



ITA_ファーストステップガイド

—第1.6.1版—

免責事項

本書の内容はすべて日本電気株式会社が所有する著作権に保護されています。

本書の内容の一部または全部を無断で転載および複製することは禁止されています。

本書の内容は将来予告なしに変更することがあります。

日本電気株式会社は、本書の技術的もしくは編集上の間違い、欠落について、一切責任を負いません。

日本電気株式会社は、本書の内容に関し、その正確性、有用性、確実性その他いかなる保証もいたしません。

商標

- ・ LinuxはLinus Torvalds氏の米国およびその他の国における登録商標または商標です。
- ・ Red Hatは、Red Hat, Inc.の米国およびその他の国における登録商標または商標です。
- ・ Apache、Apache Tomcat、Tomcatは、Apache Software Foundationの登録商標または商標です。
- ・ Ansibleは、Red Hat, Inc.の登録商標または商標です。

その他、本書に記載のシステム名、会社名、製品名は、各社の登録商標もしくは商標です。

なお、® マーク、TM マークは本書に明記しておりません。

※本書では「Exastro IT Automation」を「ITA」として記載します。

目次

目次.....	2
はじめに	3
1 ITA の概要	4
1.1 ITA とは.....	4
1.2 ITA の機能	4
1.2.1 構成管理機能	4
1.2.2 連携実行機能	4
1.2.3 ユーザー、権限管理.....	5
1.3 カスタマイズ機能	5
2 ITA のメニュー、画面構成	6
2.1 トップメニュー画面への接続.....	6
2.2 基本画面構成	9
3 ITA の操作手順	11
3.1 ITA におけるワークフローの考え方	11
3.2 ITA を利用した環境構築の手順	13
3.2.1 構築対象サーバーの登録	15
3.2.2 Movement(作業パターン)の登録.....	16
3.2.3 ワークフローの設定と実行(Symphony).....	18
3.2.4 ワークフローの設定と実行(Conductor)	19
3.3 2 台目、2 回以降の作業実行.....	21

はじめに

対象読者と目的

本書は、初めてご利用になるユーザーを対象に、ITA の機能、特徴、基本的な利用方法について説明します。

参考情報

Cobbler および、Cobbler Web の操作マニュアルは、Cobbler Project をご確認ください。

Cobbler Project : <http://cobbler.github.io/>

キックスタータファイルのリファレンスは、Redhat 社のインストールガイドをご確認ください。

Redhat : https://access.redhat.com/documentation/ja-jp/red_hat_enterprise_linux/

1 ITA の概要

1.1 ITA とは

ITA とは、サーバー・ストレージ・ネットワークなどの各機器に対して行う、環境構築や運用設定等の作業を管理、自動化するツールです。

ITA では、機器情報、設定値、作業履歴などを管理する「構成管理」機能と、各機器の構築、運用設定などをワークフローとして管理、実行する「連携実行」機能を提供します。各機器の構築、運用設定などは Ansible などの PF 構築ツールで行い、ITA ではこれらのツールと連携させるドライバーを提供します。また、これらの機能をブラウザ経由で操作する UI と、運用、保守に必要となるユーザー管理、権限管理などの機能を提供します。

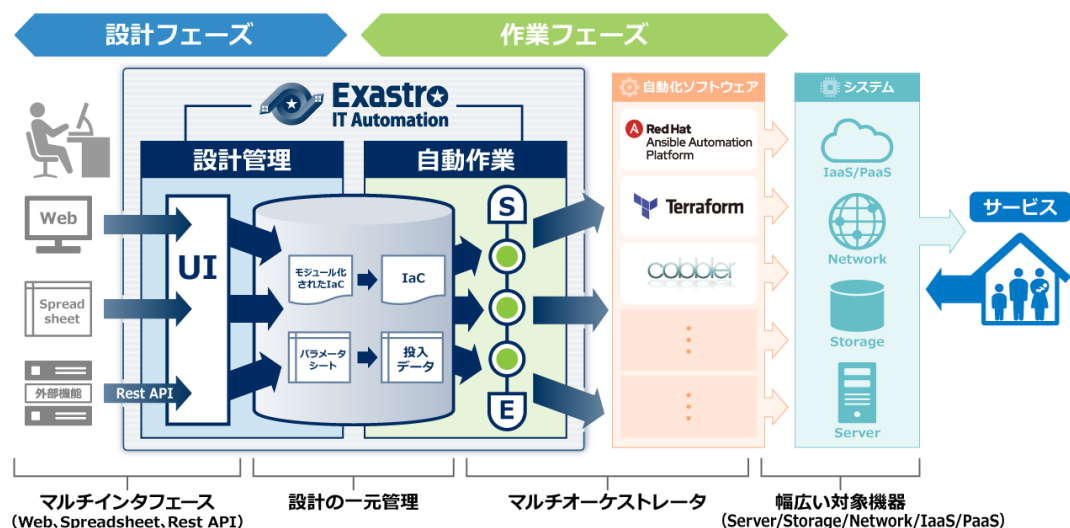


図 1.1-1 ITA の概要

1.2 ITA の機能

1.2.1 構成管理機能

- ITA では、構築、管理対象の機器情報、ネットワーク設定、OS 情報や各種設定情報を構成管理 DB で管理します。
 - 登録した情報は画面上から検索、履歴管理ができます。また一覧は“エクセル形式または独自 csv 形式”として出力でき、同じ形式でまとめて登録が可能です。

1.2.2 連携実行機能

- ITA では、さまざまな PF 構築ツールと連携することができます。各種ツールと連携するドライバーと、これらの実行を制御するワークフロー機能を提供します。
- ITA では以下のツールとの連携をサポートします。

表 1.2-1 連携ドライバー説明

ツール名	機能	内容
Ansible	システム構築	Red Hat 社が提供する OSS の PF 構築ツールです。Playbook と呼ばれる構築コードをもとに、ネットワークで接続された機器に対して、ソフトウェアのインストール、各種設定、ファイル転送、パッチの適用などを行います。
Cobbler	OS 構築	OSS のインストール自動化ツールです。あらかじめ作成したテンプレートを元に、ネットワークで接続された機器に対して、OS のインストールを行うことができます。
Terraform	システム構築	HashiCorp 社が提供するインフラストラクチャを効率化するオーケストレーションツールです。HCL(HashiCorp Configuration Language)という言語でコード化したインフラストラクチャ構成について、実行計画を生成したうえで構築を実行します。また、Policy as Code によるアクセスポリシーをコード化して管理することが可能です。

1.2.3 ユーザー、権限管理

ITA の利用者とその方の権限を管理します。

権限につきましては、利用者のアクセス可能なメニューや画面、操作制御(参照、更新)の設定を可能にします。

詳細については、利用手順マニュアル(ITA 管理コンソール)をご参照ください。

1.3 カスタマイズ機能

プロジェクトの要件に合わせて、ITA に機能を追加することができます。

ITA 付属のメニュー作成ツールを使って、プロジェクト独自の構成管理画面を作成し、ITA のメニュー追加することができます。

詳細は、「構成管理メニュー作成ガイド」をご参照ください。

表 1.3-1 カスタマイズ機能

機能名	機能	内容
メニュー作成ツール、メニュー管理機能	独自構成管理画面の追加	独自の構成管理画面を作成し、ITA のメニューに追加することができます。

2 ITA のメニュー、画面構成

2.1 トップメニュー画面への接続

ITA のトップメニュー画面への接続は、以下の手順となります。

準備作業

ITA 実装サーバーの IP アドレスとホスト名をご使用の環境の DNS サーバーまたは操作端末の hosts に設定してください。

ITA のログイン画面

ITA の URL へ接続します。ログイン画面が表示されたら、以下のログインID、初期パスワードを入力して、[ログイン]ボタンをクリックしてください。

URL	: http://([インストール時に設定した ITA ドメイン名] or [IP アドレス])
ログイン ID	: administrator
初期パスワード	: password



図 2.1-1 ログイン画面

インストール後に初めてログインした場合は、「パスワード変更画面」に遷移します。パスワード変更画面から、初期パスワードを変更してください。

(1) メインメニュー画面

ログイン直後の ITA のトップメニュー画面です。



図 2.1-2 メインメニュー画面

画面右にメインメニュー、画面左にサブメニューが表示されます。

メインメニューのメニューグループから各コンソールメニューを選択すると、選択したコンソールに対応するメニューがサブメニューに表示されます。メインメニューに戻る場合は、サブメニューの先頭の「メインメニュー」を選択してください。

(コンソールメニュー一覧)

表 2.1-1 メインメニューのコンソール一覧表

コンソール名	説明
管理コンソール	ITA システムの管理機能です。 ユーザー管理、権限管理、ITA システムの各種設定などを行います。
基本コンソール	ITA のシステムで共通に利用する機能です。 機器情報の登録、ワークフローの作成、実行などを行います。
エクスポート/インポート	ITA のデータを移行する機能です。 メニュー単位、または Symphony/オペレーション単位のデータの移行を行います。
Symphony	ITA でワークフローの作成、管理、実行を行う機能です。 Movement と呼ぶ作業パターンを組み合わせ、ワークフローを作成し、複数の機器に対して、一連の構築・設定などの作業を行います。
Conductor	Symphony と同じく、ITA でワークフローの作成、管理、実行を行う機能です。 Node と呼ぶ各種パーツを組み合わせ、ワークフローを作成し、複数の機器に対して、一連の構築・設定などの作業を行います。

メニュー作成	ITA 上で操作可能なメニューを作成する機能です。 作成したメニューは Web、Excel、RestAPI から操作でき、項目数や形式、入力制限などを自由に設計することが可能です。
ホストグループ	作業対象のホストを論理的な単位（機能・役割）のグループでまとめる機能です。 上位のホストグループから下位のホストグループへのパラメーター等の継承を行うことが可能です。
Ansible 共通コンソール	Ansible driver 共通の構成情報です。 Ansible driver, Ansible 本体のインストール設定などを管理します。
Ansible-Legacy コンソール	ITA から Ansible を利用する機能です。 Legacy コンソールでは、構築コードとして単体の YAML ファイルを使う場合に使用します。
Ansible-LegacyRole コンソール	ITA から Ansible を利用する機能です。 LegacyRole コンソールでは、構築コードとして製品部門などから提供された Role パッケージを使う場合に使用します。
Ansible-Pioneer コンソール	ITA から Ansible を利用する機能です。 Pioneer コンソールでは、SSH で接続されていない場合など、対象機器と対話的に直接やり取りしながら実行する場合に使用します。

各コンソールメニューについては、それぞれの利用手順マニュアルを参照してください。

2.2 基本画面構成

ITA が提供する各メニュー画面は、基本的に同じレイアウト構成になっています。



図 2.2-1 サブメニュー画面

各サブメニューに対して設定、登録などを行う画面を「メニュー操作部」と呼びます。このメニュー操作部の構成は以下のとおりです。

各画面項目に対する詳細な操作方法は、「利用手順マニュアル_基本コンソール」を参照してください。

表 2.2-1 メニュー操作部の説明

No	画面項目	説明
1	説明	各メニューの概要説明です。
2	表示フィルタ	各メニューで登録されている項目の検索条件を指定します。 ※メニュー画面を開いた段階では、項目には何も表示されません。 検索条件を空白にして「フィルタ」をクリックすると、すべての項目が表示されます。 項目を絞り込む場合は、検索条件を指定してください。
3	一覧/更新	表示フィルタで指定した検索条件に一致した項目の一覧が表示されます。 表示された各項目に対して更新、廃止などを実施することができます。
4	登録	各メニューに対して、新規に項目を登録します。 登録内容は各メニューによって異なりますので、各利用マニュアルを参照してください。

5	全件ダウンロードと ファイルアップ ロード編集 【一部の画面のみ】	各メニュー画面に登録されている情報を一括してエクセル形式でダウンロードすることができます。また、同じ形式のファイルで、一括して情報を登録することができます。
6	変更履歴	各メニュー画面で、登録した項目の変更履歴を表示することができます。

3 ITA の操作手順

3.1 ITA におけるワークフローの考え方

ITA では、作業の実行を「オペレーション名」と呼ぶ作業名と「Symphony(シンフォニー)」 / 「Conductor(コンダクター)」と呼ぶワークフローを紐付けて、「オペレーション名」の単位で行います。各機器に対して行う構築、設定作業を、「Movement(ムーブメント)」と呼ぶ作業パターンで作成し、ワークフローはこれらの作業パターンを組み合わせることで実行を指示します。

オペレーション名

- ・ITA での作業実行単位です。作業予定、実行履歴などを管理することができます。

Symphony

- ・ITA での一連の作業の単位です。オペレーション名と関連付けて実行します。
- ・Movement と呼ぶ作業パターンを組み合わせることで、ワークフローを作成し、複数の機器に対して、一連の構築・設定などの作業を行います。
- ・作業パターンを一つずつ(登録した Movement の実行順)で実行していきます。

Conductor

- ・ITA での一連の作業の単位です。オペレーション名と関連付けて実行します。
- ・Node と呼ぶ各種パーツを組み合わせることで、ワークフローを作成し、複数の機器に対して、一連の構築・設定などの作業を行います。
- ・Movement の並列実行、別のワークフローの呼び出し、Movement の実行結果による条件分岐によって、より複雑なワークフローの作成を可能にします。

Movement

- ・各機器に対する構築ツールを使った構築、設定などの作業の単位です。
- ・Movement は各ドライバーのコンソールメニューで作成します。作成方法はドライバーごとに異なります。作成方法は各ドライバーの利用手順マニュアルを参照してください。
- ・Movement は他の作業などでも再利用できるように、機能単位で作成することを推奨します。

(ワークフロー構成)

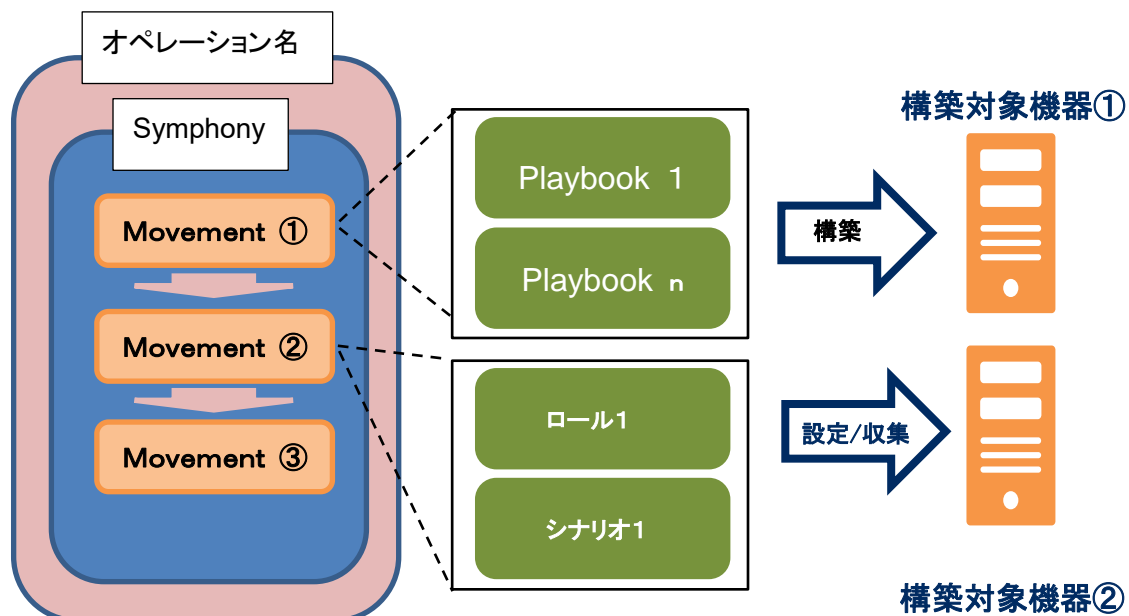


図 3.1-1 ワークフロー構成(Symphony)

(ワークフロー構成: Conductor)

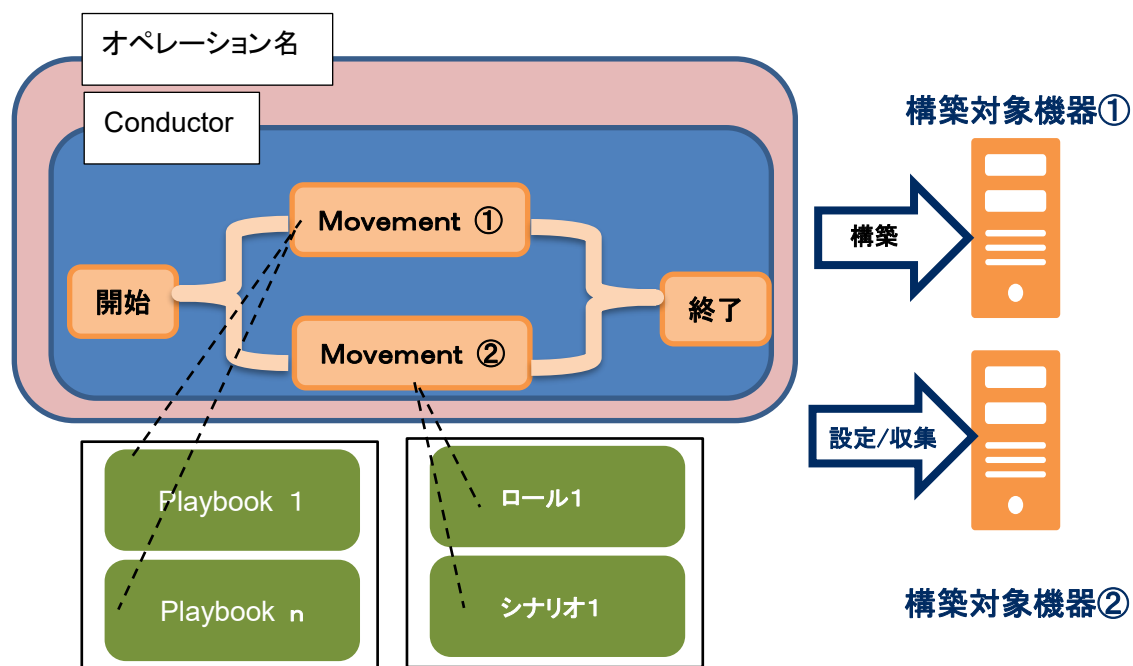


図 3.1-2 ワークフロー構成(Conductor)

3.2 ITA を利用した環境構築の手順

ITA を利用した、環境構築の標準的な作業の流れは以下のとおりです。

ここでは、対象サーバーの登録から、ワークフローの設定、実行までの一連の操作の流れを説明することで、全体の操作の流れを把握することを目的とします。

各手順での詳細な操作方法は、各コンソールの利用手順マニュアルを参照してください。

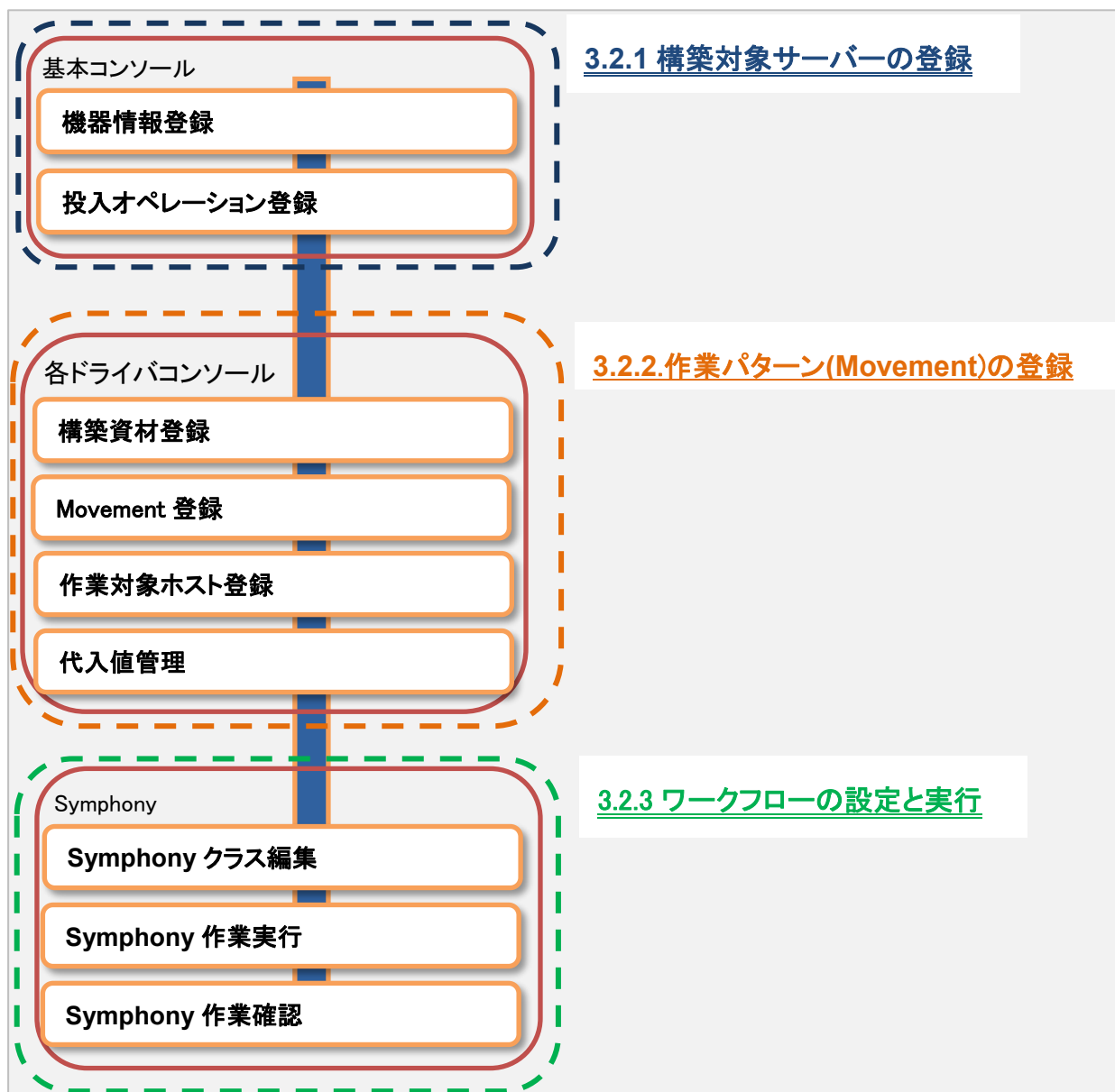


図 3.2-1 環境構築の作業(操作)の流れ(Symphony)

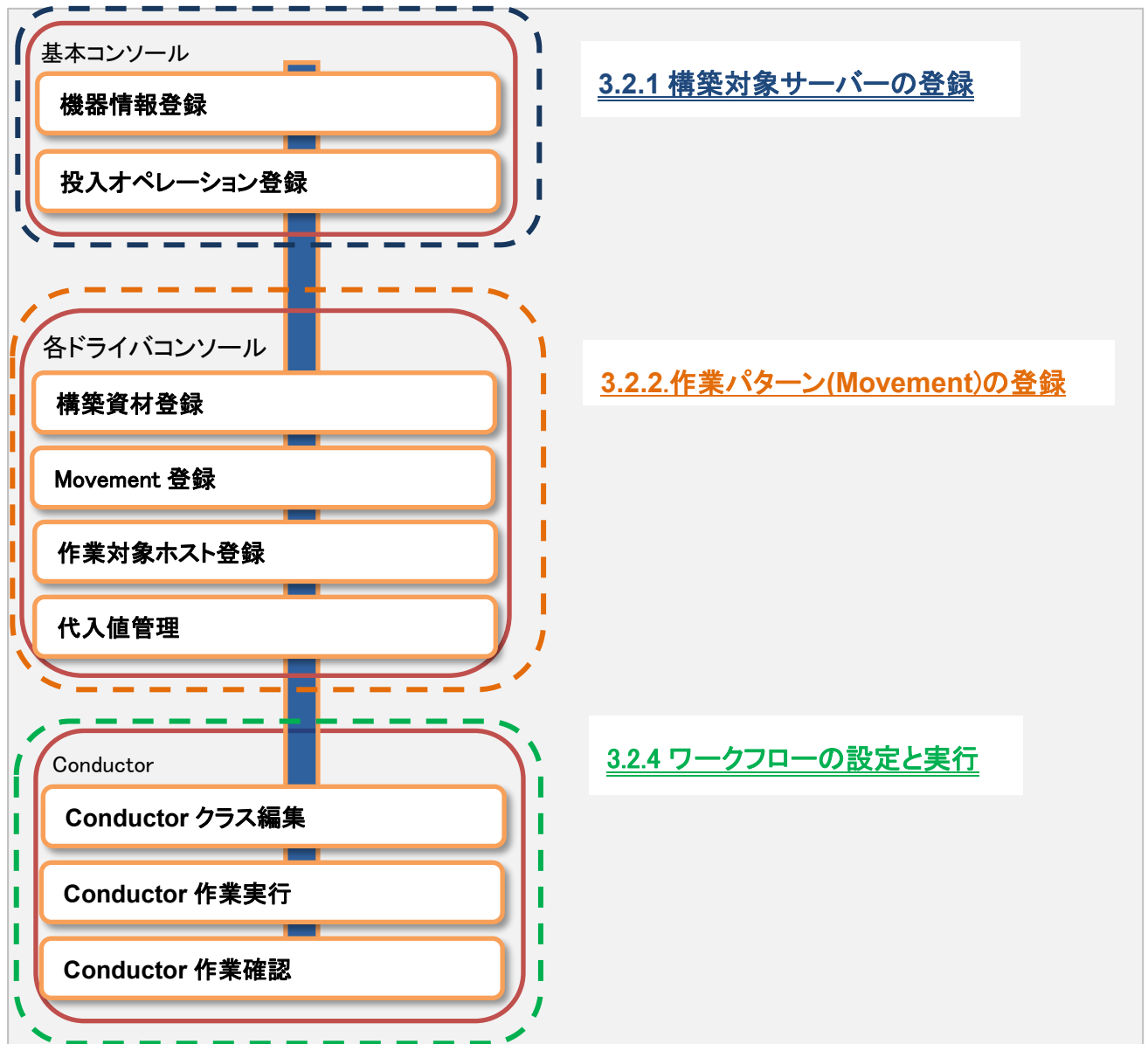


図 3.2-2 環境構築の作業(操作)の流れ(Conductor)

3.2.1 構築対象サーバーの登録

基本コンソールメニューから、構築、管理対象機器の情報を登録します。また、ワークフロー(Symphony)実行時のオペレーション名を登録します。

① 機器情報登録

「機器一覧」メニューから、構築、管理対象機器の情報(ホスト名、IP アドレス)と、接続のためのアカウント情報などを登録します。また、各機器に対する PF 構築ツールの利用情報を設定します。

詳細は、「利用手順マニュアル_基本コンソール」「4.1.2 機器一覧」を参照してください。

管理システム	HW種別	ホスト名*	IPアドレス*	EtherWakeOnLan		ログインユーザID	ログインパスワード		ssh設定ファイル
				MACアドレス	ネットワークデバイス名				
自動入力									
									最新アップロード アップロード状況

※*は必須項目です。

戻る 登録

図 3.2-3 登録画面(機器一覧)

② 投入オペレーション登録

「投入オペレーション一覧」メニューから、ワークフローとして実行するオペレーション名と実行予定日を登録します。

詳細は、「利用手順マニュアル_基本コンソール」「投入オペレーション一覧」を参照してください。

No.	オペレーション名*	実行予定日*	オペレーションID	備考	最終更新日時	最終更新者
自動入力			自動入力		自動入力	自動入力

※*は必須項目です。

戻る 登録

図 3.2-4 登録画面(投入オペレーション一覧)

3.2.2 Movement(作業パターン)の登録

各ドライバーのコンソールから、各機器に対する構築、設定を Movement(作業パターン)として作成します。

Movement(作業パターン)の作成方法は、各ドライバー、コンソールメニューによって異なります。各利用手順マニュアルの作業フローを参照してください。

ここでは例として、Ansible Legacy ドライバーでの PF 構築の Movement(作業パターン)の作成方法を説明します。詳細な手順は利用手順マニュアル(Ansible driver) の作業フローを参照してください。

① 構築資材登録

「プレイブック素材集」メニューから、Ansible の構築コードである Playbook(YAML ファイル)を登録します。

② Movement 登録

「Movement 一覧」メニューから Movement 名を登録し、「Movement 詳細」メニューから実行するプレイブック素材を紐づけます。

The image displays two screenshots of a web-based registration interface. The left screenshot shows the 'Movement' registration form, which includes fields for 'Movement' (with a dropdown menu), 'Playbook material' (with a dropdown menu), and 'Include sequence' (with a text input field). Below these fields is a note: '※*は必須項目です。' (※* is a required item). At the bottom are two orange buttons: '戻る' (Back) and '登録' (Register). The right screenshot shows the 'Movement List' registration form, which includes fields for 'MovementID', 'Movement名' (Movement name), '遅延タイマー' (Delay timer), 'Ansible利用情報' (Ansible usage information), and '備考' (Remarks). The 'Ansible利用情報' section is expanded, showing sub-fields for 'ホスト指定形式*' (Host specification format), '並列実行数' (Parallel execution count), 'WinRM接続' (WinRM connection), and 'gather_facts'. Below these fields is a note: '※*は必須項目です。' (※* is a required item). At the bottom are two orange buttons: '戻る' (Back) and '登録' (Register).

図 3.2-5 登録画面(Movement 一覧)

③ 作業対象ホスト登録

「作業対象ホスト」メニューから、実行対象のホストを設定します。

④ 代入値管理

ITA では Playbook に定義した変数の値を、ITA の画面から指定することができます。

※変数は ITA 特有の命名規則に準ずる必要があります。

機器ごとに異なる設定を、Playbook の変更なしで、ITA の管理項目として設定、管理することができます。

必要に応じて「代入値管理」メニューから、変数値を設定します。

⑤ 作業実行

Movement(作業パターン)を単体として実行することができます。

また、各対象ホストに作業を反映させずに、Playbook の記述内容をチェックするドライラン機能があります。

3.2.3 ワークフローの設定と実行(Symphony)

Symphony メニューから、3.2.2 で作成した Movement(作業パターン)を組み合わせて、ワークフロー(Symphony)を登録し、対象機器に対して作業の実行を指示します。

① Symphony クラス編集

Symphony の「Symphony クラス編集」メニューから、作業パターン(Movement)を組み合わせて、ワークフロー(Symphony)を作成します。

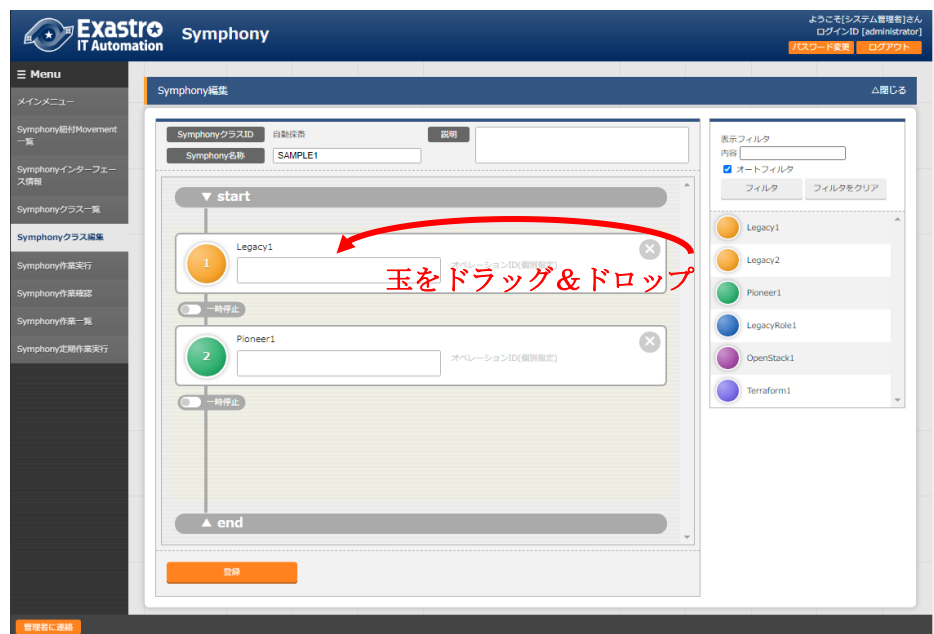


図 3.2-6 Symphony クラス編集画面

② Symphony 作業実行

Symphony の「Symphony 作業編集」サブメニューを使用し、実行する Symphony とオペレーション名を選択し、Ansible などの構築ツールへ作業実行を指示します。

③ Symphony 作業確認

Symphony の「Symphony 作業実行確認」サブメニューを使用し、作業結果を確認します。

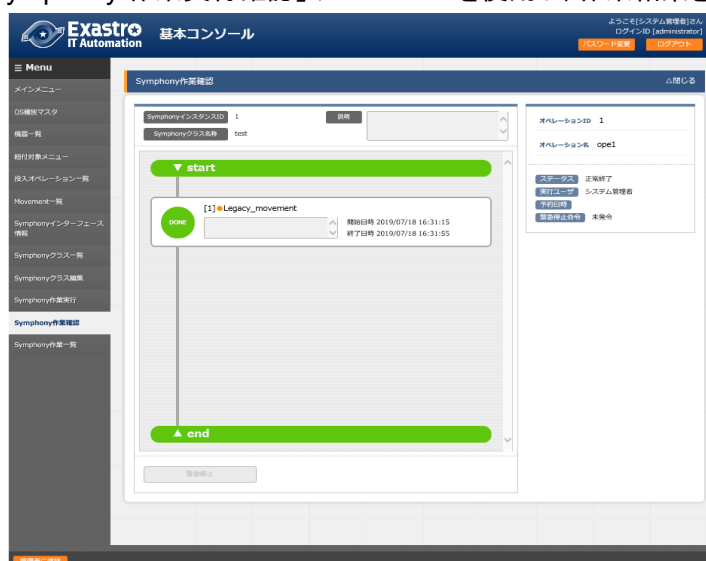


図 3.2-7 Symphony 作業確認画面

3.2.4 ワークフローの設定と実行(Conductor)

Conductor メニューから、3.2.2 で作成した Movement(作業パターン)を組み合わせて、ワークフロー(Conductor)を登録し、対象機器に対して作業の実行を指示します。

① Conductor クラス編集

Conductor の「Conductor クラス編集」メニューから、作業パターン(Movement)を組み合わせて、ワークフロー(Symphony)を作成します。

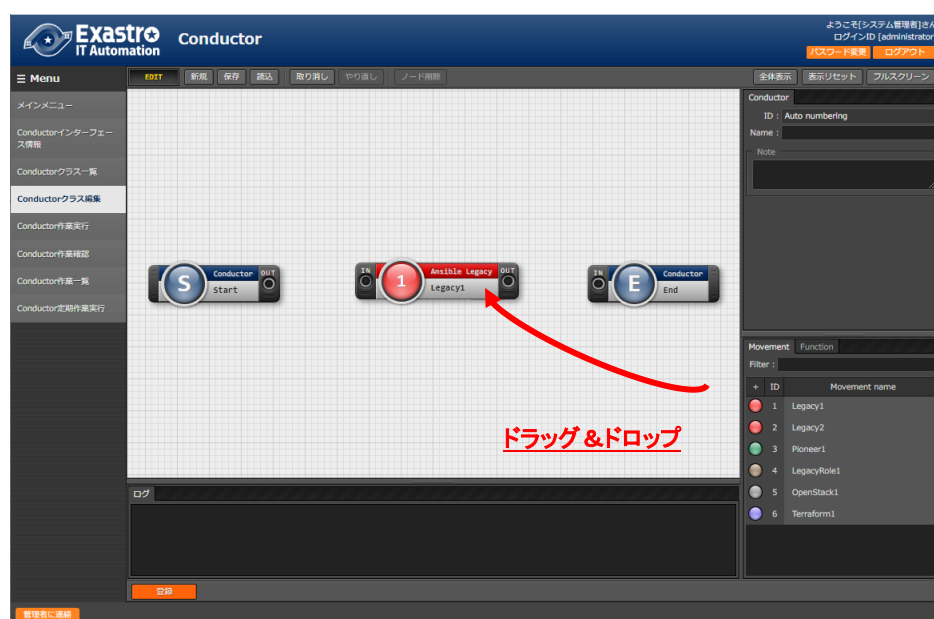


図 3.2-8 Conductor クラス編集画面

② Conductor 作業実行

Conductor の「Conductor 作業編集」サブメニューを使用し、実行する Conductor とオペレーション名を選択し、Ansible などの構築ツールへ作業実行を指示します。

③ Conductor 作業確認

Conductor の「Conductor 作業実行確認」サブメニューを使用し、作業結果を確認します。

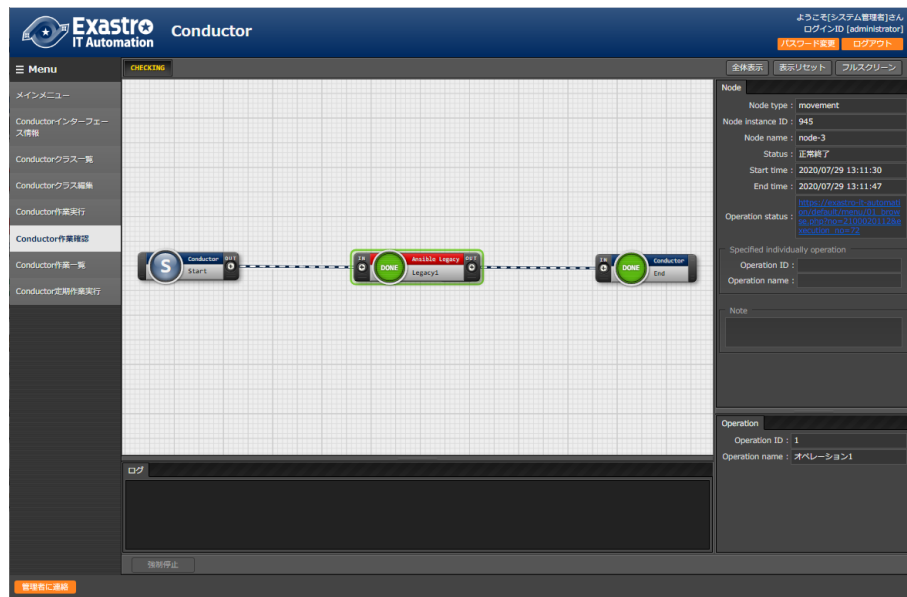


図 3.2-9 Conductor 作業確認画面

3.3 2 台目、2 回以降の作業実行

類似環境の構築において、2 台目、2 回目以降の作業実行では、作成した Movement (作業パターン)、ワークフローを再利用することができます。

初回構築時には、「3.2 ITA を利用した環境構築の手順」で説明した操作が必要となりますが、2 台目、2 回目以降の作業では以下の作業フロー (2 台目以降のサーバー構築) となります。緑色で囲んだ機器情報登録、投入オペレーション登録、作業対象ホスト登録、代入値管理を構築仕様に応じて変更して、ワークフロー (Symphony / Conductor) に実行の指示を与えるだけで、2 台目、2 回目以降の環境を構築することができます。

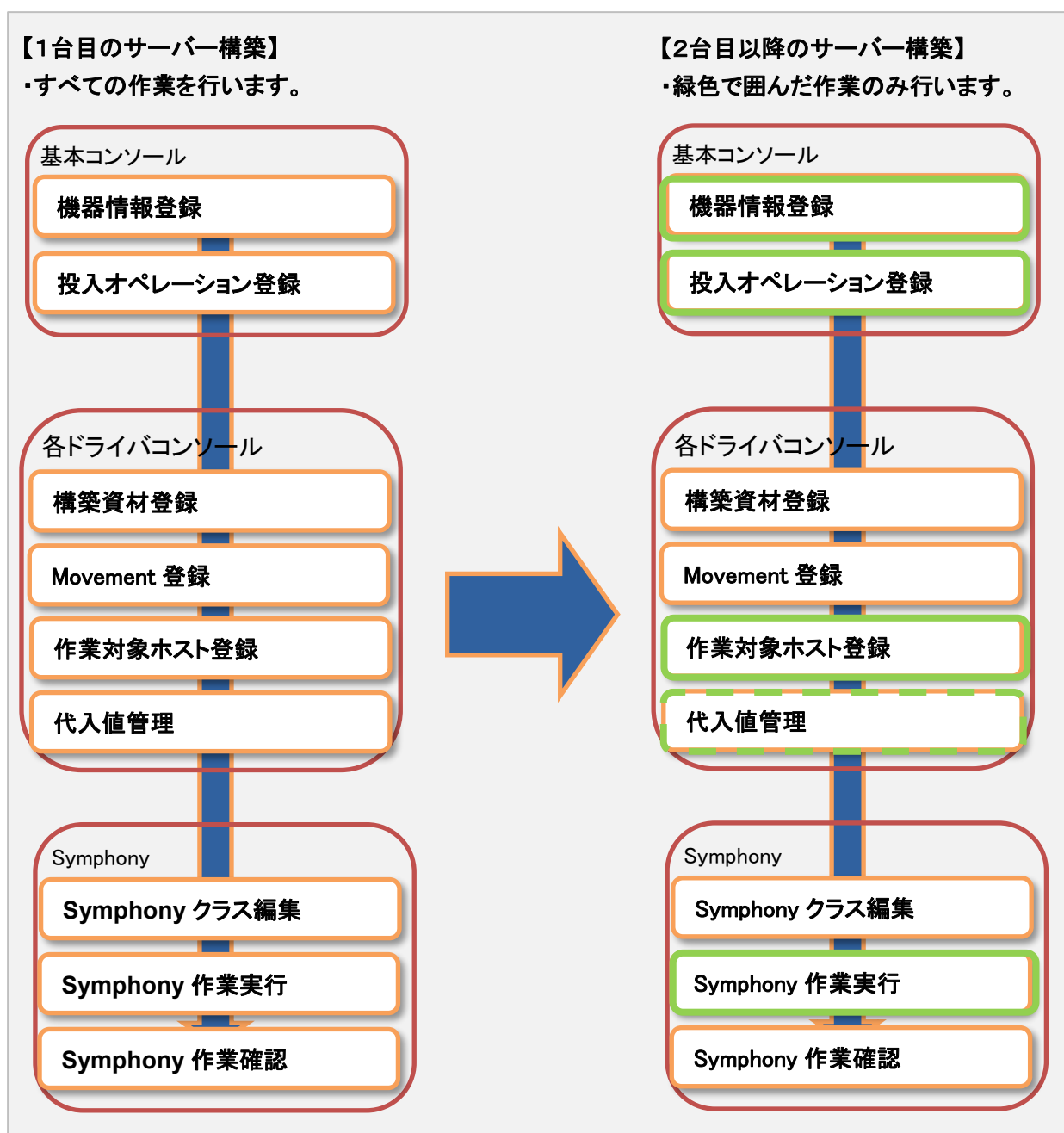


図 3.3-1 類似環境における 2 回目以降のワークフロー