



Puppetによる自動化で 現場の生産性と 開発スピードを向上。

CASE STUDY



NTT DATA

株式会社NTTデータ

設立 1988年5月23日
本社所在地 〒135-6033 東京都江東区豊洲 3-3-3
豊洲センタービル
資本金 1,425億2千万円
事業概要 システムインテグレーション事業、
ネットワークシステムサービス事業 他

URL <http://www.nttdata.com/jp/ja>

業種

情報・通信業

背景

人によるインフラ基盤の構築にかかってしまう時間やコストを削減したい。

ポイント

Puppet Enterprise(以下、Enterprise)の導入時に専門のチームを組み、開発担当のメンバーに対してPuppet Language(以下、Language)のレクチャーを実施。精度の高いプライベートクラウドをスムーズに導入できた。

ソリューション

Enterpriseを導入することで、サーバー構築の自動化を推進する。導入後も、運用と検証のサイクルを回し続けることで自動化の範囲を広げ、トータルで高い費用対効果を維持。

効果

申請・承認プロセスが不要になり、以前に比べて1ヵ月以上の工期短縮に繋がった。また、サーバーを構築する際に発生しがちな人為的なミスもほぼゼロになり、効率化だけでなく品質向上にも期待。

時間がかかるプロセスを省くことで、 工期を劇的に短縮。

私の主な業務は、インフラの自動化推進および社内へのプライベートクラウドの導入・普及です。人によるインフラ基盤の構築には膨大な時間とコストがかかります。また、社内各プロジェクトに必要なサーバーを用意するには、毎回申請や検証、承認を経なければなりません。このプロセスに1〜2ヵ月かかることもあります。スピード感が要求される昨今の現場においては致命的な遅れに繋がりがねず、人間が介在することでミスが生じやすくなることも問題です。これらの課題を解消するために、自動化ツールの導入を検討しました。

Enterpriseは、サーバー構築や管理の際に生じる設定作業を自動化できるツールです。インフラ基盤の自動化による恩恵は、時間やコストの削減や作業の迅速化に加えて、手間がかかる申請・承認プロセスをすべて省けること。実際、プロジェクトの初期段階での工期を1ヵ月以上も削減できました。他にも、サーバー構築作業が長時間に渡った際に発生しがちな人為的ミスが生じなくなり、品質向上にも繋がります。「マイナスがゼロになる」というのはベネフィットとしては地味かもしれませんが、プロジェクトの効率化と品質向上の観点からは重要なポイントです。自動化には導入コストこそかかるものの、使用、検証、修正のサイクルをしっかりと回し続けていけば、トータルでは高い費用対効果が期待できます。

導入へのハードルが低いことが、 最大の決め手に。

いくつか存在する自動化サービスのなかでEnterpriseを選んだのは、マニフェストと呼ばれる専用のプログラムの可読性が高かったためです。マニフェストの記述に用いるLanguageは、競合他社製品で使われている言語と比べても非常にシンプルに構成されています。プログラミングに精通していないメンバーであっても記述内容から完成形をすぐにイメージできるほどです。技術者ではないメンバーも開発に関わることの多い当社にとって、利用に向けたハードルの低さは大きなポイントになりました。同様に、有料サービスのPuppet Forge (以下、Forge) も、スムーズな導入を支えてくれます。これはPuppetに関連するアプリケーションやサービスを紐付けるためにあらかじめ用意されたモジュールで、導入の際に生じる開発作業を省略できます。モジュールは開発自体よりも、その動作検証に時間がかかります。けれども、Puppet側が開発し、動作を保証してくれるので、導入時の負担を大幅に減らしました。

導入時に一番の課題となったのは、Puppetを扱う技術の習得です。いくらForgeでモジュールを用意できるとはいつても、プライベートクラウドで利用するための開発は不可欠です。マニ



株式会社NTTデータ
技術革新統括本部 システム技術本部
方式技術部 企画推進担当
課長代理

菅原 亮 氏

フェストを書ける技術者が、何人かは必要となります。それにLanguageがシンプルな言語だとしても、いきなりマニフェストを書けるはずはありません。導入に際してはまず開発メンバーの教育が必要となりました。コストも時間もかかりますが、のちのち自分たちで修正を加えながら運用していくことを思えば、ここでの教育には十分なリターンが見込めます。

さまざまな業務の自動化を、 もっとPuppetで推進したい。

「インフラ領域を専門とし、シェルスクリプトやオブジェクト指向言語をある程度書ける」という条件で集めたのが、今回の開発メンバーです。全員に技術的なベースがあったこともあり、2日ほどのレクチャーで、1週間後にはある程度はマニフェストを書けるまでに上達しました。通常の言語と比較しても、習得が非常に容易だと実感した瞬間です。そこからテスト開発での実習をはさみ、本番の開発に乗り出しました。トータルで1ヵ月ほどの準備期間がかかりましたが、Languageの習得者を集めるよりも、ずっと効率的だったと感じています。

現在は、Puppet Discovery (以下、Discovery) のような関連サービスの導入も検討しています。トラブル発生時にサーバーの設定を変更することがしばしばありますが、その際に問題となるのが設定の戻し忘れによって起こるセキュリティトラブルです。Discoveryは、それを回避するために人力で行ってきた膨大な確認作業を、完全に自動化してくれます。これ以外にも、自動化すべき業務はまだたくさんあるでしょう。Puppetさんには、今後もあらゆる業務の自動化を推し進める製品を、どんどん開発していただきたいですね。