

---

---

# サーバ構築自動化における 課題解決への取り組みについて

## 富士通エフ・アイ・ピー株式会社

---

### ■ 執筆者 Profile ■



岩里 洸介

2016 年 富士通エフ・アイ・ピー株式会社 入社  
2017 年 現在 IT プロフェッショナル技術支援部所属

### ■ 論文要旨 ■

サーバ構築自動化は弊社において、新たな取り組みの一つである。筆者は構成管理ツール Ansible を用いた、OS 環境設定およびミドルウェア環境設定の自動化を担当している。

作業を進める中で、Ansible に関するノウハウ不足による課題に直面した。条件によって Ansible の処理内容が変わり、想定と異なる動作をするといった課題である。筆者は、上記の課題解決に取り組み、サーバ構築自動化を実現することができた。さらに、得られたノウハウを元にサーバ構築自動化に関する他の成果物に対する改善提案を行い、ノウハウを部署全体に展開した。

本稿では、筆者が経験したサーバ構築自動化における課題解決への取り組みについて述べる。また、改善提案によりノウハウを展開した事例について述べる。

## ■ 論文目次 ■

<b>1. はじめに</b> .....	《 3 》
1. 1  当社の概要	
1. 2  インフラ構築強化チーム	
<b>2. 参画プロジェクト</b> .....	《 4 》
<b>3. Ansible を用いた構築自動化</b> .....	《 4 》
3. 1  構築自動化とは	
3. 2  Ansibleについて	
<b>4. 構築自動化を進める上での課題と取り組み</b> .....	《 5 》
4. 1  作業実施時に発生した課題	
4. 2  課題への取り組み	
<b>5. ノウハウの共有と展開</b> .....	《 7 》
5. 1  サーバ構築自動化に関する他の成果物に対する改善提案	
5. 2  インフラ構築強化チームへのノウハウの横展開	
<b>6. まとめ</b> .....	《 8 》
6. 1  OJT 期間の成果と振り返り	
6. 2  今後の展望	

## ■ 図表一覧 ■

<b>図1</b> 構築自動化ツールを用いたサーバ構築の流れ .....	《 4 》
<b>図2</b> playbookの記述例 .....	《 5 》
<b>図3</b> エラーの循環 .....	《 6 》
<b>図4</b> 処理内容が変化する例 .....	《 7 》
<b>図5</b> 処理内容が変化しない例 .....	《 7 》

## 1. はじめに

### 1. 1 当社の概要

当社は国内最高水準のデータセンターを全国に展開し、データセンターをベースにお客様に「アウトソーシング」、「クラウド」、「ソリューション」の三つのサービスを提供している。それら三つのサービスにおいて、システムの企画から設計、開発、運用、保守まで、お客様システムのライフサイクル全般をサポートしている。

サービスを提供するに当たり、当社に必要なものはお客様環境へ安心安全なシステムを導入するための「技術力」である。特にシステム基盤構築における技術力はお客様システムの品質に直結し、当社のデータセンター事業の軸となる。

当社の IT プロフェッショナル技術支援部は高い技術スキルを保有するプロフェッショナル社員を中心に組織化されており、そのスキルを「組織的」に有効活用することをミッションとして、当社の技術推進に貢献している。当部署では、技術推進の一環として、2013年に「インフラ構築強化チーム」を発足し、若手社員を中心にシステム基盤の技術力強化に取り組んでいる。

筆者は 2016 年 4 月に入社し、新人研修後に IT プロフェッショナル技術支援部に配属され、「インフラ構築強化チーム」の一員となった。

### 1. 2 インフラ構築強化チーム

インフラ構築強化チームとは、当社のシステム基盤構築における技術力強化を目的として発足したチームである。インフラ構築強化チームはシステム基盤構築を行う部署、またはシステム基盤構築のスキルを必要とする部署から選抜された部員で構成される。チームメンバーの目標はシステム基盤構築作業に必要な二つの能力を向上させることである。その二つの能力とは以下の通りである。

#### (1) 「個別製品技術力」

サーバ (PRIMERGY、SPARC Servers)、ネットワーク (Catalyst、IPCOM)、ミドルウェア (Systemwalker、Symfoware) などのシステム基盤を構成するそれぞれの要素に関する能力を指す。実プロジェクトでの経験はもちろんのこと、教育受講や資格取得により能力の向上を図る。

#### (2) 「基盤 SE 実践能力」

システム基盤作業を実践する上で必要となる能力を指す。富士通の基盤構築作業標準である ITIMAP (IT Infrastructure architectural Method And Process) に準拠し、7 種類の項目に分類されている。

筆者は 2 年の在籍期間中に複数のシステム基盤構築プロジェクトを経験し、OJT を通して上記の能力の向上に取り組んでいる。

## 2. 参画プロジェクト

筆者は新人研修終了後から約7ヶ月間、OJTの一環として、EDIサービスのシステムリリースを行うプロジェクトに参画した。本プロジェクトでは富士通のクラウドサービス「FUJITSU Cloud Service K5 IaaS（以下 K5 IaaS という）」上にEDIサービスのシステム基盤を構築する。構築台数が仮想マシン約170台（今後、顧客増加に合わせて拡張予定）と非常に多く、構築からテスト、運用の各工程における工数を削減する必要がある。そこで、新たな取り組みとして、構成管理ツール「Ansible」による構築自動化に取り組んだ。

図1に本プロジェクトで採用された構築自動化方式を示す。仮想マシン、仮想ネットワークの構築、OSインストールはK5 IaaSに搭載された機能を用いて自動化し、OS環境設定、ミドルウェアの導入はAnsibleを用いて自動化する。

筆者が担当した作業は開発用ユニットの構築自動化及び環境構築である。担当範囲の構築台数は約45台、作業期間は約2週間である。また、構築自動化は新たな取り組みであるため、社内ナレッジは無く、ノウハウも確立されていない状態である。

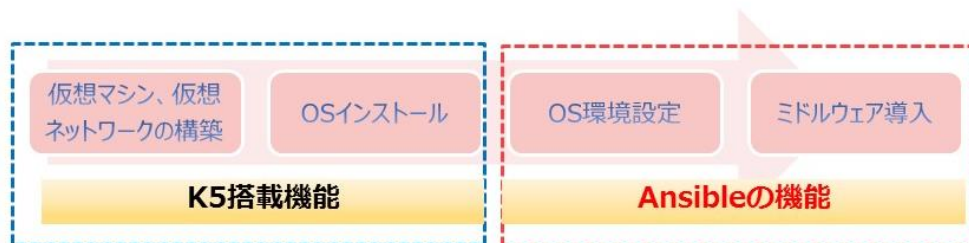


図1 構築自動化ツールを用いたサーバ構築の流れ

## 3. Ansible を用いた構築自動化

### 3.1 構築自動化とは

構築自動化とは、サーバの構築作業を構成管理ツール等を用いて自動化することである。従来、サーバの構築は手作業で行われており、以下のようなリスクに対してどのように対処するかが課題であった。

#### (1) 作業工数の増加

手作業によるサーバ構築の場合、1台ずつ構築しなければならないため、構築台数の増加に伴って、作業工数が増加する。

#### (2) 人為的な作業ミスの発生

手作業の場合、設定ミス、オペレーションミスといった人為的な作業ミスが起きる可能性がある。また、人為的である以上、上記の作業ミスを完全に抑制することはできない。

一方、Ansibleを用いた構築自動化では自動でOS環境設定、ミドルウェアの導入が可能である。上記の課題への対策として構築自動化を行った場合、手作業による構築作業と比較して下記のような効果が期待できる。

#### (1) 構築自動化ツールの実行による、複数台同時構築

各サーバの共通部分の構築作業は同時実行が可能である。さらに、変数処理を有効に活用することにより、OS 環境設定の大部分を同時実行することが可能となる。

#### (2) 人為的ミスの抑制

事前に作成した定義ファイルの内容にしたがって自動的に処理が行われるため、人為的ミスを抑制することができる。

Ansible では実行環境の導入や設定ファイルの作成が必要である。しかし、複数台を同時に構築することができるため、本プロジェクトのように構築台数が多い場合は構築に要する工数を大幅に削減することが可能だと考えられる。

以上のことから、本プロジェクトでは構築自動化ツールとして Ansible を選定した。

### 3. 2 Ansible について

Ansible はあらかじめ用意しておいた設定ファイルにしたがって、ミドルウェアの導入、サービスの起動/停止、ネットワーク設定といったサーバの各種設定を自動的に実行する構成管理ツールの一つである。対象サーバのホスト名を記載するインベントリ、処理内容を記載する playbook を事前に作成することで、構築作業を自動化することができる。

## 4. 構築自動化を進める上での課題と取り組み

### 4. 1 作業実施時に発生した課題

構築自動化を進める上で一つの大きな課題に直面した。それは、「実行時エラーが解消できない」という課題である。

筆者は事前に対象サーバに対する処理フローを作成し、そのフローにしたがって playbook を作成した(図 2)。しかし、動作確認を行ったところ、エラーが発生し異常終了

```
1 #install interstage application server+
2 #include e-exea-apl_vars.yml+
3 #
4 #hosts: apl+
5 #become: yes+
6 #vars_files:+
7 # - /opt/ansible/group_vars/all_servers_vars.yml+
8 # - /opt/ansible/group_vars/e-exea-apl_vars.yml+
9 #
10 #tasks:+
11 # - name: mkdir+
12 #   file: path=[[item]]+
13 #   state=directory+
14 #   with_items:+
15 #     - '{{ mountdir_apl }}'+
16 #     - '{{ target_dir_apl }}'+
17 #   tags:+
18 #     - interstage_apl_install+
19 #
20 # - name: copy iso file+
21 #   copy: src=[[ local_iso_path }}/Interstage_AS_V11_1_1.iso+
22 #         dest=[[ target_dir_apl ]]+
23 #   tags:+
24 #     - interstage_apl_install+
25 #
26 # - name: mount iso file+
27 #   mount: name=[[ mountdir_apl ]]+
28 #         src=[[ target_dir_apl }}/Interstage_AS_V11_1_1.iso+
29 #         fstype=iso9660+
30 #         opts=ro+
31 #         state=mounted+
32 #   tags:+
33 #     - interstage_apl_install+
```

**処理A**

- ・ミドルウェアのインストール、設定ファイル配置
- ・上記処理の終了後 サービスを再起動する。

**処理B**

- ・ミドルウェアの機能を用いた処理

※前提条件:  
サービスが起動していること

図 2 playbook の記述例

した。発生したエラーに対して、原因を特定して対処を行った後、再度動作確認を行った。その結果、他の箇所でもエラーが発生した。そして、他の箇所のエラーに対処すると、対処済みの元のエラーが再発し、エラーへの対処作業が循環するという状況に陥った(図 3)。

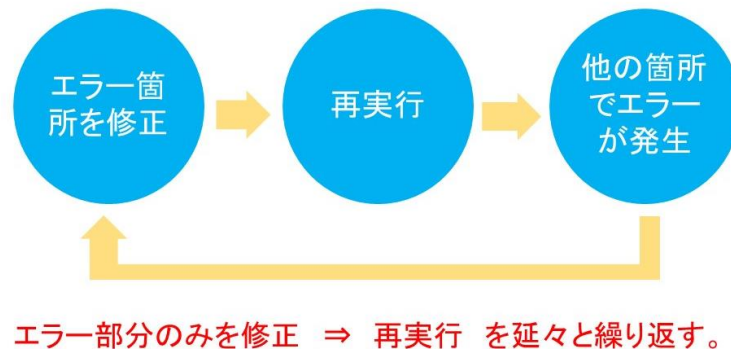


図3 エラーの循環

#### 4.2 課題への取り組み

前節で述べた課題を受けて、筆者は「Ansible の動作が想定と異なっているのではないか」という仮説を立てた。playbook に記述された処理の実行順序という観点で、再度マニュアルを読み直したところ、Ansible の一部の機能では、条件により処理の実行順序が入れ替わり、playbook の処理内容が変化することに気が付いた。図4に playbook の処理内容が変化する例を示す。図4ではサーバの実行条件により、サービス起動が処理Bの前に実行されるケースと処理Bの後に実行されるケースが存在する。後者の場合、処理Bの実行時にサービスが起動されていないため、実行時エラーが発生する。

このようなエラーの発生を防ぐため、筆者は二つの観点から playbook の再作成を行った。

- (1) 処理の実行順序に関して、実行条件の制約を受けない Ansible の機能を洗い出し、エラーの発生原因となっている機能の代わりに使用する。
- (2) 処理の実行順序が入れ替わらないように処理フローを再構築する。

これにより、図4の処理を図5のように修正することができる。図5では処理Bの前で確実にサービスを起動させることができるため、サーバの実行条件に依存せず、処理Bを正常に実行することができる。筆者は以上のような対処を重ね、再度 playbook を作成した。そして、動作確認を行ったところ、正常に動作することを確認することができた。本番の実行でも正常に動作し、無事、構築自動化を実現することができた。

筆者が担当した作業は開発用ユニットの構築自動化及び環境構築では、仮想サーバ45台の構築を10日間で行った。手作業で構築を行った場合の仮想サーバ1台あたりの構築にかかる工数を3.0Hとすると、仮想サーバ45台では135H(=約17日間)となるため、作業工数は約6割減となった。

さらに、作業期間10日間のうち Ansible の環境整備や playbook 作成を除いた、対象サ

サーバの実構築期間は 2 日間である。Ansible を導入することにより、障害発生時や仮想サーバの拡張などによる仮想サーバ再構築は playbook の再度実行することで実施できるため、さらに大きく作業工数を削減されることが期待できる。

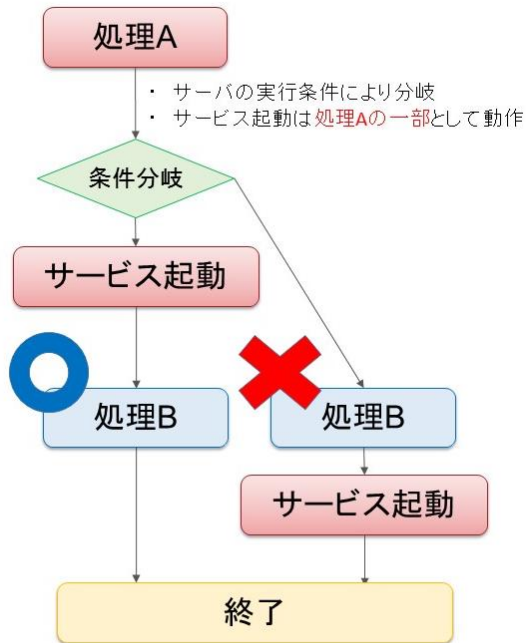


図 4 処理内容が変化する例

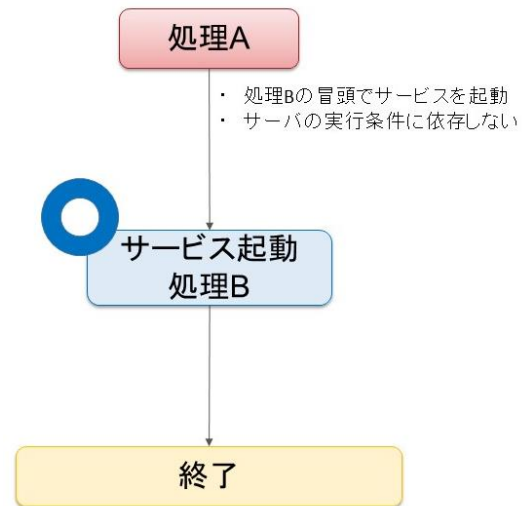


図 5 処理内容が変化しない例

## 5. ノウハウの共有と展開

筆者は Ansible を用いた構築自動化の実現という形でプロジェクトに貢献することができた。さらに、この経験を通して得られた Ansible に関するナレッジ、ノウハウを展開することでプロジェクトだけでなく原籍部署に対しても貢献することができた。本節では、その事例を紹介する。

### 5. 1 サーバ構築自動化に関する他の成果物に対する改善提案

プロジェクト内で既に作成されていた playbook の中には、可読性が低いものや実行条件によって実行順序が入れ替わる状態のものも多くあった。筆者は今回の構築経験から得られた知識を元に、「条件分岐処理の記述の簡略化」や「変数処理による playbook の一般化」といった改善提案を行った。

「条件分岐処理の記述の簡略化」は可読性を改善し、「変数処理による playbook の一般化」は管理する playbook やコンフィグファイルの個数を約 10%削減した。以上の改善提案を行うことで、プロジェクトの成果物の品質向上に貢献し、playbook を作成する際に注意すべき観点やノウハウを共有することができた。本プロジェクトでは試行錯誤を繰り返しながら playbook を作成した。今後は、これらのノウハウの共有により新規 playbook 作成や既存 playbook の修正に要する工数を、さらに 10%程度削減できる見込みである。

## 5. 2 インフラ構築強化チームへのノウハウの横展開

筆者は Ansible を用いた構築自動化のノウハウに関する資料を作成した。プロジェクト派遣元である IT プロフェッショナル技術支援部に対して、部会での発表等を通して共有し、ノウハウの横展開を行った。

情報展開を行った後、他の部員からの構築自動化についての問い合わせも 2 件あり、徐々に構築自動化への取り組みが広がっていることを実感している。

## 6. まとめ

### 6. 1 OJT 期間の成果と振り返り

筆者は OJT 期間の取り組みの中で、主に二つの成果を得ることができた。

- (1) 前例のない新たな取り組みでも、上流工程の設計書やマニュアルから必要な情報を引き出す力の向上

前述したとおり、Ansible を用いた構築自動化は新たな取り組みであり、社内ナレッジやノウハウは確立されていなかった。そのような環境の中でも、適切な観点を持った上で、上流工程の設計書やマニュアルを読み込み、必要な情報を引き出すことができるようになった。また、収集した情報を理解して成果物の作成に活かす力を養うことができた。

- (2) 改善提案を行い、プロジェクトの作業品質の向上に貢献する力の向上

自分の担当作業で得た知識、情報をプロジェクトメンバーに対して展開することができるようになった。また、新人という立場であっても成果物に対して改善案を提案することができるようになり、積極的にプロジェクトの作業品質の向上に貢献することができるようになった。

### 6. 2 今後の展望

筆者は現在、別のプロジェクトに派遣され、自治体関連のプロジェクトに参画している。現在参画中のプロジェクトでも Ansible による構築自動化を導入しており、Ansible に関するノウハウの展開を引き続き行っている。今後も新しい技術を積極的に取り入れ、その結果をナレッジとしてプロジェクト内、部署内にフィードバックし、部署全体の技術力向上に貢献していきたい。