、ロードバランサーとプロキシサーバーのセットアップや、ドメインネームシステム (DNS) インフラストラクチャのシステムへのエントリー作成が挙げられます。



仮想マシンの自動化

タスクを自動化するポイントの1つは、反復可能で日常的なタスクをスクリプト化し、デプロイできるようにすることです。これを達成すると、より価値が高いタスクや、戦略的でプロアクティブなタスクに取り組む時間をチームに割り当てることができます。

自動化の対象となる日常的なタスクの一例は、仮想マシン (VM) テンプレートを Red Hat Enterprise Linux® の新しいビルドから作り出すことです。Red Hat Enterprise Linux 7 ベースの VM テンプレートを使用していて、Red Hat Enterprise Linux 8 がリリースされたとします。すべてのテンプレートを新しいバージョンでアップデートしなければなりません。これは手作業では骨が折れ、ミスが発生しやすいタスクです。

また別の日常的なタスクは、チームメンバーから要求される VM のプロビジョニングです。自動化によって、このようなタスクの効率が大幅に向上します。さらには、チームが新しい VM の要求に使用できるセルフサービスポータルを作成すると良いでしょう。VM に固有の構成がある場合は、そのリクエストをインターセプトして適切なアクションを講じます。そうでなければ、自動化で VM の作成を処理し、VM の準備ができたなら通知 (E メールまたはチャットメッセージ) を送信します。

忘れがちなものもう1つの重要な観点は、VM を削除してリソースを保全することです。エンジニアがタイムリーにリソースのクリーンアップすることを忘れることはよくあります。管理者として、まだ使用している人が居ないか確認してから VM を手動で削除することもできますが、VM の有効期間についての一連のルールを作成することもできます。そうすれば、このルールに適合しない VM は自動的に削除されます。

自動化の導入は自分の仕事をなくすことではありません。何度も繰り返してきた作業をやめて他のことに集中できるようにすることなのです。

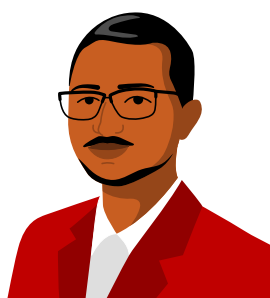
まとめ

シンプルで反復可能な、時間がかかることが多いタスクのさまざまな例を見てきました。適切に計画して、ある種のツールの力を借りれば、こうしたタスクを完全に自動化して、GitHub などの一元化されたリポジトリに保存できます。このステップを自動化すると、大幅に時間を節約でき、チームの生産性を向上できます。最も重要な点は、複雑なタスクが含まれる場合のミスを防止できることです。これは、システムのあるセグメントからの入力を受け取り、別のセグメントに渡し、完了に進む場合によく発生します。



あらゆるジョブを自動化する必要があるわけではありません。自動化が困難だと感じる場合、通常は小規模で低リスクのタスクから始めて、他のタスクを部分的に自動化しながらリポジトリを構築していくことができます。自動化は、長期的にはチームの働き方に大きな変化をもたらします。

要するに、自動化の導入は自分の仕事をなくすことではありません。何度も繰り返してきた作業をやめて他のことに集中できるようにすることなのです。



Shashank Nandishwar Hegde

Hegde は Red Hat でソリューション・エンジニアとして働いています。主に Red Hat OpenShift と Red Hat Ansible Automation Platform を扱っています。オープンソースソフトウェアとクラウド、セキュリティ、ネットワーク・テクノロジーに熱意を持っています。



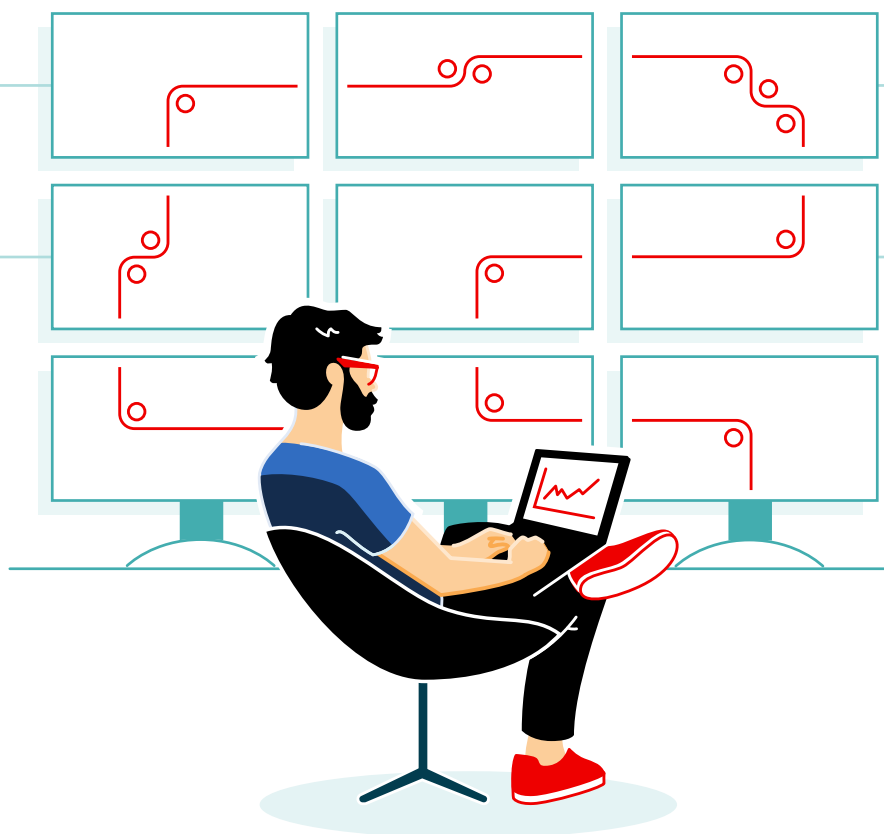
Kedar Vijay Kulkarni

Kulkarni は Red Hat のシニアソフトウェアエンジニアで、Red Hat OpenShift ネットワークと連携しています。ソフトウェア・デファインド・ネットワークの機能、パフォーマンス、スケーリングを中心に取り組んでいます。以前は、Red Hat Ansible Automation Platform、Red Hat Satellite、Red Hat CloudForms (アップストリーム ManageIQ) プロジェクトの専任で、DevOps 自動化エンジニアとして、内部インフラストラクチャのデプロイメントと管理を主に担当していました。

第2章

IT 自動化のメリット

- 15 数字で見る：チームを IT 自動化に参加させる
- 18 自動化を使用して、増加したワークロードに対処する
- 21 業務を制御する：新しい働き方の創出
- 24 事例：IT 自動化で時間を節約する



数字で見る: チームを IT 自動化に参加させる

退屈なタスクを自動化して、より魅力的なプロジェクトに重点を移す方法

著者: Red Hat Ansible Automation Platform テクニカルマーケティング シニアマネージャー
Andrius Benokraitis (アンドリウス・ベノクレイティス)、Red Hat Ansible Automation Platform
担当プロダクトマーケティングマネージャー Cindy Russell (シンディ・ラッセル)

自動化の賛同者は一般に、自動化プロジェクトの成功によってチームが達成した結果を見ることで生まれます。私たちの多くにとって、数百以上のシステム、ネットワーク、クラウド実装に対してアップデートを一貫して正しく迅速に実行できるときの最も魅力的なメリットは、おそらく「睡眠時間が増える」ことでしょう。問題が発生して夜中に叩き起こされるのを喜ぶ人はいません。

セキュリティとコンプライアンスのニーズについては、リスクを速やかに解決するようチームが圧力を受けているのを何度も見てきました。適切な修正を自動化 Playbook で作成できれば、グローバル・インフラストラクチャ内の影響を受けるすべてのソリューションに対して実行できます。このソリューションは迅速で効果的なので、リスクは速やかに修復され、同じことの繰り返しを人が行うことでたまに紛れ込むミスは発生しません。

メリットは他にもあります。たとえば、価値を付加する重要な優先事項に取り組む時間を増やせることや、経験やスキルを発展させることなどです。ここでは、私がチームメイトに IT 自動化を売り込むときに使った、いくつかの統計を紹介します。

数時間を数分に短縮した事例

自動化のメリットは劇的に現れることがあります。適切な自動化ソリューションを導入して、数時間を数分に短縮した事例は枚挙にいとまがありません。2つの例を見てみましょう。

20 時間 > 30 分

Blue Cross and Blue Shield of North Carolina では、仮想マシン (VM) やサーバーを 1 つプロビジョニングするたびに 20 時間 (およそ 10 営業日) 以上の作業が必要となり、マネージドサービスのプロバイダーへの料金も発生していました。Red Hat Ansible Automation Platform により、サーバー 1 台の VM プロビジョニングは 30 分で完了できるようになりました。1,000 台の VM をすばやくプロビジョニングする必要があった場合、そのプロセスは 3 日未滿で完了します。自動化されていなかったときは、同じ作業を完了するにはスタッフ 12 人で 16 週間を要すると見積もられていました。²

² Red Hat 事例、「保険会社が社内の IT 自動化により 7 万時間分の作業を削減」、2021 年 5 月。



20分>1分未満

米国コロラド州の**デンバー市**は、Ansible Automation Platform の導入前は、個々の Microsoft Teams 環境の作成に 20 分かかっていました。Ansible Automation Platform の導入後は、同じプロビジョニングプロセスの所要時間は 1 分未満になりました。コロナ禍によって Microsoft Teams の利用が 514% 増加しましたが、自動化が整備されていたことで、大幅な時間の節約につながりました。³

IT 部門がこのようなニーズにすばやく対応する必要がある場合、自動化によってジョブが迅速で効率的になります。同様の事例は、Red Hat の**事例のページ**でご覧になれます。

自動化のスキルが求められている

誰もが自分のスキルの向上を常に望んでおり、これはすべての技術プロフェッショナルにとって基本的なニーズです。自動化は、レパートリーに加えるべき重要な新しいスキルです。いくつかのトレンドを見ていきましょう。

Ansible は好まれるテクノロジーの中で 9 位にランキング

自動化スキルはアンケートや動向に関するレポートで常連となっています。2020 年、**Stack Overflow Developer Survey** では、Ansible は最も好まれるツールのリストで 9 位に選ばれました。⁴ その前年、Indeed.com の **Hiring Lab レポート** では、Ansible は急成長スキルの 3 位となり、2014 年から 2019 年にかけて 1,300% の成長を遂げています。⁵

コロナ禍で
業務に影響が及ぶなかで、
リモート管理に役立つことから、
自動化はエンタープライズ向け
テクノロジー戦略の最前線に
あります。

40,000 を超える LinkedIn のジョブが Ansible スキルに言及

コロナ禍で業務に影響が及ぶなかで、リモート管理に役立つことから、自動化はエンタープライズ向けテクノロジー戦略の最前線にあります。企業は自動化のメリットを直に確認し、このことからエンタープライズ IT 自動化および管理戦略が迅速になりました。インターネットを検索すると、自動化の成長トレンドについて多数のアナリストの見解が見つかります。この成長は求人にも明らかに現れています。LinkedIn の検索では、Ansible を職務内容に含む求人情報がおよそ 40,000 件見つかります。⁶ Ansible 自動化スキルが望まれていることは明らかで、取得しておくの良いスキルです。

価値のあるプロジェクトを提供する

ビジネスのサポートを最適にするには、部門の壁を越えた IT チームが団結して重要な取り組みから成果を出す必要があります。たとえば、デジタル・トランスフォーメーション・アプリケーション、ハイブリッドクラウド・プラットフォームの拡張、テクノロジースタックのモダンイ

3 Red Hat 事例、「**デンバー市、Red Hat Ansible Automation Platform でリモートワークをサポート**」、2021 年 4 月。

4 Stack Overflow、「**2020 Developer Survey**」、2021 年 7 月にアクセス。

5 Flowers, Andrew、「**Indeed Tech Skills Explorer: Fastest-Rising Tech Skills**」、Indeed.com Hiring Lab、2019 年 11 月 26 日。

6 LinkedIn の Ansible を職務内容に含む求人は 2021 年 6 月 23 日時点で 40,600 件。同日の検索で「Ansible engineer」の求人はさらに多かった。



ズ、エッジまたは IoT (モノのインターネット) 実装などです。このような取り組みには、新出のテクノロジーの学習や使用が伴うことが多く、顧客や企業のリーダーなどの主なステークホルダーに対して価値を実証したり提供したりするのに役立ちます。このような高度なプロジェクトに集中する時間が増えれば、プロフェッショナルとしての開発や成長につながります。

自動化は、機械的なタスクの自動化や、価値の低い作業の完了を支援できるので、より魅力的なプロジェクトに集中できるようになります。たとえば、500 のシステムにパッチを適用してセキュリティの脆弱性を修復しなければならないとします。前述のお客事例で紹介したように、自動化によって所要時間が大幅に短縮されます。その結果、より革新的なプロジェクトに集中する時間が増え、これは個人にとっても企業にとっても良いことです。

手動タスクから解放され、多数の自動化スキルを身に付けたところで、自動化アーキテクトとしてのキャリアを前進させたいと思うかもしれません。このハンドブックでは、「自動化ファースト」のコミュニティを形成し、企業の自動化戦略を先導するためのベストプラクティスの詳細を説明します。

あなたや会社が自動化のメリットを数値で確認できるようになることを願います。学習を首尾良く進めて、自動化の賛同者として **AnsibleFest** にご参加いただければ幸いです。

お勧めの学習資料:

- **Red Hat トレーニングモジュール** (無料)
- **オンデマンドまたは今後の Web セミナー**
- **Ansible Automation Platform の動画**
- **最初の自動化プロジェクトを構築して実行**できる、無料トライアル



Andrius Benokraitis

Benokraitis は Red Hat Ansible Automation Platform のテクニカルマーケティング担当シニアマネージャーです。IBM、Nortel、Cumulus Networks などの企業で、コンピュータ・ソフトウェア業界における 20 年以上の経験があります。ネットワーク自動化、エンタープライズ向け Linux、ビジネスアナリティクス、テクニカルライティング、戦略的アライアンスの分野のスキルを有しています。



Cindy Russell

Russell は Red Hat Ansible Automation Platform のプロダクトマーケティングマネージャーです。技術担当者やソフトウェア製品向けのその他のマーケティングプログラムの開発に長年従事し、自動化、分析ツール、データベースおよび関連する機械学習テクノロジー、開発者ツールなどを対象としています。IBM、Sun Microsystems、複数のスタートアップ企業および小規模な独立系ソフトウェアベンダー企業での業務経験があります。

自動化を使用して、 増加したワークロードに対処する

退屈で手間のかかる、日常的なタスクにうんざりしていませんか？
自動化によって効率が向上し、新しい革新的な機会に集中する時間を
確保できます。

著者: Red Hat シニアコンサルタント Ricardo Gerardi (リカルド・ジェラルディ)

多くの組織で情報テクノロジーのフットプリントが拡大するにつれて、システム管理者のロールも拡大しています。先進的なシステム管理者は、物理的なマシンやオペレーティングシステムのみならず、無数の仮想システム、クラウド環境、ネットワークデバイス、コンテナワークロードを取り扱います。管理されるシステムの複雑さと台数は、小規模企業であっても、一般に数年前より1桁以上増加しています。

システム管理者の役割は、このようなすべての異種システムをデプロイ、メンテナンス、アップデートし、正しく機能させ、顧客と会社の両者がこのテクノロジーからメリットを受けられるようにすることです。これが動的なジョブであり、やりがいがあるエキサイティングなタスクを実行するチャンスが得られることはまちがいありません。また、新しいスキルを学習したり、刺激的なプロジェクトに取り組んだりすることにもつながります。

ただし一般的に、ジョブの性質や管理するシステムの数から、複数のありふれたタスクも実行しなければなりません。そうしたタスクの例として、サーバー群へのマイナーアップデートの適用や、一部のログのクリーンアップが挙げられます。アップグレードの準備として、またはセキュリティ監査の対応として、インストールされているソフトウェアのバージョンの報告を求められることもあります。これらのタスクはシンプルですが、処理が必要となるデバイスの数と種類を考え合わせると、完了までに長時間を要します。このようなタスクに時間を費やしていると、より価値がある、またはより刺激的な活動に取り組む時間が削られてしまいます。

繰り返される退屈なタスクを処理していると、やりがいを感じなくなったり、押しつぶされそうになったりしませんか？そう感じるのは、あなたひとりではありません。これまでの年月で何度も同僚からそういう話を聞かされてきましたし、私自身もそう感じたことがあります。

変化を起こす

それでは、状況と時間の使い方を改善するには、どうしたらよいでしょうか。その答えは複雑で、場合によっては企業が体系的に対処する必要があります。その他の問題については、技術的な専門知識を使用して、これらの繰り返される日常的なタスクを自動化し、働き方を改善できます。



IT プロセスの自動化を検討するとき、通常は、自動化アーティファクトを開発して大規模または複雑な問題を解決するのにかかる時間と労力を評価します。この方法には多くのメリットがあります。シンプルな繰り返しタスクに自動化を適用すると、同じメリットを実現できます。特に、自動化を再利用して数百または数千のシステムを管理できる場合はなおさらです。

メリットの一部を紹介します

1. **効率**: 多数のターゲットに対してより効率的にタスクを実行し、大変な仕事をコンピュータに肩代わりさせます。
2. **標準化**: タスクを一貫した方法で実行し、いつでも同じ結果が出るようにします。たとえば、あるデバイスを小文字で設定し、他のデバイスを大文字で設定したために、後でトラブルシューティングが困難になる問題が発生するなど、一般的な不注意によるミスを防止します。
3. **ミスの減少**: システムを操作する人数が少なければ、障害につながる構成ミスや事故の確率が低下します。たとえば、SQL クエリの DELETE 文や UPDATE 文で WHERE 句を抜かしてしまったことはありませんか？
4. **個人の達成感**: 達成感や知的なやりがいのあるタスクに取り組むほうが、一般には仕事の満足感が高くなります。

どこから手を付けるか

自動化のコンセプトをシンプルなタスクに適用するのも、自動化を始めるには良い方法です。短期間で成果を挙げて即座にメリットを実現し、その後、より複雑な問題を自動化するための学習と準備を進められます。実際、私は数年前に、この方法で Ansible Automation Platform を使い始めました。私は Ansible Automation Platform を使用して、管理している数百台のサーバーを定期的にチェックするスクリプトを置き換えました。初期構成にかかった時間は数時間で、その後、ある環境を使用してこれらのチェックとその他の多数の小さなタスクを自動化し、これらをすばやく一貫して実行できました。

私の考えでは、これは自動化へのプロセスを開始する良い戦略です。定期的に行われ、長い作業時間がかかるか実行が面倒な問題やタスクを探し、自動化ソリューションを選択し、適用してこの問題を解決し、同時にソリューションおよび自動化の内部の動作を全体として学習します。

自動化のコンセプトを
シンプルなタスクに適用するのも、
自動化を始めるには良い方法です。



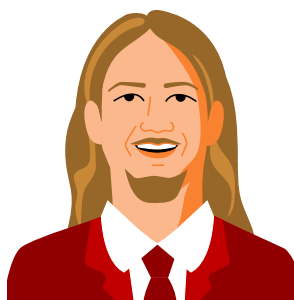
どこから手を付けてよいかわからない場合のために、最初の自動化プロジェクトに適した候補となるタスクの例をいくつか紹介します。

- 接続性テスト
- メンテナンスのパッケージ：パッケージのインストール/アップデート/削除
- インストール済みのアプリケーションのバージョンの報告
- ユーザーの管理
- ユーザーパスワードのリセット
- コアファイルやログなど、使われていないファイルのクリーンアップ
- システムまたは構成のバックアップまたはリストアの実行

そこからの針路

シンプルな繰り返しタスクを自動化すると、こうしたタスクを効率的に完了でき、その時間をより重要で、おそらくはより刺激的なプロジェクトの取り組みに割り当てることができます。

自動化へのプロセスは困難ではあるかもしれませんが、しかし、お勧めできると断言します。ぜひお試しになり、会社と自身にとって数多くのメリットを手に入れてください。



Ricardo Gerardi

Ricardo Gerardi は Red Hat Canada のシニアコンサルタントで、Red Hat Ansible Automation Platform および Red Hat OpenShift による IT 自動化を専門としています。通信分野での経験があり、TELUS でシニアアーキテクトとして勤務しました。また、IBM Brazil および IBM Canada ではネットワーク管理ソリューションのシニアコンサルタントおよびプリセールス・スペシャリストとしての経験もあります。

業務を制御する： 新しい働き方の創出

提供時間の短縮、人的ミスの低減、新しい仕事への時間の確保のため、日常的なタスクを自動化して効率化するツールの開発事例

著者：Red Hat テクニカルアカウントマネージャー Thomas Tuffin (トーマス・タフィン)

あなたは運用チームのシステム管理者で、アジャイル手法を使用して作業し、スクラム、かんばん、さらには ScrumBan に基づくフレームワークに従っています。チームは常に進化する環境を担当しています。日々のサポートチケットに加えて、納期までに完了しなければならない、ステークホルダーへの納品物があります。ワークロードは厳しく、バックログは積み重なるばかりです。このように強いプレッシャーのかかる状況では、ステークホルダーへ即座に価値をもたらすタスクが最優先され、最も注力されます。即座に価値を生まないタスクの優先度は低くなります。残念なことに、自動化ツールとフレームワークの開発は優先度が低いグループに振り分けられることが一般的であるため、チームは手作業での処理を続けることになります。こうした状況に心当たりがあるのは、あなただけではありません。

自動化の開発にフォーカスし直すのは、簡単にできるとは限りません。うまくいけば、チームから強い支持を得て、経営陣からも前向きに強化してもらえます。しかし、優先度については強い抵抗を受けることがあり、それには正当な理由があります。成果を待っているステークホルダーがおり、あなたの主な責務はステークホルダーが必要とするものを期日までに手渡すことなのです。

運用チームとしては、おそらく他のチームやサプライヤーとの依存関係があるでしょう。チェーンの上流で遅れがあると、ドミノ倒しで下流まで影響が波及し、納品が遅れてしまいます。スクラムで作業するチームにとって、スプリントの最後までに完了しなかったストーリーやサブタスクがそれ以降のスプリントに持ち越されることは珍しくありません。このバックログのためにチームへのプレッシャーが高まり、以降のプロジェクトが遅延し、環境の改善を目指すタスクは先送りされてしまいます。このような状況では、チームは制御能力を失ってしまったと感じ、長期的なソリューションにあまり関心が向けられないため、チーム内に不満もたまります。改めて言いますが、このような状況に陥っている運用チームは珍しくありません。

自動化を開始する方法

それでは、何ができるのかを考えてみましょう。当然ですが、チーム、プロダクトオーナーまたはプロジェクトマネージャー、必要に応じてマネージャーと、これをテーマとして話し合うことから始めましょう。経営陣の関心を引くには、リソースを自動化の開発に投入することでどの程度の時間と労力を節約できるか、いくつかの例を挙げるのも良い方法です。まずは、今



チームでタスクにどの程度の時間がかかっているか、これらの機能の自動化にどの程度の時間を要すると思うか、自動化の後にはタスクにどの程度の時間がかかるかを概説した提案書を作成するとよいでしょう。

自動化できるタスクに対して余分に時間を割り当てることから始めるのも良いアイデアです。構成管理データベース (CMDB) にアイテムを追加するプロセスを効率化するなど、比較的小さなジョブから始めます。良い CMDB にはアプリケーション・プログラミング・インタフェース (API) があるので、この API を活用するためにシェルまたは Python スクリプトを作成することが、サーバーのデプロイと管理を自動化するワークフローにとって不可欠です。API を理解し、この API を操作するスクリプトを作成したら、次は他のスクリプトが使用できるライブラリの構築に進みます。ライブラリを構築することで、チームが API を操作する方法を標準化できます。また、以降のすべてのスクリプトで、API にアクセスする必要があるときにいつもこのライブラリを使用できるので、求める機能のスクリプトがすでにあるのに気づかず新しく作成してしまうことも防げます。最初に形を作るのに少しばかりの作業が必要になるかもしれませんが、スクリプトとツールの将来的な開発の時間と労力は節約されます。

CMDB API への標準化されたアクセスを有するライブラリとアセット情報を引き出せるスクリプトがあれば、サーバーのデプロイメントプロセスの一部の自動化を開始できます。すでに PXE (Preboot Execution Environment) およびキックスタートファイルをデプロイ手順の一部に使用している場合は、作業の大部分を自動化してみても良いでしょう。これを実現するには、任意の言語で記述したスクリプト、または **Ansible Playbook** を使用できます。テンプレートモジュールなどの Ansible の組み込みモジュールの一部を採用すると、PXE ブートファイルおよびキックスタートファイルを CMDB から取得したデータを使って生成できます。Ansible にはさまざまなメーカーのアウトオブバンド管理インタフェースに接続するモジュールもあるので、アウトオブバンドのホスト名、IP アドレス、電源管理の設定などのタスクが簡単になります。

もちろん、他のツール、カスタムスクリプト、各種のオープンソースおよびプロプライエタリー・ソフトウェア・ソリューションを使っても、同じ結果を得ることができます。課題に対処する方法は1つだけではありません。同じことがサーバーのデプロイメントにも当てはまります。スキルとツールを自在に使用してプロセスを効率化し、繰り返しタスクに要する時間を削減します。

**スキルとツールを自在に
使用してプロセスを効率化し、
繰り返しタスクに要する時間を
削減します。**

いくつかの小さなタスクを自動化したら、何らかの好ましい効果が出てくるはずです。たとえば、使いにくい CMDB インタフェースをクリックして情報を入力したり検索したりする必要はなくなります。さらに、サーバーのデプロイメントプロセスの一部が自動化されます。最後に、自動化にはもう1つの大きなメリットがあります。それは、人的ミスによる問題の削減です。IT 業界では、さまざまなシステム障害、停止、セキュリティ侵害の原因は多くの場合において人的ミスであることがよく知られています。⁷ 自動化するタスクがどれほどシンプルでも、特に日常的なタスクの場合は、この点で軽減に役立ちます。

⁷ Tessian およびスタンフォード大学。「[The psychology of human error](#)」、2020年7月。



環境を制御する力を取り戻す

多くのチームが自動化ツールの開発に携わるようになるにつれ、ツールのエコシステムは有機的に成長していくでしょう。この成長により、コードの品質、保守性、責任などの課題が生じます。早い段階でガイドラインを制定して、一定の基準を満たし、作業方法に従っているかを確認することが重要です。この作業の一部は、Git や Gerrit などのバージョン管理システムおよびツールを使用して変更を追跡し、チーム内のコードのコラボレーションをサポートすることで達成できます。

繰り返しタスクを処理するための健全なツールセットを用意できたら、チームが標準の手段として自動化を使うようになります。何か新しいことを計画するときには、まず「これは自動化できるか?」という質問から入るようになります。自動化を多く実装すればするほど、現状を打破して環境を制御する能力を取り戻そうとする意識が強く働くようになります。日常的なタスクを自動化して効率化するツールを開発すると、ステークホルダーへの納品時間が改善されて人的ミスが減少するほか、あなたやチームが環境の将来に取り組む時間を増やすことができます。



Thomas Tuffin

Tuffin は Red Hat のテクニカルアカウントマネージャーです。スウェーデン在住のオーストラリア人で、オープンソースソフトウェアに情熱を傾け、ブロックチェーンなどの新興テクノロジーに高い関心を持っています。いつも何か (どこか) 新しいものを見つけようとしており、旅行をしているかプロジェクトに取り組んでいます。

事例: IT 自動化で時間を節約する

新しい刺激的なことに取り組む時間を作り、数日かかっていた
タスクの処理時間が自動化で数分に短縮するのを目撃する

著者: Red Hat OpenShift ネットワーク シニアソフトウェアエンジニア Kedar Vijay Kulkarni

2020 年、私は、最新の **Red Hat Satellite** ビルドの新しい VM イメージを作成するプロセスを自動化するチームで働いていました。私たちの目標は、VM デプロイメント、スナップショット、クリーンアップ、テンプレート作成を自動化することでした。簡単のように思われますが、大変な作業でした。チームの時間を節約するには自動化が必要であることは明らかで、自動化インタフェースとして Red Hat Ansible Automation Platform を選びました。このストーリーはここから始まります。

Red Hat Ansible Automation Platform を使ったことがある方なら、便利に使うためにはまず多くの設定をしなければならないことをご存じでしょう。たとえば、ログインと認証をセットアップし、プロジェクト、認証情報、インベントリ、インベントリソース、ジョブとワークフローのテンプレート、通知、スケジュールなどを準備する必要があります。こうした作業が、私が Red Hat Ansible Automation Platform **Configuration-as-Code ATCasC** の作成を支援した理由です。

この自動化プロジェクトは、Ansible Automation Platform ユーザー・インタフェースで実行できるあらゆるものを YAML シリアル化言語に変換できます。設定は 1 つの Playbook コマンドで実行でき、新規インストールした Ansible Automation Platform 全体を完全に機能する状態へと一気に持っていきます。

これは大きな成功です。なぜでしょうか。構成さえ記述してしまえば、**Configuration-as-Code** 手法を使用して新しいインスタンスを起動させるまで 30 分とかかりません。このアプローチを使用する前は、新しいインスタンスをデプロイ、セットアップ、構成してプロダクションで使える状態にするには、1 日以上が必要でした（誰が担当するかや、担当者の知識レベルによって多少の上下はあります）。

Config-as-Code 手法を開発するまでは、手動でデプロイメントするのに 1 - 3 時間、それを構成するのにその日いっぱいがかかっていました。プロセスをスピードアップするには、2 人がかりになります。たとえば、5 つの認証情報、2 つのインベントリ、2 つのインベントリソース、20 - 40 のジョブテンプレート、5 - 10 のワークフローがある 1 つのプロジェクトでは、マウスを使ったユーザー・インタフェースでは作成に数時間かかります。苦勞しながらなんとかやり遂げたとします。このインスタンスを失ったとしたらどうなるでしょうか。何も構成を作成していなければ、これを再度作成するには、純粋に自分の記憶かチームのドキュメントに頼ることになります。

これが、構成をまず記述することが大切だと理解した理由です。そして、構成を正しく記述することが、チームにとって学習の機会となりました。その理由は、構成には標準のプログラミング言語がなかったことです。そのため、チームは YAML 構成のスキーマを学習する必要がありました。この学習段階を通過すると、効率が向上しました。

準備時間を自動化したので、障害が発生した場合、バックアップを実行し、適切な構成ファイルで即座に実行できると確信できました。それでは、YAML 構成の完全な作成完了とテストについてはどうでしょうか？

この課題の全体像を捉えるため、必要な手順をまとめると、Ansible Automation Platform でジョブテンプレートとして実行される新しい Playbook を作成する場合、適切なプロジェクトをプロジェクトの YAML ファイルに追加してから、適切な認証情報、インベントリ、ジョブテンプレートを正しいファイルに追加する必要があります。作成するコードは少なくとも約 50 行になります。このコードを理解して記述するには、30 分 (自分の作業内容を把握している場合) から 3 - 4 時間 (初心者の場合) が必要です。

コードを記述するプロセスは、経験を積まなければ迅速になりません。しかし費やした時間だけの価値はあり、再現性と一貫性を身に付けられます。Infrastructure-as-Code のすべての長所と短所を適用できます。

次に行うのは、構成 (コード) のテストです。これを行うため、私のチームでは 2 時間程度をかけて、プロダクション・インスタンスと同様で、かつ提案されたすべての変更を含むテストインスタンスを起動していました。そして、マージ要求を完全にテストするにはどのようなジョブが必要かを突き止めます。最後にマージを行います。この負担のかかる作業には合計で 1 日か 2 日を要していました。

このテストを自動化するため、GitLab 継続的インテグレーション (CI) を使用した自動化アプローチを考案しました。この自動化処理では、新しいプルリクエスト (PR) がオープンされるたびに、GitLab CI がこの PR 用に新しいテストインスタンスを作成します。自動化により、2 - 4 時間 (インスタンスのデプロイ担当者による) が削減されました。デプロイは GitLab が実行するので、削減される時間もその分増加しました。

次の課題は、プルリクエストをどのようにテストすべきかの判断です。小さな PR なら、何をテストするかはすばやく容易に理解できます。しかし複雑な PR の場合は関連するファイルが 10 を超え、しかも master ブランチにマージする前に PR が適切にテストされていないと、何に障害が起きるか予想が困難でした。そして、プロダクション・インスタンスは master ブランチのコードに基づいて実行されています。

この問題を克服して、PR を分析してテストするための時間を節約するため、Ansible Genealogist という新しいプロジェクトを考案しました。このプロジェクトでは PR を数分で検査し、テストが必要な内容を文書化します。

コードを記述するプロセスは、
経験を積まなければ
迅速になりません。



タスク	手作業に要する時間	自動化を使用した場合の時間 ⁸
新しいプロダクション対応の Ansible Automation Platform インスタンスのデプロイ	約 1 - 2 日	30 - 45 分以下
プロダクションに移行する前に新しい構成をテストするためのテストインスタンスのデプロイおよび構成	約 4 - 6 時間	30 - 45 分以下
新しい各 PR についてテストが必要な内容の判定	約 1 - 2 時間	5 - 10 分以下
テストの実行	約 2 - 6 時間 (複雑な PR の場合はこれ以上)	5 分以下 (自動化テストスクリプトを起動して、後で結果を確認するだけ)
何らかの障害により、プロダクションで実行していたプロダクション・インスタンスを失ったために再デプロイする	見積りなし、障害なので全員が総出であたる (チームメンバーが解決するために何をして何を管理すべきかを知っている場合は、約 1 - 2 日)	30 - 45 分以下
新しいジョブテンプレートの追加や認証情報の更新など、プロダクションへの変更を行う	大変な作業。何か間違ったことをすれば、悪い結果になる。プロダクションの更新前に変更をテストすることにすれば、1 日程度の作業が見込まれる	30 分以下。 変更は PR プロセスの一環としてテストされるため。プロダクションへの移行は基本的に CI/CD の CD 部分

ご覧のように、自動化によって、タスクの処理時間を単位日から分単位に短縮することができました。そして、自動化によって私たちの仕事が無くなったかといえば、そんなことはありません。自動化するタスクは増え続けているからです。私たちのグループの目標は、仮想マシンの標準的な管理タスクである、デプロイメント、テンプレート、スナップショットなどを自動化することでした。時間の節約はこのプロジェクトの重要な部分でした。障害発生時のために、繰り返し可能なイベントの作成も目指していました。自動化とテンプレートにより、障害復旧の状況における効率を大幅に向上させるチャンスが得られました。



Kedar Vijay Kulkarni

Kulkarni は Red Hat のシニアソフトウェアエンジニアで、Red Hat OpenShift ネットワークと連携しています。ソフトウェア・デファインド・ネットワークの機能、パフォーマンス、スケーリングを中心に取り組んでいます。以前は、Red Hat Ansible Automation Platform、Red Hat Satellite、Red Hat CloudForms (アップストリーム ManageIQ) プロジェクトの専任で、DevOps 自動化エンジニアとして、内部インフラストラクチャのデプロイメントと管理を主に担当していました。

第3章

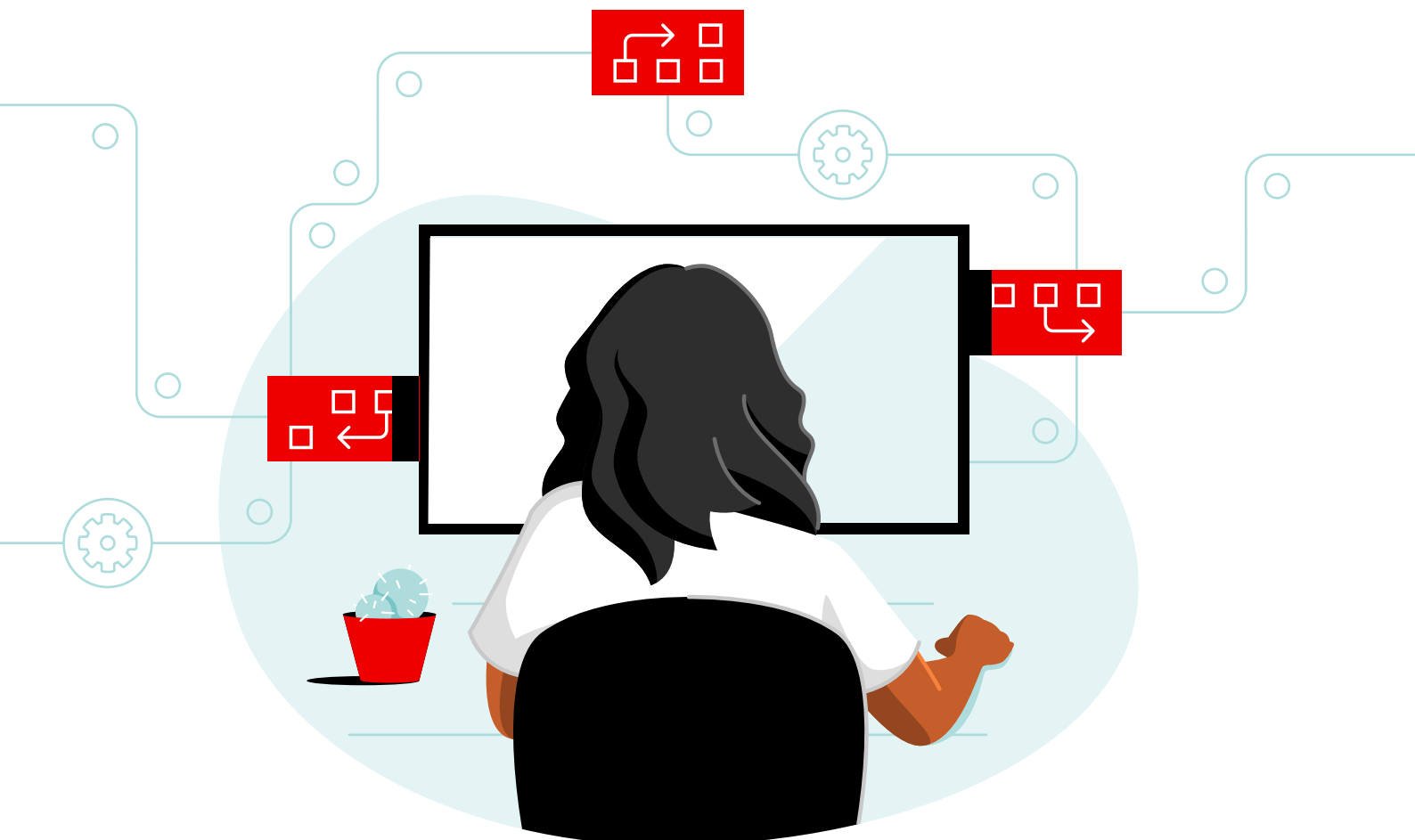
IT 自動化の エキスパートになる方法

28 自動化アーキテクトの登場

32 IT 自動化を成功させるために必要な 8 つのスキル

37 Linux システム管理者の賃金プロフィールと可能性を向上させる 6 つの方法

41 チームを自動化ファーストの考え方に転換する 5 つの方法



自動化アーキテクトの登場

IT キャリアを進展させ、自動化アーキテクトとしての地位を確立するために使えるヒント

著者: Red Hat テクニカルアカウントマネージャー Joseph Tejal (ジョゼフ・テハル)

私たちの多くが、その出現を目の当たりにしました。自動化の課題の煙と闇の中から登場した新しいロール、それが自動化アーキテクトです。

自動化は 2015 年以来、話題を集めてきました。Ansible などのオープンソース・プロジェクトが台頭し、普及を進めてきました。多くの期待や関心が寄せられましたが、それに劣らず懐疑的な意見もあり、運用センターでは不安が高まりました。現在まで時を進めると、自動化は今や大半の組織にとってテクノロジー戦略の中心的存在です。状況は常に進化し、また、さらに複雑化しています。

自動化によるイノベーションの促進に関するこの [Forrester の調査](#) では、企業が数ある目標の中でも自動化の取り組みを優先していることが分かります。⁹ 企業は計画において自動化が持つ重要性を認識し、ビジネスのあらゆる側面を自動化しようとしています。これを成功させるには、誰かがこの業務の音頭を取らなければなりません。

それでは、時間を巻き戻して、自動化アーキテクチャがどのように進化して現在のような形になったかを振り返ってみましょう。

自動化を始める

初めは、運用チームと個人が自動化は容易で面白いと認識しました。彼らの大半は開発の経験を持ちませんでしたが、一部の日常的なルーチンタスクを自動化する方法を学びました。Ansible などの習得しやすいツールによって、熱意に満ちたメンバーは容易に学習できました。システム管理者や、インフラストラクチャ・エンジニアおよびデータベースエンジニア、さらにはサービスデスクの担当者までもが、自動化を学習しました。

休憩中の雑談や廊下での立ち話から、経営陣を納得させるための実践的なデモまで、関心は高まっていきました。マシンのプロビジョニング、アプリケーションの再起動、パッチとメンテナンスの実行など、自動化する作業は増え続けました。人々はコラボレーションしてアイデアを組み立て、優れたものを各自の分野で実行させて全員が連携できる方法を話し合うようになりました。

しかし、自動化の個々の小さなピースをつなぎ合わせて使おうとし始めたとき、チームは問題を抱えるようになりました。その理由はおそらくおわかりでしょう。

9 Forrester Consulting, Red Hat 後援、「[エンタープライズオープンソースのオートメーションがイノベーションを推進](#)」、2020 年 7 月。



課題に直面する

程なくして、状況は複雑化してきました。対象となるユースケースはまだ他にもあり、自動化のルーチンとワークフローを連結する必要が出てきました。徐々に課題が見えてきましたが、そのうえ悪いことに、成功を見た管理職から、一部の人がこれまで以上に多大な要求を突きつけられたのです。

ここでは、自動化によって発生する可能性のある問題の一部を紹介します。

- 競合する優先事項と目標
- 標準、ポリシー、ガバナンスの欠如
- 分断された組織文化に起因するボトルネック
- 一部の必須スキルの欠如
- 実装のセキュリティ上の懸念
- 再利用性、スケーラビリティ、コントロールの検討不足

多くの運用チームが、こうした課題をチャンスに転換しています。これらの課題の一部を提示された当初は、人は混乱して、当事者意識や責任感がないために方向を見失っていきます。しかし、このような課題は、環境やタスクがどれほど複雑でなくても、自動化プラクティスを構築して成功させるための主導的立場を獲得するチャンスをもたらします。

自動化アーキテクトが問題を切り抜ける方法

自動化に対する全体的なアプローチを導入するには、インフラストラクチャ自動化成熟度に関するこの [Forrester の調査](#) で概説されている、基本的なコンピテンシーを見つけ出せる推進役が必要です。¹⁰

以下のようなコンピテンシーがあります。

- 戦略
- プロセス
- 優先順位
- 人材
- 構造
- 運用

組織でプロセスを進めていくには自動化アーキテクトが必要です。自動化アーキテクトはどの方向に進むべきか、どうしたら目標達成を成功できるかを理解しています。自動化アーキテクトの役目は、前述した課題に対処して、誰もが自動化の価値を完全に活用できるように支援することです。

[Red Hat の自動化アーキテクトのハンドブック](#) に記載されているように、組織を効果的に主導するために必要なタスクには以下のものがあります。



自動化の現状を把握して目標を設定する

このプロセスからアーキテクトは、どの既存のツールと機能が便利で、他に何が必要かを、はっきり認識することができます。達成可能な目標を設定して優先度を設定することが必須です。これにより、全員が進捗を確認して、後退や迂回があったとしても、正しい方向に進むことができます。この取り組みにより、全員が目標を念頭に置いて前進し、基本的タスクに集中できます。

統一とコラボレーションを促進する

埋まらないギャップ、異なる意見、対立する働き方は必ず出てきます。自動化アーキテクトは全員をまとめて、ビジネス目標と優先事項に向けた取り組みに足並みを揃えて集中させなければなりません。こうした取り組みにより、要件をしっかりと定義し、誰もがそれを参照して自由に議論できるようにできます。オープンでコラボレーティブなアプローチとツールを使用して、チームが協力して働くように奨励します。分断されたグループの壁を取り払うアクティビティを取り入れ、集団としての成功や、多くの人やチームを巻き込んで実現された成功を表彰します。

自動化アーキテクトは全員をまとめて、ビジネス目標と優先事項に向けた取り組みに足並みを揃えて集中させなければなりません。

自動化資産と機能を向上させる

自動化インベントリーの計画を完了したら、コアリポジトリにリソースを一元化して、全員がこれらのリソースを共同で取り扱えるようにします。セキュリティとベストプラクティスに焦点を当てた標準、適切なレビュー、バージョン管理を設定すると、チーム間で信頼できる手法が確立されます。ここまでくれば、チーム構造に基づく効果的な学習スタイルを使用して、これらのツールとワークフローに基づいて、機能のギャップへの対処に着手できるようになります。今では、さまざまなツールを学習するための多数の無料リソースがあります。

全員の意識を変革して自動化のマインドセットと文化を進展させる

全体的なアプローチを作成し、自動化がどのようにチームに働きかけてメリットをもたらすかの理解を浸透させるには、自動化の支援者と擁護者の確保が大切です。自動化アーキテクトは、自動化における信頼、熱意、関心を生み出すクリエイティブな方法を見つけなければなりません。このアプローチには、ランチミーティング、成功を示すダッシュボード、デモ、楽しく参加できるコーディングコンテスト (code-a-thon) などがあります。

適切なガバナンスと管理で自動化ソリューションを拡張する

優れたフレームワークでは、セキュリティとベストプラクティスを損なうことなく、チーム内で自動化をスケーリングして拡大することができます。自動化アーキテクトは、組織のポリシーに準拠するガバナンスを確立して、統制と堅固な監査によってソリューション全体の責任をメンバーに持たせ、適切に管理された特定済みのリスクの範囲内で成功とイノベーションを実現するチャンスを全員に与える必要があります。



確立された自動化アーキテクト

複雑なテクノロジーの発展によって急速な変化が起きる時代にあって、自動化アーキテクトなどの新しいロールの登場は状況を活気づけるものです。システム管理者、各分野のエキスパート (SME)、SRE (サイト信頼性エンジニアリング)、エンジニアにとって、各自の時間のほとんどを運用と自動化へのプロセスに注げる大きなチャンスです。自動化の力によって、各自の狭い専門分野の壁を超えて、自分たちの視野を広げて見識を高める原動力となります。組織面での既存の課題と弱点を理解して分析する、戦略的な立場になります。その後、ビジネス目標に合わせたソリューションを提供してこれらの問題を解決する方法を示すことができ、他のメンバーと協力しながらエンドツーエンドの自動化を成功させるために最適なアプローチとツールを使用します。

このロールでさらに前進するには、継続的に学習して自分自身を成長させ、自動化の問題の理解を深め、適切なソリューションで解決に当たれるようにする必要があります。詳細については、[自動化アーキテクトのハンドブック](#)をご覧ください。



Joseph Tejal

Tejal は Red Hat のテクニカルアカウントマネージャーです。彼は根っからの UNIX システム管理者で、サービス改善プロジェクトに参加し、Ansible を使用して日常的なタスクを自動化しています。さまざまなクライアントへの自動化プロジェクトの提供に関わり、ニュージーランドでの Ansible および OpenShift の交流会の開催に積極的に協力しています。

IT 自動化を成功させるために 必要な 8 つのスキル

スクリプト作成、コラボレーション、ソースコード管理などで 自動化スキルを向上させる

著者: Red Hat シニアアプリケーションデプロイメントソリューションアーキテクト
Chad Ferman (チャド・ファーマン)

振り返ってみれば、私は幸運でした。2000 年以前、私が働き始めた IT ショップにはすでに自動化チームがありました。このチームには魔法のような力があり、問題が発生する前に先回りして修正していました。そうではなくても、発生した時点で問題に対処し、夜中に誰かをたたき起こすようなことはありませんでした。そのとき私は 19 才で、これが通常ではないことも、自動化とは何であるかということすら、理解していませんでした。そしてある日、唐突に理解したのです。このやり方なら、積み残しの仕事のすべてを処理する時間が作れるし、そのうえ、この仕事をもう手作業でしなくてもよいのだと。コマンドラインで実行できることなら何でもコードとして簡単に保存でき、人間が介入しなくても体系的に再度実行できるとわかり、そこから針路を変更し、今に至ります。

自動化の力

私の直前のロールは企業全体の自動化戦略担当のアーキテクトでしたが、そのときは「自動化はスクリプト作成だけではない」をモットーとしていました。自動化には他にもまだ多くの要素があるので、先にそちらに少し触れておきましょう。スクリプトは確かに自動化の基盤ですが、それだけで成り立っているわけではありません。何かを繰り返し実行できるようにするには、システムの状態を知らせる情報を受け取り、システムの期待通りに動作しているかを知る必要があります。ここで登場するのが可観測性と監視です。最終目標を達成するためにプログラム上で何を実行する必要があるか、情報を活用して意思決定を下せます。情報を受け取って自動化を行うというフィードバックループが構成されたら、サービス提供を継続的に改善するサイクルが出来上がります。

キャリアを成功させるためにこれらのスキルを追加すべき理由

自動化は、アプリケーション・デプロイからインフラストラクチャ・デプロイ、ビジネスプロセスなどの一切を強化します。付加価値を得る可能性は無限です。DevOps エンジニア、サイト信頼性エンジニア、アジャイルコーチ、プロダクトオーナー、統合エンジニア、人工知能 (AI)/機械学習 (ML) 運用、ビジネスプロセスエンジニアなど、考えられる職種は数えきれません。システムが相互にやり取りしてビジネス価値をもたらす仕組みを理解することは、多くの業界で強く求められているスキルです。自律的に動作する繰り返し可能なプロセスを作成したいと考えているなら、これこそがあなたにふさわしいジョブです。



少ない人数でより多くの仕事をすることは、特に不況の中では多くの組織が取り組んでいることです。手作業をなくせば、改善のための時間を確保できます。このアプローチにより、信頼性と、予期しないダウンタイムのタイムリーな解決が実現するので、経営陣と信頼関係を築けます。自動化を成功させるパターンを示すと、誰もが共通の目標を理解して、それを中心に連携できるようになります。

たとえば、成功する自動化のパターンを実証したことで、私はより大きな責任と機会を与えられました。以前の会社では、私は3年間で既存のプロセスを改善したことで DevOps エンジニアからエンタープライズアーキテクトに昇格しました。サーバーのデプロイに数カ月かけていたところを、フルスタックのアプリケーションサーバーを30分で提供できるようになりました。これはビジネスアプリケーションを構成、実行し、ビジネス価値を提供する、エンドツーエンドのデプロイメントでした。

自動化はその楽しさから、非常に大きなやりがいを得られることがあります。40ステップの CI/CD パイプラインを使って、セキュリティと API を実行して検証し、コード分析と lint を実行し、ユーザー・インタフェース要素が正しい場所にあることを確認し、回帰テストを実行してダッシュボードでパイプラインのステータスが緑 (すなわち成功) で表示されるのを見るのは、わくわくするものです。

自分がしていることに価値があり、同僚の誰もが繰り返すことができると分かるのは、心地よいものです。また、自分の変更が他の人の変更には影響せず、他人の変更によって自分の作業に支障が生じることもない分かっていることも良いことです。テレビを見ているほうが、自分で演技するよりも楽ではないでしょうか。作業が目的どおりに処理される様子をくつろいで見ていればよく、もしテストに失敗した場合は、何が起きたかが通知されるので、後で修復できます。

自動化に必要なスキル

自動化の MVS (minimal viable skills、最低限必要なスキル) について耳にしたことはありませんか？これには、スクリプト作成、コラボレーション、ソースコード管理、Kubernetes、セキュリティ、テスト、可観測性、監視、ネットワークアウェアネスなどのスキルがあります。

スクリプト作成

プラットフォームの組み込み言語 (Windows では PowerShell、Linux では bash) でスクリプトを作成する能力に長けているなら、スタートするには有利な立場にあります。しかし、より複雑な自動化に取り組むようになれば、Python などの汎用言語の知識があるほうが望ましいでしょう。Python を利用するのは、ここ10年でネットワーキング、サーバー、ストレージ、AI/ML のデファクトスタンダード以上の存在になったためです。自動化のフレームワーク全体は、Python で記述されています。

コラボレーション

スクリプト作成はともかく、コラボレーションはなぜ必要なのでしょう。真のエンドツーエンドの自動化を実現するには、複数のチームが協力して作業する必要があります。ネットワーク、ストレージ、ファイアウォール、プロキシなどが実際にはどのように機能するのか理



解している人は非常に少数なので、これらのすべてを結び付ける共通部分が必要です。このため、自動化のコードをどこにどのように保存し、どのようにデプロイするかについて、企業としての戦略を用意することが重要なことです。コードだけでなくアーキテクチャ設計や API を共有する共通の場所がなくてアーキテクチャの異なる部分を伝達できなければ、インフラストラクチャ、アプリケーション、サービスのお客様への提供を適切に自動化することは不可能です。

ソースコード管理

GitHub、GitLab、Azure DevOps、Bitbucket などの **Git** 管理ツールにすべてのコードを一元化すると、他のチームやグループ内の他のメンバーとのコラボレーションが容易になります。バグを見つけたときに課題を気軽に作成して、適切に文書化する (ただ「壊れている」だけではなく) ことは、コードのコミットの準備ができていない場合に開始する良い方法です。課題を確認できたら、プルリクエストを作成してコードに修正をコミットするか、他の人が発行したプルリクエストをレビューしてその機能のテストをサポートします。完璧な人などいないので、コードのレビューは本当に歓迎され、見逃されていたものが見つかる場合があります。

アプリケーション・プログラミング・インタフェース (API)

API と Playbook の一元化されたカタログを作成して、全員が作業できるようにすることは、自動化を成功させるために不可欠です。このスキルには、API をただアプリケーション内に配置するよりも意義があります。サービスとのインタフェース方法をエンジニアに聞かなくても誰もが使用できるようにすることが、サービス提供の最適な方法です。このようにすると、開発者に従来のインフラストラクチャ・サービスが必要な場合、クラウドリソースとまったく同じように、チケットを発行したり電話をかけたりしなくてもリクエストできます。

コンテナと Kubernetes

コンテナと Kubernetes は、先進的なアプリケーションをハイブリッドクラウドにデプロイする、デファクトの方法となりました。コンテナを構築して、デプロイ、スケーリング、監視、再デプロイする方法の確実な理解は、企業から高い人気を寄せられているスキルです。このスキルは、機械学習やアプリケーション開発から、ビジネスインテリジェンス、サイバーセキュリティまで、企業のさまざまな部分に適用されます。コンテナにより、ローカルマシン上で稼働するコンポーネントが、任意のプラットフォーム上でまったく同じようにアプリケーションレベルで動作することを保証できます。次のステップに進んでコンテナを Kubernetes でデプロイすると、コードを通じて、あらゆる環境に対するシームレスなデプロイに必要なすべてのことが揃っていることを確認できます。

ネットワークアウェアネス

私がネットワークアウェアネスの話をするときにはいつも、クラウドでは光の速度という物理的法則を解決できないという点から始めます。データとユーザーがどこに位置し、データがどこで計算されるかを把握しておくことが重要です。まず、エンドユーザーの、アクセスするデータとの相対的な場所を把握します。ユーザーとおなじ場所で処理が行われるようにします。これができていないのがよくあるミスで、あちこちで見かけます。これがあると、アプリ



ケーションのパフォーマンスが低下してしまいます。そしてもちろん、責められるのはまずい配置ではなく、アプリケーションです。レイテンシーの考慮もまた当然と考えられるもので、帯域幅に恵まれた国の場合は特にそうです。レイテンシーのラウンドトリップテストを実施して、エンドユーザーのエクスペリエンスを確認してください。たとえば、誰かがテキサスで開発したアプリケーションに対して、エンドユーザーの全員がシンガポールにいたことがあります。慎重に検討した後、アプリケーションのプラットフォームはシンガポールに移され、処理に数分かかっていたアプリケーションのユーザーエクスペリエンスは数秒に短縮されました。

テスト

テストもまた、持っているといいのに忘れられているスキルで、キーを1つ打ち間違えただけでプロダクション環境が損なわれることを防ぎます。動作させようとしている内容が期待通りに実行されるかの検証は、成功するためには非常に重要で、問題を解決するため一晩中作業するはめになるような意図しない結果を招かないようにします。これはバックエンドテストに限りません。ユーザーインターフェース (UI) 要素と API のテストと検証のための優れたツールがあり、変更によって既存の機能やエンドユーザー・エクスペリエンスが影響を受けないようにできます。

セキュリティ

セキュリティをアプリケーションを組み込むことは、ランサムウェアや暗号通貨をマイニングしようとクラウドデプロイを乗っ取る不正ユーザーがはびこる現代では、不可欠です。セキュリティ統合は、アプリケーションをデプロイする CI/CD パイプラインの一部であるべきです。このパイプラインでは、静的コード分析、アーティファクト管理とロック、安全なライブラリ、デプロイされたときに同じコードまたはアーティファクトであると保証するためのコード署名などが必要です。CI/CD はアプリケーションのインスタンス化のみを担当します。デプロイ先のプラットフォームのセキュリティ強化も必要です。さらに、既知の脆弱性があるライブラリを実行していないと検証し、ライブラリにパッチが必要な場合はチームに通知できるものが必要です。

サイバー犯罪の増加により、
セキュリティテストは
多数の企業が採用するロールの
1つになりました。

サイバー犯罪の増加により、セキュリティテストは多数の企業が採用するロールの1つになりました。セキュリティテストをアプリケーション・サプライチェーンに組み込むことは、ビルド、デプロイ、コードが目的とされていることのみを実行するかのランタイム検証までのライフサイクル全体で、多くの組織が実施するようになりました。ライブラリとコンテナなどのアーティファクトのソースの検証手段として、署名されたライブラリと実行可能ファイルは標準となりました。信頼された署名済みライブラリとコンテナを使用し、カスタムライブラリとアーティファクトリポジトリを組織に提供することが、セキュリティ意識の高い組織にとって標準となりました。



可観測性と監視

アプリケーションの状態と、どのようにその状態になったのかを理解することも、タスクを適切に自動化するために必要なスキルです。サービスで起きていることを把握していなければ、プロアクティブな自動化を作成して問題を修復したり、一貫した状態を適用して今後の問題の発生を防ぐことはできません。多くの人は監視止まりで、問題が発生すると、根本原因分析 (RCA) を使用して何が起きたかを突き止めます。可観測性は RCA に必要なツールを提供し、これを常に備えておくと、通常は監視によって確認できるステータスの良・不良以外に、何が起きているかを把握できます。この点についての優れた資料として、[DevOps 監視ガイド](#)をご確認ください。

まとめ

自動化は、先進的な IT 分野において、最もやりがいのある仕事だと思います。大局を見据えて、全体的な物事の仕組みについて理解する必要があります。「これが仕組みだ」と言われただけでは満足せず、自分で試してみないと気が済まない人にとっては、これが最適な仕事です。人間が手作業であることを反復プロセスに転換して、より価値が高い作業に専念できるようにすると、企業のコストを節約し (最終的にはこれに行き着くことを誰もが知っています)、より画期的なプロジェクトに集中できるようになります。ここで取り上げたすべてのスキルは相互に関連し、自動化の優れたエキスパートになるために役立ちます。これらのスキルは一度に獲得できるものではないので、時間をかけて楽しみながら取り組み、手作業をなくしましょう。



Chad Ferman

Ferman は、Red Hat のシニアアプリケーションデプロイメントソリューションアーキテクトです。彼は 25 年以上、エンタープライズ IT で小売りおよび石油・ガスの公共部門および民間部門で働いてきました。彼の役割は、インフラストラクチャ運用からマイクロサービス・アプリケーション開発、エンタープライズ戦略に至る、すべてを範囲にしていました。最近 Red Hat に入社し、お客様のエンタープライズ向けソフトウェア開発および先進的手法への文化の変革を成功させるべく支援しています。

Linux システム管理者の 賃金プロファイルと可能性を向上させる 6つの方法

キャリアや給料を向上させる必要がある方のために、
適切な方向に進むための優れたヒントをいくつか紹介します。

著者: Red Hat テクニカルアカウントマネージャー Joseph Tejal

はじめに

今年もこの季節がやってきました。マネージャーと自分のパフォーマンスについて話し合う時期です。問題は、このチャンスを活かして、給料アップや昇進の候補の座を得るための準備ができていのかどうかということです。

最良のシナリオは、あまり多くを語らなくて済むことです。業績や組織に付加した価値が物を言い、マネージャーは自分の業務を楽しんでくれたことを感謝し、昇進を認めてくれます。これは、システム管理者としての自分が得る収入の可能性を高める1つの方法で、在籍している組織内で成長するというものです。

今の職場以外でチャンスを得る場合、自分の実績を転職先の雇用主に納得させて、面談時に自分の売り込みを成功させる自信はありますか？最良のシナリオは、自分の取り組みと成功を難なく披露し、相手が感心をもって詳しく聞きたがり、最終的には有利な条件で就職できるというものです。さらには、あなたのプロフィールとブランドがコミュニティで高い評価を得ていれば、複数の企業が競ってあなたを獲得しようとするでしょう。

以上、システム管理者として収入の可能性を増やす方法をいくつか挙げてみたわけですが、大きな問題は、こうした面談やチャンスをどのように手にすればよいのか、またそれらに対してどのように準備するかということです。

ここからは、私の経験や私が他の人々から聞いた経験に基づいて、ヒント、アドバイス、アイデアを紹介します。

代わりの利く存在になる

これは直感的な方向性とは真逆ですが、私が学んだのは、前進するためには、それまでの仕事は後ろに置いて価値を付加することに集中しなければならないということです。

チームと組織に付加を価値するとは、日常業務を**改善**、**自動化**、文書化して、誰でもできるようにすることです。誰もしなくて済むようになれば、なお良いといえます。主な問題点の解決を目標として、誰もがあなたの業務を確実にこなせるようにします。こうすると、全員から信頼感を得るための時間ができ、より価値のある業務に取りかかれます。



現状に甘んじてはいけません。自分自分や職場の「あたりまえ」を疑い、非効率であったり古くなったりしている場合は特に挑戦してみます。関係者は、あなたが生み出す価値や改善に強い印象を受けるでしょう。

自分のキャリアと目標をコントロールする

自分のキャリアと目標を管理すると、自分自身の限度、さらには組織の限度を克服できるようになります。支援は得ようと思っても得られないけれど、自主性があれば、利用できるリソースで学習して前進できると、聞いたことがあるでしょう。そうした技術的なトレーニングを、要求する前からでも体験できるオープンソース・プロジェクトが多数あります。

無料トライアルやオンラインで利用できるワークショップを使用して、**Red Hat Ansible Automation Platform**、ハイブリッドクラウド、Kubernetes、**Red Hat OpenShift** などの新しいテクノロジーについて調査し、概念実証 (PoC) を実施しましょう。これらのリソースは、会社による今後の学習や開発投資の候補となるために役立ち、正式なトレーニングや**認定試験**を受ける資格を得られます。

他の人に説明責任があったほうが集中しやすくなるので、一緒に学習する仲間、技術上のコーチやメンターを見つけるとよいでしょう。技術面での成長だけでなく、リーダーやマネージャーとして他のシステム管理者を管理できるよう、統率力の成長も目指しましょう。

技術面での成長だけでなく、
リーダーやマネージャーとして
他のシステム管理者を
管理できるよう、統率力の成長も
目指しましょう。

ソフトスキルに取り組む

今の時代、技術的なスキルを持っていることだけが収入を増やす方法ではありません。たとえば、感情知性、態度、他人と協力する能力、コミュニケーションスキルは、自分を際立たせる独自のメリットを与えてくれます。

組織内で尊敬できる同僚やコーチを探すことから始めましょう。トーストマスターズなどのスピーチクラブに参加して、会議室や講演会での対人関係やコミュニケーションのスキルを強化するのもよいでしょう。ライティングもまた注目に値する手段です。**Enable Sysadmin**、**opensource.com**、**medium.com** など、貢献したり他者から学習したりできる場所が多数あります。

圧倒されたり、心地の良い領域から出るのを恐れたりしないでください。多くの人々が、キーボード、端末、技術知識を使わなくても発揮できる、自分でも気づかなかった強みを発見しています。



他人とコラボレーションする

一部のシステム管理者がトップシークレットを握って有利な立場を確保し、他の全員は重要な知識を持たないでいるという時代は、過去のものになりました。

今は、イノベーションはコラボレーションから湧き出てくる時代です。他のチームを協力して自分の知識を広げ、同時に他者の知識に貢献しましょう。関心のあるチームで交代で担当する取り組みを行うこともできます。小規模なプロジェクトを立ち上げて、チーム内で一部のプロセスを改善するか、現在あるものより優れていると思われる、アップデートされたツールやテクノロジーを調査してみてもよいでしょう。このコラボレーションは、部門の壁を排除し、文化の変革を起こし、興味をかきたてます。これにより、あらゆる人が成功できるようになります。

全員の作業と進行状況を文書化して公開し、新しく誘われた人々が容易に参加できるようにします。

他者につながって自分の評判を上げる

キャリア上の人脈と照会先があると有利になります。一緒に働いてきた主な人々からの前向きなフィードバックがあると、どのような業績を上げて期待を上回ってきたかがマネージャーに伝わります。

コミュニティの一員となることで、そこで働くかもしれないさまざまな人々とのつながりを作ることができます。**イベントに参加したり主催したり**しましょう。会社や地域のコミュニティでのランチミーティングや交流会などです。そうした場に出ていくと、他者の助けになり、自分を知ってもらうことができます。

チームまたは部門内の少人数のグループやランチミーティングから始めて、成功事例や画期的なプロジェクトについて話し合うこともできます。それがうまくいけば、取り組んでいる素晴らしい内容について技術会議に論文を発表したり提案書を提出したりすることができ、これによってあなたの評判がさらに上がります。

仕事を楽しむ

元気をチャージしてくれるものを見つけましょう。仕事によって時には気力が削がれることがあります。困難な状況での対応があなたを形作ります。自分のキャリアやその大部分の局面を楽しんでいるなら、自分が役立ち、価値を付加できる状況に置かれていることで生き生きと活動できます。

楽しみながら自分のスキルと強みを発揮できるチャンスを見つけて、弱みを強化する機会もいくつか取り入れます。視野を広げて新たに身に付けた熱意をもってあたれば、仕事は退屈にはなりません。

他者と交流し、自分の分野で創造性を発揮し、いつものやり方ではない方法でチャンスを手に入れることで、活気を得る方法はたくさんあります。仕事をしない充電日を設けて、エネルギーを取り戻しましょう。



コラボレーションが導く明るい未来

収入を得て自分を維持する能力は、なぜ働くかという動機の1つです。キャリアの成長、個人の達成感、楽しさとともに、得られる収入が増加すると、将来に向けた勢いがつきます。他者に伝えられるサクセスストーリーを用意しておきましょう。ただし、このプロセスは一朝一夕でできるものではないことに注意してください。目標を念頭に置いて日々取り組んだハードワーク、投資、専念、熱意のすべての集大成です。

自分のブランドとストーリーを構築するためにあらゆるチャンスを活用しましょう。チャンスが訪れたら、業績そのものが影響を与え、成功へ導く手助けとなります。

「6 ways to increase your Linux sysadmin earning profile and potential」から引用、
公開先 Enable Sysadmin、Creative Commons 4.0 BY-SA license の下で発行、
<https://www.redhat.com/sysadmin/increasing-earning-potential>



Joseph Tejal

Tejal は Red Hat のテクニカルアカウントマネージャーです。彼は根っからの UNIX システム管理者で、サービス改善プロジェクトに参加し、Ansible を使用して日常的なタスクを自動化しています。さまざまなクライアントへの自動化プロジェクトの提供に関わり、ニュージーランドでの Ansible および OpenShift の交流会の開催に積極的に協力しています。

チームを自動化ファーストの考え方に転換する 5 つの方法

DevSecOps から組織は競争力を得られます。この 5 つの戦略を使用して始めましょう。

著者: Red Hat コンサルティング シニアアーキテクト Allen Eastwood、Red Hat サービス・ポートフォリオ管理チーム プリンシパルソリューションマネージャー Larry Spangler (ラリー・スパングラ)

自動化ファーストの考え方により、あらゆる組織が大幅に変革される可能性が高く、タスク自動化から始めて、複雑なワークフロー・オーケストレーションへと移行し、最終的にはインテリジェントな運用と「プッシュボタン」式のエンドユーザーサービスへと革新していきます。DevSecOps への確固たるコミットメントを体現し、このタイプの変革がもたらす競争力が認められます。ですが実際には、こうした状態に達して必要なサポートを見つけて構築することは難題です。あらかじめ自動化を個々の部門で成功させていた場合ですら、そうです。

早期に成功を積み重ねる

最初から、ドメイン内にある最も重要で時間を要する手動プロセスから自動化を始めたいと思いがちですが、過程の早い段階で小さな成功を積み重ねることを目指すほうが大きな利点を得られます。範囲が広いプロセスの個別の部分から自動化を始めると、後でより複雑な自動化を行うための構成要素が得られ、自動化の価値を早い段階で提示できます。

反復しながら進行させ、対象とするそれぞれの反復とその特徴を考慮しなければなりません。開発者の仕事をしている実感が得られるはずですが。範囲と機能を反復的に増加させると、徐々に大きな目標を立てられるようになり、構築してテストしたばかりの自動化を活用することもでき、構築に自信を持てます。

こうした提示できる成功を収めると、自動化の取り組みをチーム、マネージャー、他のチームに広めるときに役立ちます。

スキルアップとサードパーティのサポートを取り入れる

トレーニングと認定は、自動化ファーストの考え方を取り入れるためのあらゆる側面にとって重要です。チームが自動化を確実にデプロイするために重要なだけでなく、組織内で自動化導入を広めるための勢いを作るためにも役立ちます。トレーナーとなれる人材を育成する手法を活用して他のチームを強化しながら、確立された自動化の標準とガバナンスでコンプライアンスを確保します。



多数の組織が外部コンサルティング会社との戦略的提携を使用して、**自動化導入プロセスの迅速化**を図っています。これは恒久的にアウトソースされたり仕事を失ったりする恐れにつながり、組織内の IT プロフェッショナルにとっては落ち着かない状況です。しかし優れたコンサルティングでは、メンタリング関係を通じてこうしたチームの実力を発揮させて強化する、測定可能で有限の目標を設定します。すると、自動化が運用され、より複雑なプロジェクトとユースケースの取り組みを続けられる能力が身に付きます。

味方を見つける

大規模な変更には腰が引けることがあり、エンタープライズ自動化などの取り組みには、この種の変革に伴う自然な抵抗を克服する機会に喜んで対応する人物が必要です。このような人物は組織に比較的最近加わっていることが多く、「何にも縛られるな」という信念のもとに問題に取り組みます。この人物があなたかもしれません。いずれにせよ、アイデアを行動に移し、開発と運用など、一緒に働くのを嫌がるチームであっても、チーム全体への普及を始めるには必要な人材です。

自動化ファーストの考え方を定着させるには、コラボレーションが必要です。

ガバナンスと標準を確立する

自動化ファーストの考え方では、新しいタスクとプロセスを自動化する準備ができていないことを前提としています。準備とはつまり、信頼できる情報源を確立し、進行しながら文書化して他のチームが成果を安全に利用できるようにすることです。コードと変更を追跡し、自動化コードを自分の情報やデータモデルから分離し、特権情報を信頼できるツールでセキュリティ保護し、ファイル共有によるソースコードの分散を防止する方法が望まれます。コードを文書として使用すると、コラボレーションが育成され、さらには組織が変革されて自動化プラクティスが生まれ、組織内で共有できるようになります。

一元管理された自動化コードのほか、標準を定義して進化させ、チームとプロジェクトを越えた自動化の統合およびオーケストレーションへの対応を向上するために必要です。広範な導入と再利用が増加したら、チームを越えた課題に対処するための統括グループの確立を検討します。これには通常、標準ツール、自動化テストの要件、変更のデプロイまたはロールバックのベストプラクティスが含まれます。

コミュニティを作成してコラボレーションする

自動化ファーストの考え方を定着させるには、コラボレーションが必要です。世の中を懐疑的な目で見ると多くの人は、この現実に気が削がれるかもしれません。しかし DevSecOps の根幹は、楽しくあろうとなかろうと、これらのチームを集結させることです。プロセスは魔法ではありません。すべての問題が解決される訳ではありません。結果は複雑なオーケストレーションで、自動化を規定する包括的な標準とプロセスを含みます。これらの自動化は組織内の実際の弱点をターゲットとし、測定可能な実際のビジネス価値を働いているチームにもたらすものであるべきです。

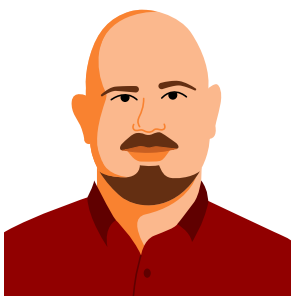


コラボレーションをすばやくより効率的にする方法の1つは、組織内に自動化コミュニティを設立することです。このコミュニティは自動化のコアチームとなり、組織全体からメンバーを募る必要があります。その使命は、アプローチを共有し、課題に対処し、他のチームの自動化導入の成功を補助することです。これらの他のチームの取り組みを支援して、成功と可能性を示し、標準プラクティスを指揮し、問題の解決と自動化利用の促進を支援する知識 (および Playbook) を共有します。

終わりはあるのか

自動化は、ビジネス上のインパクトを狙った場合に最も強力になります。こうしたインパクトとして、内部の生産性 (プッシュボタン式的环境など) や顧客対応 (迅速な顧客対応時間など) が挙げられます。自動化の導入が組織内で広まると、進化を続けて価値を増加させるには、根本的なアプローチを考え直す必要があることがわかってきます。たとえば、途中で承認を受ける必要があったプロセスを再構築して、完全に自動化します。自動化するタスクを厳密に個別検討するよりも、手動タスクを自動化されたタスクで置き換えるだけに留まらず、ワークフローとプロセスを継続的に最適化できるプラットフォームを構築します。イベント駆動型の自動化された修復を実行し、ダウンタイムと顧客サービスの中断を最小化するパッチによってコンプライアンスを自動化し、不要な手作業を排除するためのより安全な方法を生み出すプロセスとワークフローを設計します。

根本的には、自動化は単純なソリューションですが、自由度が非常に高いソリューションでもあります。そして変化は迅速に発生します。自動化は本質的に柔軟でビジネスニーズに適応できるソリューションなので、目標は急速に変化します。その結果として自動化ファーストの考え方が得られますが、回避策としての自動化、反復および実用最小限の製品 (MVP)、作業の最適化は、能力や進化するビジネスニーズに適応していかなくてはなりません。



Allen Eastwood

Eastwood は Red Hat コンサルティングのシニアアーキテクトで、20 年以上の経験を積んでいます。顧客の IT システムの自動化を支援するかたわら、アイスホッケーを楽しんでいます。



Larry Spangler

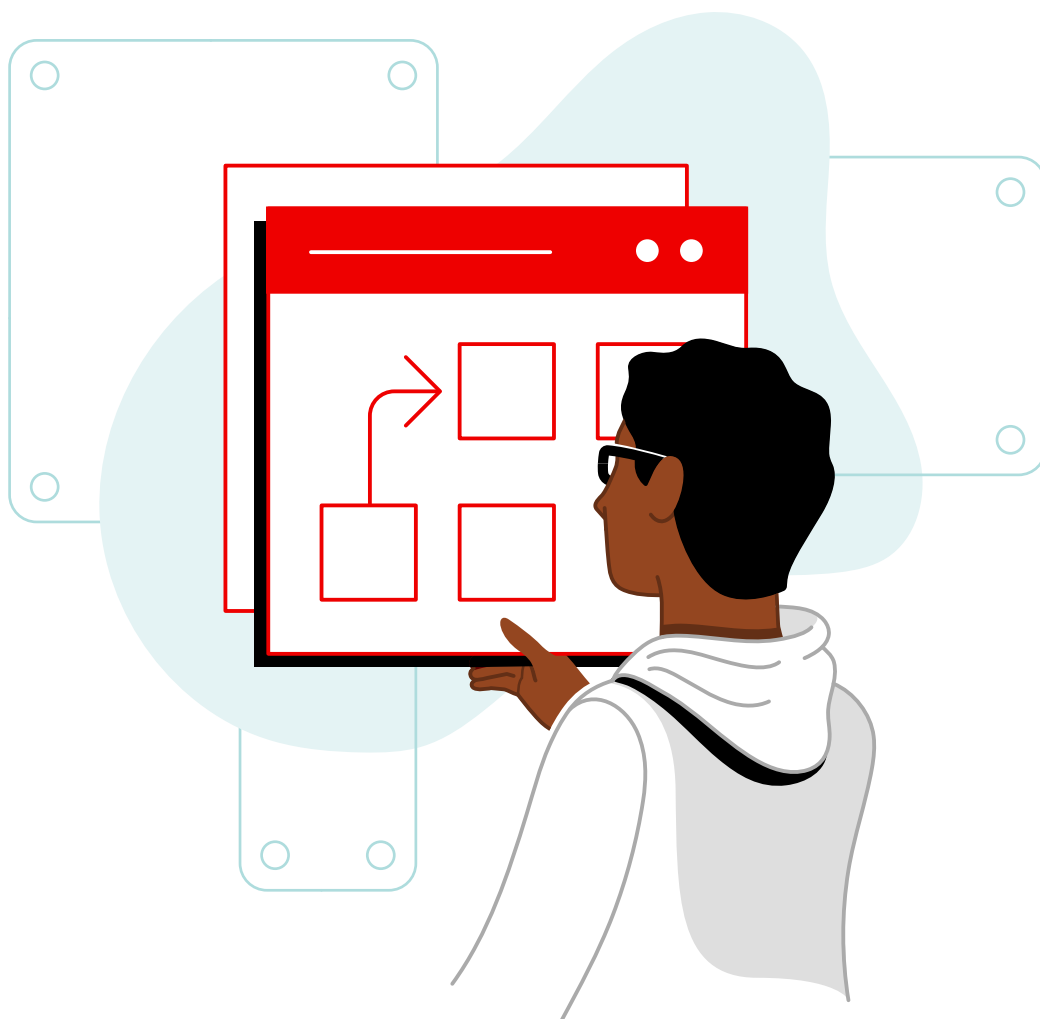
Springer は Red Hat サービス・ポートフォリオ管理チームのプリンシパルソリューションマネージャーで、Red Hat のグローバルサービス組織向けの自動化サービス製品の定義、開発、管理を率いています。IT ソフトウェア開発、システム管理、プロフェッショナル・サービスの分野で 30 年以上の経験を積み、幅広い経験と知見を実用面に活用しています。

まとめ

IT 自動化を開始する方法

45 IT 自動化を受け入れる：キャリアにとってプラスになる理由と、開始する方法

48 始めるには



IT 自動化を受け入れる： キャリアにとってプラスになる理由と、 開始する方法

Red Hat は、ビジネスに即座にインパクトを出せる包括的なトレーニングによって、自動化への移行の効率を向上させることができます。

著者：Red Hat グローバルラーニングサービス自動化プラクティス マネージャー James Mighion (ジェームズ・ミジオン)、Red Hat トレーニング プリンシパルテクニカルアーキテクト Steven Bonneville (スティーヴン・ボネビル)

2019 年、プロジェクトとチーム全体に対する自動化機能を、ガバナンスと制御によって作成、共有、再利用する必要があることを **Comcast** が見出しました。このような業務を管理するにはインハウスの専門知識も必要だったため、Comcast は Red Hat と提携して 5 日間のオンサイトトレーニングを実施しました。Comcast の音声エンジニアは Red Hat Ansible Automation Platform に関する Red Hat トレーニングコースを受講し、エキスパートが指導する実践的なカリキュラムにより、IT スタッフに自動化を実装、管理、トラブルシューティングする機会を与えました。

そのうえ、Red Hat トレーニングと認定のメリットは各個人にも及びました。多数の IT プロフェッショナルが各自で自動化トレーニングを探し、会社への価値や個人の市場価値を強化できることを見つけ出しました。たとえば IT プロフェッショナルの Christian Sandrini 氏は、**2021 年 Red Hat 認定プロフェッショナル・オブザイヤー** に選ばれました。Red Hat 認定スペシャリスト - Ansible Best Practices - や Red Hat 認定スペシャリスト - Ansible Automation Platform - など、十個あまりの Red Hat 認定を獲得しています。

Sandrini 氏は Red Hat ラーニングサブスクリプションの登録者として Red Hat トレーニングに積極的に参加しています。彼は次のように述べています。「Red Hat ラーニングサブスクリプションは私の学習スタイルに合っていたので役に立ちました。自習型の学習プラットフォームがあり、特定の動画を見直したりラボを行ったりでき、実践性の高いものです。テクノロジーを極めてすばやく習得でき、ある製品について知らなくても、クラスにサインアップしてその製品がどのようなものかすぐに学習できました。もう 1 つ気に入っている点は、Red Hat ラーニングサブスクリプションの早期アクセスセクションで、今後発表されるテクノロジーの概要を見られることです」

Sandrini 氏が知識の実現に向けて踏み出した最初のステップは、所属する組織に Red Hat Ansible Automation Platform を導入することでした。彼の目標は、機能を自動化して、Infrastructure-as-Code アプローチによってチームがサーバーをプロビジョニングして構成する方法をモダナイズすることでした。この成果についてチームから圧倒的な好評を

得て、Ansible Automation Platform をデプロイしました。Sandrini 氏は次のように言います。「変化が起きたのはこのときです。ロールベースのアクセス制御を使えるようになり、他のチームが関心を持つようになりました。Ansible は Linux に使えるだけでなく、その周辺の他のコンポーネントについても使用できることが示されました」

Sandrini 氏はチームにとって欠かせない存在となり、IT 自動化の実装とメンテナンスを行い、Red Hat トレーニングと認定を通じて取得した知識を使用してプロセスを効率化し、コストを削減しています。チームがトレーニングを利用してスキルを獲得して向上させ、チームメンバー間に知識を共有できるように支援しています。

**Red Hat は 20 年以上、
現在および先進の動向に後れを
取らないように、テクノロジーの
プロフェッショナルに
トレーニングと認定を
提供してきました。**

IT 自動化に熟練したプロフェッショナルの需要は高まるばかりで、新しい戦略やテクノロジーを把握していることの重要性も増しています。常に進化している IT 分野では、トレーニングと専門性の開発は個人レベルでも組織レベルでも競争に生き残るための鍵となります。Red Hat トレーニングと認定はコース内容を絶えず更新して業界に足並みを揃えているため、IT 自動化においてキャリアを発展させたいと思う熟練した認定済みのプロフェッショナルは、成功するために必要なツールを手に入れられます。

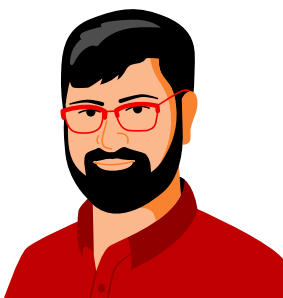
Red Hat は 20 年以上、現在および先進の動向に後れを取らないように、テクノロジーのプロフェッショナルに各種のトレーニングと認定を提供してきました。Red Hat トレーニングコースにはさまざまなニーズや学習スタイルに適した選択肢があり、従来のクラスルームベースのトレーニングから、オンサイトトレーニング、多様なバーチャルラーニングのクラスを用意しています。平均して IT に 20 年、Red Hat 製品に 11 年の経験を有する Red Hat 認定インストラクターが、専門知識を活用してビジネス目標に即座にインパクトを与えられるよう努めます。

Red Hat ラーニングサブスクリプションを利用すると、学習者はリアルタイムでバーチャルトレーニングを自分のデバイスから完了したり、自習型コース、動画、ラボのすべてを含むカタログに即座にアクセスしたりできます。このコンテンツのすべてが一貫して更新され、業界と足並みを揃えます。実際、Red Hat ラーニングサブスクリプションのカタログにあるコースの 41% が、昨年追加または更新されています。Red Hat ラーニングサブスクリプションの一部のレベルには、Red Hat 認定試験の受験料金が含まれているので、知識を獲得してから実力を証明するまでの手続きが単純化されます。

Red Hat トレーニングは自動化およびデジタル・トランスフォーメーションの最新情報の把握に役立ち、Red Hat 認定は IT プロフェッショナルにスキルがあり、進化する業界の課題に対して最も野心的なプロジェクトに取り組む準備ができていと検証します。Red Hat は現在、IT 自動化を中心とした認定を 2 つ提供しています。Red Hat Enterprise Linux Automation with Ansible (RH294) と RHCE 認定試験は、ワークフローの自動化、DevOps 手法の採用、開発効率を向上させるための Ansible Automation Platform の使用に対する基礎を築きます。さらに前進するには、Advanced Automation: Red Hat Ansible



Best Practices (DO447) と Red Hat 認定スペシャリスト試験 - Advanced Automation: Ansible Best Practices - で、既存の Ansible インフラストラクチャを大規模なエンタープライズ環境の事業部門に使用して拡張するために必要なスキルを開発し、実証します。Red Hat 認定プロフェッショナルは、自動化における専門知識により、組織が最適な効率性とコスト削減を達成するよう支援します。さらに、オンボードの時間短縮やコスト削減の増加から、組織は認定プロフェッショナルを採用して維持することに関心を持っています。



Steven Bonneville

Steven Bonneville は Red Hat のトレーニング部門におけるプリンシパルテクニカルアーキテクトです。Red Hat に 20 年以上在籍している Bonneville は Red Hat テクノロジーの深い専門知識を提供し、Red Hat のトレーニングカリキュラムのコースを設計し、カリキュラムの開発者の指導と指揮を行い、今後のトレーニング開発の方向性を助言します。長年にわたって、RHCE トレーニングトラック、システム管理、仮想化、ストレージの上級コースなど、Red Hat Enterprise Linux システム管理カリキュラムを担当しています。これらのコースの多数について最初のバージョンを作成しています。現在注力しているのは、Red Hat Ansible Automation Platform を使用してクロスプラットフォームの自動化トレーニングを開発することです。



James Mighion

James Mighion は Red Hat のグローバルラーニングサービス組織における自動化プラクティスのマネージャーです。2011 年から Red Hat に在籍し、それ以来さまざまな役職を歴任しています。自動化に熱意を持ち、オープンソースプロジェクトに貢献しています。

今すぐ始める



Red Hat Ansible Automation Platform の試用版を始める

Red Hat Enterprise Linux® 向け Red Hat Ansible Automation Platform の単一のセルフサポート式 60 日間サブスクリプションを取得します。

→ [Ansible Automation Platform を 60 日間試用する](#)



Red Hat Ansible Automation Platform の基礎を確認する

一連のオンデマンドオンライン動画を試しましょう。Ansible Essentials: Simplicity in Automation Technical Overview (DO007) では、Ansible Automation Platform の構成管理、プロビジョニング、デプロイ、およびクラウド、仮想、物理環境にわたるコンピュート・インフラストラクチャの管理について説明します。

→ [Ansible Automation Platform を無料のトレーニングで学習する](#)



Enable Sysadmin: システム管理者向けのコミュニティ

Enable Sysadmin はコミュニティによる投稿で、ガイド、ハウツー、チュートリアル、説明、ヒントとコツなどを公開しています。投稿内容には既存および先進テクノロジーの概念、ソフトウェアとアプリケーション、システム管理者がよく使用するスクリプトおよびプログラミング言語、一般的な課題と解決策などがあります。

→ [ニュースレターを購読する](#) | [コミュニティに参加する](#)



自動化を活用するリーダーシップでチームをつなぐ

デジタルリーダーは単に既存のワークフローを自動化しているだけではありません。自動化により共有価値を生み出す方法も学んでいます。すでに IT の自動化に着手している組織もあるでしょう。その場合、その取り組みは高度な段階にあるでしょうか。それともまだ初期段階でしょうか。現在の自動化の成熟度を判定できるよう、Red Hat ではオンライン診断を用意しました。次のステップの特定や、進捗をサポートするリソースの入手にお役立てください。

→ [診断する](#)