



利用手順マニュアル

astrollAnsible driver

—第1.0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。

astroll の正式名称は「astroll IT Automation」になります。

目次

はじめに	4
1 Ansible driver 概要	6
1.1 Ansible について	6
1.2 Ansible driver について	7
2 Ansible driver での変数取り扱い	8
2.1 変数の種類	8
2.2 変数の抽出および具体値登録	8
2.3 代入値登録による変数の扱い	8
3 Ansible driver コンソールメニュー構成	9
3.1 メニュー/画面一覧	9
4 Ansible driver 利用手順	11
4.1 作業フロー	11
4.1.1 Ansible-Legacy 作業フロー	12
4.1.2 Ansible-Legacy Role 作業フロー	14
4.1.3 Ansible-Pioneer 作業フロー	16
5 Ansible driver 機能・操作方法説明	19
5.1 基本コンソール	19
5.1.1 機器一覧	19
5.1.2 紐付対象メニュー	22
5.1.3 投入オペレーション一覧	23
5.2 Ansible 共通コンソール	24
5.2.1 インタフェース情報	24
5.2.2 グローバル変数管理	26
5.3 Ansible-Legacy / Legacy Role / Pioneer コンソール	28
5.3.1 Movement 一覧	28
5.3.2 プレイブック素材集 (Ansible-Legacy のみ)	29
5.3.3 ロールパッケージ管理 (Ansible-Legacy Role のみ)	31
5.3.4 対話種別リスト (Ansible-Pioneer のみ)	32
5.3.5 対話ファイル素材集 (Ansible-Pioneer のみ)	33
5.3.6 Movement 詳細	35
5.3.7 テンプレート管理 (Ansible-Legacy / Ansible-Pioneer)	37
5.3.8 ファイル管理	38
5.3.9 多段変数最大繰返数管理 (Ansible-Legacy Role のみ)	39
5.3.10 代入値自動登録設定	40
5.3.11 作業対象ホスト	41
5.3.12 代入値管理	42
5.3.13 作業状態確認	44
5.3.14 作業管理	46
5.3.15 作業実行	47
6 構築コード記述方法	48
6.1 Playbook (Ansible-Legacy) の記述	48
6.2 対話ファイル (Ansible-Pioneer) の記述	48
6.3 ロールパッケージ (Ansible-Legacy Role) の記述	58
6.4 astrollreadme (Ansible-Legacy Role のみ) の記述	58

6.5	読替表 (Ansible-Legacy Role のみ) の記述	58
6.6	BackYard コンテンツ	58
6.7	Ansible 利用ガイドライン astroll 追加ルール	58
7	運用上の注意点	59
7.1	ログレベルの変更	59
7.2	起動周期の変更	61
7.3	メンテナンス方法について	63
7.3.1	Ansible driver 独立型プロセスの起動/停止/再起動	63
8	トラブルシューティング	64
9	付録	65
9.1	Ansible 実行時に使用される投入データと astroll メニューの紐づけ	65
9.1.1	Ansible-Legacy 投入データ	65
9.1.2	Ansible-Pioneer 投入データ	67
9.1.3	Ansible-LegacyRole 投入データ	69

はじめに

本書では、astroll の機能および操作方法について説明します。

関連マニュアル

- ・ 本製品におけるマニュアル構成は以下になります。

Ansible driver を初めて利用される方は、まずファーストステップガイドで全体の操作の流れを把握し、利用手順マニュアル(astroll 基本コンソール)で共通画面、機器情報登録画面などの操作方法を参照してください。

また、構築作業を定義するプレイブックは、作成において記述規定がありますので、「Ansible 利用ガイドライン」を参照ください。

No.	マニュアル名	説明
1	インストールマニュアル astroll	astroll のシステム構成、動作環境と、環境構築、インストール手順の概要を説明したドキュメントです。
2	RHEL6.x_環境構築マニュアル	インストールマニュアルの別冊資料です。 RHEL 6.x 環境への環境構築、インストール手順を説明します。
3	RHEL7.x_環境構築マニュアル	インストールマニュアルの別冊資料です。 RHEL 7.x 環境への環境構築、インストール手順を説明します。
4	ファーストステップガイド	astroll の全体の概要、機能、操作の概要について説明したドキュメントです。astroll を初めて利用される方は、本ドキュメントで全体の操作の流れを参照してください。
5	利用手順マニュアル astroll 基本コンソール	astroll の基本機能である基本コンソールの機能、操作方法について説明したドキュメントです。機器情報の登録、ワークフローの作成、実行などについて説明しています。各 Driverをご利用の方は、本ドキュメントも合わせて参照してください。
6	利用手順マニュアル astroll 管理コンソール	astroll の管理機能である管理コンソールの機能、操作方法について説明したドキュメントです。 ユーザー管理、権限管理、astroll システムの設定機能などを説明します。
7	【本書】 利用手順マニュアル astrollAnsible driver	Ansible driver の機能、操作方法について説明したドキュメントです。
8	利用手順マニュアル astrollAnsible driver 別紙 Ansible 利用ガイドライン astroll 追加ルール	Ansible driver 利用手順マニュアルの補足資料です。 astroll から Ansible を利用するときの注意、制限事項と、エラーメッセージとその対処方法などについて説明します。
10	利用手順マニュアル astrollCobbler driver	Cobbler driver の機能、操作方法について説明したドキュメントです。

No.	マニュアル名	説明
11	構成管理メニュー作成ガイド	独自の構成管理画面を作成し、astroll のメニューに追加する手順を説明したドキュメントです。
12	利用手順マニュアル astroll_Ansible 系ドライバー共通	Ansible 系統のドライバーにおける共通機能および操作方法について説明したドキュメントです。

1 Ansible driver 概要

本章では Ansible および Ansible driver について説明します。

1.1 Ansible について

Ansible とは、多数の構築管理対象に対して、アプリケーション/システムのデプロイ作業を容易にする PF 構築自動化ツールです。

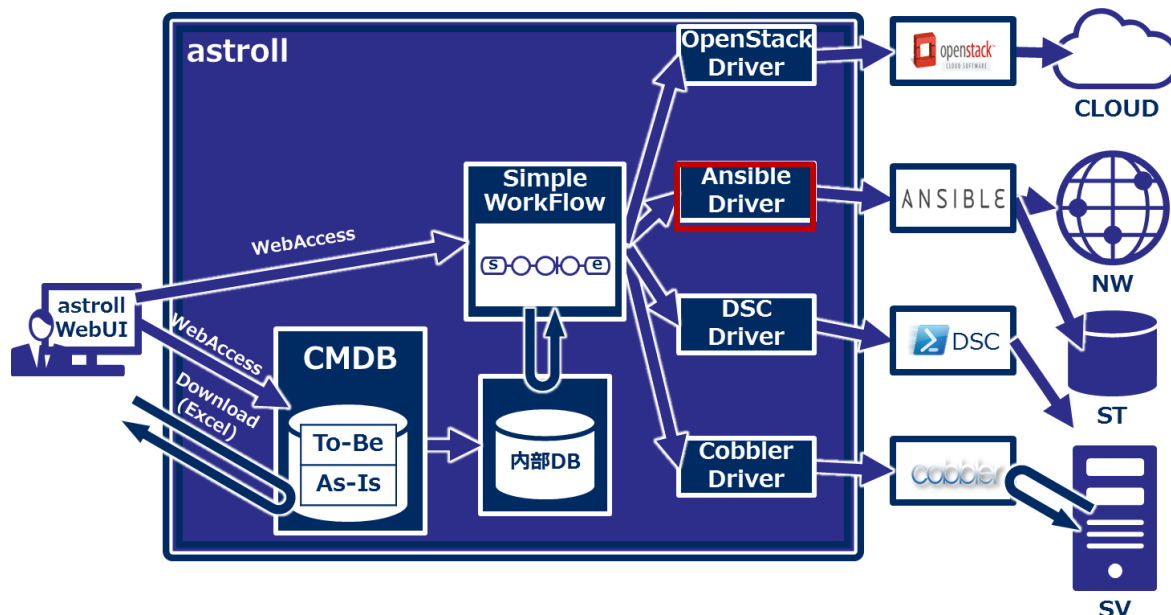
Ansible は、Playbook という YAML 形式のテキストファイルに定型処理をタスクとして記述し、それを Ansible に実行させることにより、さまざまな処理を実現できます。
タスクはモジュールと呼ばれる処理プログラムと紐付いており、さまざまな機器に対する制御を行うことができます。

Ansible の詳細情報については、Ansible の製品マニュアルを参照してください。

Ansible のバージョンは 2.2.1.0 となります。最新の Ansible に対応した記法は使えないことがありますので、注意してください。

1.2 Ansible driver について

Ansible driver は、astroll システムのオプションとして機能し、astroll システムで登録した構築対象のサーバ・ストレージ・ネットワーク各機器に対し、Ansible を用いて実際の運用設定を自動的行います。



Ansible driver には用途に応じて以下 3 つのモードを用意しています。

① Legacy モード

Ansible 標準の機能を用いて各種ホストへ設定を投入します。

構築コードを単体 YAML ファイルとして登録し、作業パターンをその組み合わせで構成します。

OS, NW の環境設定などの作業用に使われることを想定します。

② Legacy Role モード

Legacy モードと同じく、Ansible 標準の機能を用いて各種ホストへ設定を投入します。

構築コードをパッケージとして登録し、作業パターンを Role の組み合わせで構成します。

製品部門などが提供する Role パッケージを用いて、製品のインストール、環境構築などを行う際に使われることを想定します。

③ Pioneer モード

Ansible に独自モジュールを追加し、対話形式による設定投入を可能とします。

サーバ、ストレージ、ネットワークを問わず、Telnet, SSH でログイン可能なあらゆる機器に対応しています。対象機器と直接やり取りが必要となるため、相応のITスキルが必要となります。

また、Ansible driver は、Playbook 中の変数を画面から設定することができます。詳細は本書「[2Ansible driver での変数取り扱い](#)」をご参照ください。

2 Ansible driver での変数取り扱い

2.1 変数の種類

Ansible driver では、Playbook 中の変数を astroll の設定画面から指定することができます。
詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

2.2 変数の抽出および具体値登録

各モードとも、astroll にアップロードした Playbook 等の資材から変数を抜き出します。
詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

変数の抽出方法は以下のとおりです。

(1) Ansible-Legacy

「プレイブック素材集 (本書: [5.3.2 プレイブック素材集 \(Ansible-Legacy のみ\)](#))」でアップロードした Playbook より、以下の書式の変数定義を抽出します。

`{{△VAR_xxx△}}` または `{%△VAR_xxx△%}`

※ △: 半角スペース xxx: 半角英数字とアンダースコア (_)

Ansible-Legacy では具体値の登録の仕方で通常変数か複数具体値変数かを決定します。詳しくは「[5.3.12 代入値管理](#)」を参照してください。

(2) Ansible-Pioneer

「対話ファイル素材 (本書: [5.3.5 対話ファイル素材集 \(Ansible-Pioneer のみ\)](#))」でアップロードした対話ファイルより、Ansible-Legacy と同様の抽出を行います。

(3) Ansible-Legacy Role

「ロールパッケージ管理 (本書: [5.3.3 ロールパッケージ管理 \(Ansible-Legacy Role のみ\)](#))」でアップロードしたロールパッケージ内の Playbook より、Ansible-Legacy と同様の抽出を行います。

詳しくは「ロールパッケージの記述 (本書: [5.3.3 ロールパッケージ管理 \(Ansible-Legacy Role のみ\)](#))」を参照してください。

また、読替表を作成することで defaults 変数定義ファイルまたは astrollreadme に定義されている「VAR_xxx」以外の変数を astroll で扱うことが出来ます。詳しくは「[6.5 読替表の記述](#)」を参照して下さい。

2.3 代入値登録による変数の扱い

Playbook で定義した変数の値は代入値登録機能により上書きすることができます。
詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

3 Ansible driver コンソールメニュー構成

本章では、astroll コンソールのメニュー構成について説明します
なお、Web コンソールへのログイン方法、およびメニュー画面の構成要素/基本的な操作については、「[利用手順マニュアル\(astroll システム\)](#)」を参照してください。

3.1 メニュー/画面一覧

① astroll 基本コンソールのメニュー

Ansible driver で利用する astroll 基本コンソールのメニュー一覧を以下に記述します。

表 3.1-1 基本コンソール メニュー/画面一覧

No	メニューグループ	メニュー・画面	説明
1	astroll 基本コンソール	機器一覧	作業対象システム一覧をメンテナンス(閲覧/登録/更新/廃止)します
2		紐付対象メニュー	代入値自動登録設定と連携する CMDB を管理します
3		投入オペレーション一覧	投入オペレーション一覧をメンテナンス(閲覧/登録/更新/廃止)できます

② Ansible 共通コンソールのメニュー

Ansible 共通コンソールのメニュー一覧を以下に記述します。

表 3.1-2 共通コンソール メニュー/画面一覧

No	メニューグループ	メニュー・画面	説明
1	Ansible 共通コンソール	インタフェース情報	astroll システム・Ansible driver サーバと Ansible サーバが共有するディレクトリのパスおよび Ansible サーバへの接続インタフェース情報を管理します
2		グローバル変数管理	Playbook や対話ファイルなどで共通利用する変数(以降、グローバル変数と称す)と具体値を管理します

③ Ansible コンソールのメニュー

各 Ansible コンソールに対応するメニュー一覧を以下に記述します。

表 3.1-3 Ansible driver コンソール メニュー/画面一覧

No	メニューグループ			メニュー・画面	説明
	Ansible コンソール				
	Legacy	Legacy Role	Pioneer		
1	○	○	○	Movement 一覧	Symphony に登録する Movement の一覧を管理します
2	○			プレイブック素材集	Playbook ファイルを管理します
3		○		ロールパッケージ管理	ロールパッケージを管理します
4			○	対話種別リスト	同一目的の対話ファイルを対話種別としてまとめる種別を管理します
5			○	対話ファイル素材集	対話種別に紐づける OS 種別と astroll システム独自フォーマットの作業手順ファイル(以降、対話ファイルと称す。)を管理します
6	○	○	○	Movement 詳細	Movement とプレイブック素材の関連付けを管理します
7	○		○	テンプレート管理	Playbook 内の template モジュールで使用するテンプレートファイルと埋め込み変数を管理します
8	○	○	○	ファイル管理	Playbook 内の各モジュールで使用する素材ファイルと埋め込み変数を管理します
9		○		多段変数最大繰返数管理	多段変数が繰返配列で構成されている場合の最大繰返配列数を管理します
10	○	○	○	代入値自動登録設定	CMDB のメニューに登録されているオペレーションとホスト毎の項目値を紐付ける Movement と変数を管理します
11	○	○	○	作業対象ホスト	Movement で使用するホストを管理します
12	○	○	○	代入値管理	変数の代入値を管理します
13	○	○	○	作業実行	作業実行する Movement とオペレーションを選択し実行を指示します
14	○	○	○	作業状態確認	作業実行状態を表示します
15	○	○	○	作業管理	作業実行履歴を管理します

※ Ansible RestAPI は、設定等ユーザー操作を行わないため、ここでの説明は省略します。

4 Ansible driver 利用手順

各 Ansible コンソールの利用手順について説明します

4.1 作業フロー

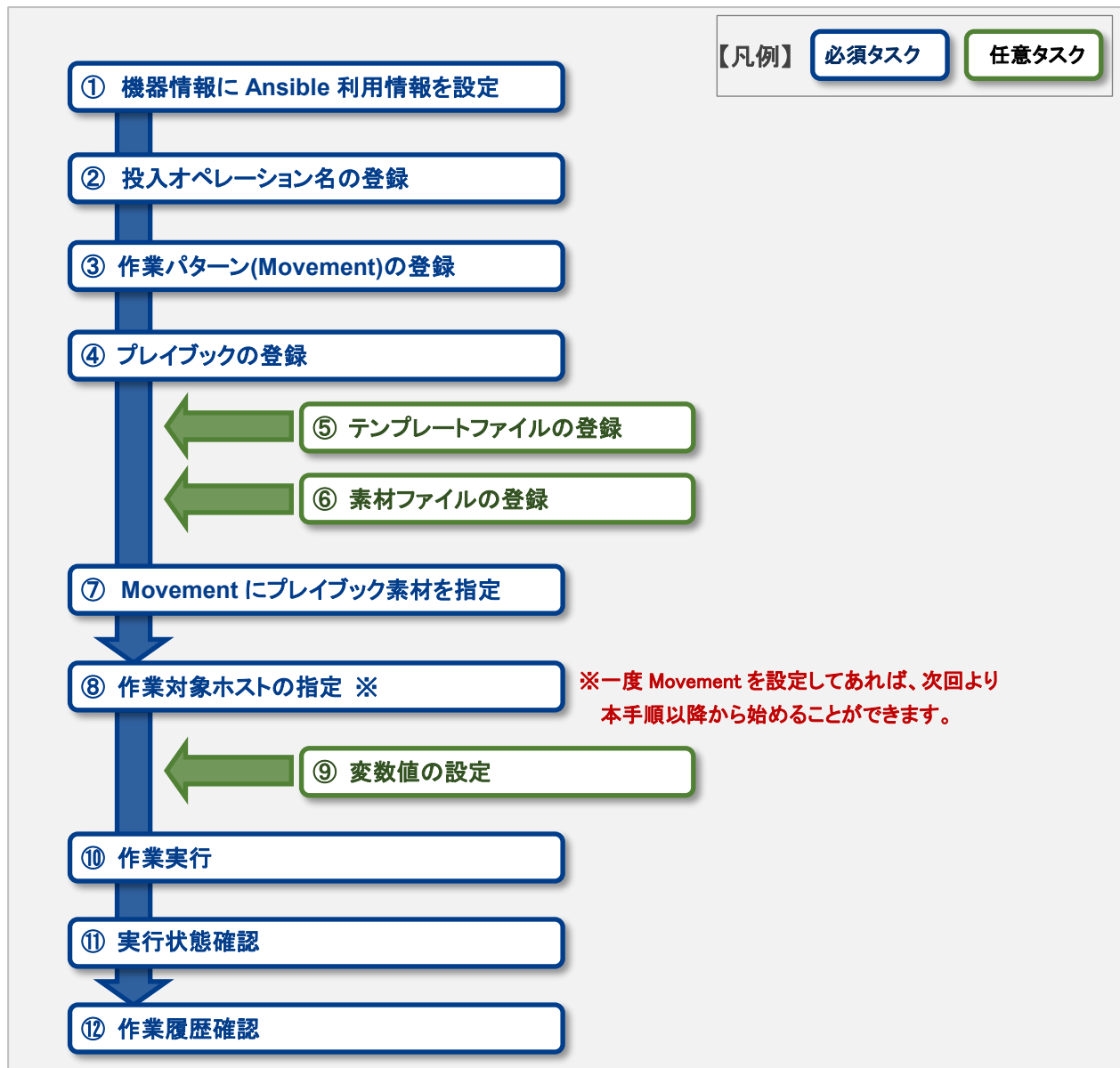
各 Ansible コンソールにおける標準的な作業フローは以下のとおりです。

各作業の詳細は次項に記載しています。

astroll 基本コンソールの利用方法は、「利用手順マニュアル(astroll_基本コンソール)」を参照してください。

4.1.1 Ansible-Legacy 作業フロー

以下は、Ansible-Legacy で作業を実行するまでの流れです。



● 作業フロー詳細と参照先

① 機器情報に Ansible 利用情報を設定

astroll 基本コンソールの機器一覧の画面から、各機器に対して Ansible 利用情報を設定します。
詳細は [5.1.1 機器一覧](#) を参照してください。

② 投入オペレーション名の登録

astroll 基本コンソールの投入オペレーション一覧の画面から、作業用の投入オペレーション名を登録します。
詳細は [5.1.3 投入オペレーション一覧](#) を参照してください。

③ **作業パターン(Movement)の登録**

Ansible-Legacy コンソールの Movement 一覧の画面から、作業用の Movement を登録します。
詳細は [5.3.1 Movement 一覧](#) を参照してください。

④ **Playbook の登録**

Ansible-Legacy コンソールのプレイブック素材集の画面から、作業で使用する Playbook を登録します。
詳細は [5.3.2 プレイブック素材集 \(Ansible-Legacy のみ\)](#) を参照してください。

⑤ **テンプレートファイルの登録(必要に応じて実施)**

Ansible-Legacy コンソールのテンプレート管理の画面から、Playbook 内で定義している template モジュールの template ファイル(src)と template 埋め込み変数の登録／更新／廃止を行います。
詳細は [5.3.7 テンプレート管理 \(Ansible-Legacy/Ansible-Pioneer\)](#) を参照してください。

⑥ **素材ファイルの登録 (必要に応じて実施)**

Ansible-Legacy コンソールのファイル管理の画面から、作業対象サーバに配置するファイルを登録します。
詳細は [5.3.8 ファイル管理](#) を参照してください。

⑦ **Movement にプレイブック素材を指定**

Ansible-Legacy コンソールの Movement 詳細の画面から、登録した Movement にプレイブック素材を指定します。
詳細は [5.3.6 Movement 詳細](#) を参照してください。

⑧ **作業対象ホストの指定**

Ansible-Legacy コンソールの作業対象ホストの画面から、作業対象ホストを指定します。
詳細は [5.3.11 作業対象ホスト](#) を参照してください。

⑨ **変数値の設定(必要に応じて実施)**

Ansible-Legacy コンソールの代入値管理の画面から、Movement に登録した Playbook 内で定義した変数の値を設定します。変数を利用していない場合、設定は不要です。
詳細は [5.3.12 代入値管理](#) を参照してください。

⑩ **作業実行**

Ansible-Legacy コンソールの作業実行の画面から、実行日時、投入オペレーションを選択して設定して処理の実行を指示します。
詳細は [5.3.15 作業実行](#) を参照してください。

⑪ **作業状態確認**

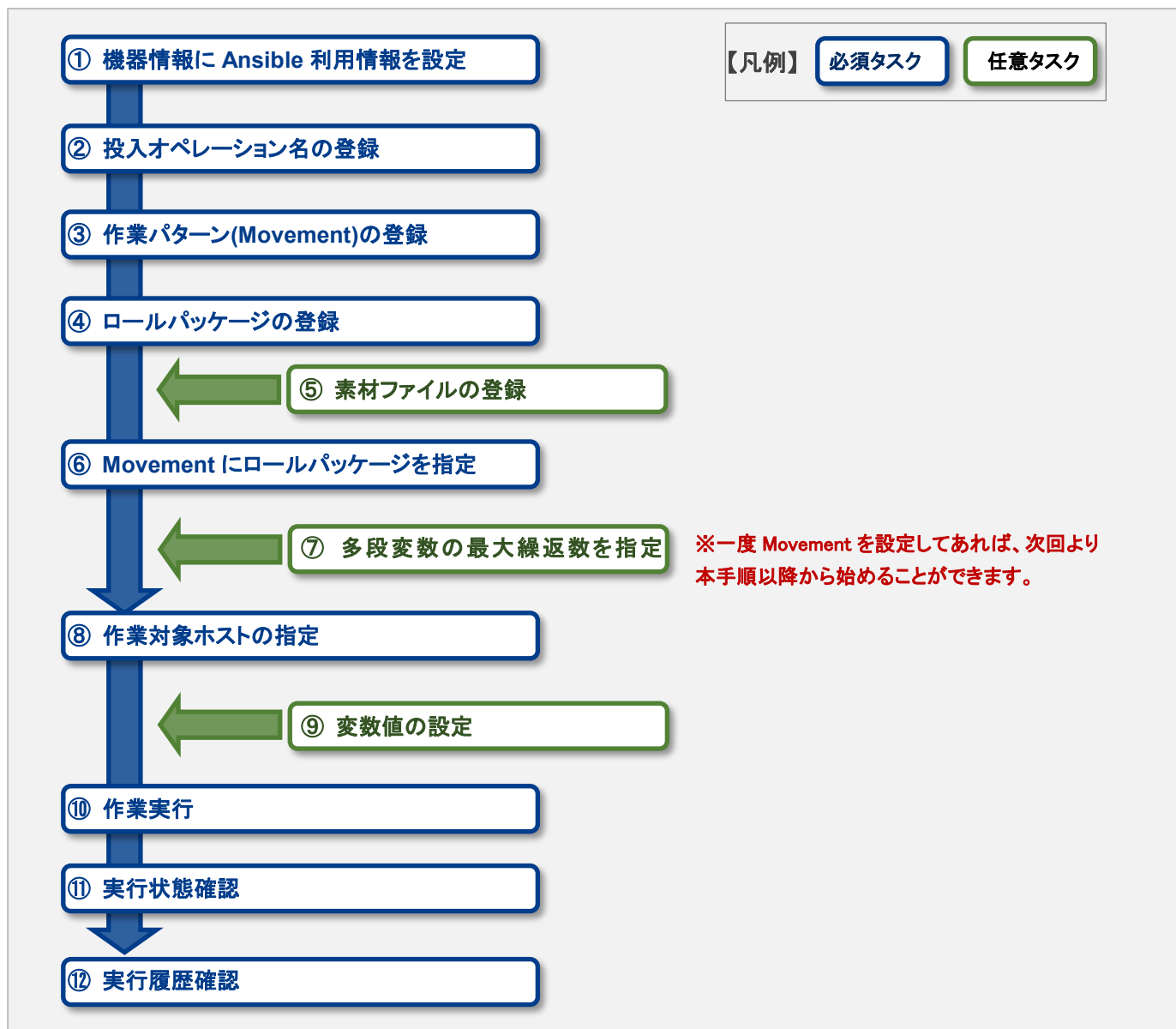
Ansible-Legacy コンソールの作業状態確認の画面では、実行した作業の状態がリアルタイムで表示されます。また、作業の緊急停止や、実行ログ、エラーログを監視することができます。
詳細は [5.3.13 作業状態確認](#) を参照してください。

⑫ **作業履歴確認**

Ansible-Legacy コンソールの作業管理の画面では、実行した作業の一覧が表示され履歴が確認できます。
詳細は [5.3.14 作業管理](#) を参照してください。

4.1.2 Ansible-Legacy Role 作業フロー

以下は、Ansible-Legacy Role で作業を実行するまでの流れです。



● 作業フロー詳細と参照先

① 機器情報に Ansible 利用情報を設定

astroll 基本コンソールの機器一覧の画面から、各機器に対して Ansible 利用情報を設定します。
詳細は [5.1.1 機器一覧](#) を参照してください。

② 投入オペレーション名の登録

astroll 基本コンソールの投入オペレーション一覧の画面から、作業用の投入オペレーション名を登録します。
詳細は [5.1.3 投入オペレーション一覧](#) を参照してください。

③ **作業パターン(Movement)の登録**

Ansible-Legacy Role コンソールの Movement 一覧の画面から、作業用の Movement を登録します。
詳細は [5.3.1 Movement 一覧](#) を参照してください。

④ **ロールパッケージの登録**

Ansible-Legacy Role コンソールのロールパッケージ管理の画面から、作業で使用するロールパッケージを登録します。
詳細は [5.3.3 ロールパッケージ管理 \(Ansible-Legacy Role のみ\)](#) を参照してください。

⑤ **素材ファイルの登録 (必要に応じて実施)**

Ansible-Legacy Role コンソールのファイル管理の画面から、作業対象サーバに配置するファイルを登録します。
詳細は [5.3.8 ファイル管理](#) を参照してください。

⑥ **Movement にロールパッケージを指定**

Ansible-Legacy Role コンソールの Movement 詳細の画面から、登録した Movement にプレイブック素材を指定します。
詳細は [5.3.6 Movement 詳細](#) を参照してください。

⑦ **多段変数の最大繰返数を指定**

Ansible-Legacy Role コンソールの多段変数最大繰返管理の画面から、多段変数で配列定義されているメンバー変数の配列の最大繰返数を指定します。
詳細は [5.3.9 多段変数最大繰返数管理 \(Ansible-Legacy Role のみ\)](#) を参照してください。

⑧ **作業対象ホストの指定**

Ansible-Legacy Role コンソールの作業対象ホストの画面から、作業対象ホストを指定します。
詳細は [5.3.11 作業対象ホスト](#) を参照してください。

⑨ **変数値の設定**

Ansible-Legacy Role コンソールの代入値管理の画面から、Movement に登録した Playbook 内で定義した変数の値を設定します。変数を利用していない場合、設定は不要です。
詳細は [5.3.12 代入値管理](#) を参照してください。

⑩ **作業実行**

Ansible-Legacy Role コンソールの作業実行の画面から、実行日時、投入オペレーションを選択して設定して処理の実行を指示します。
詳細は [5.3.15 作業実行](#) を参照してください。

⑪ **作業状態確認**

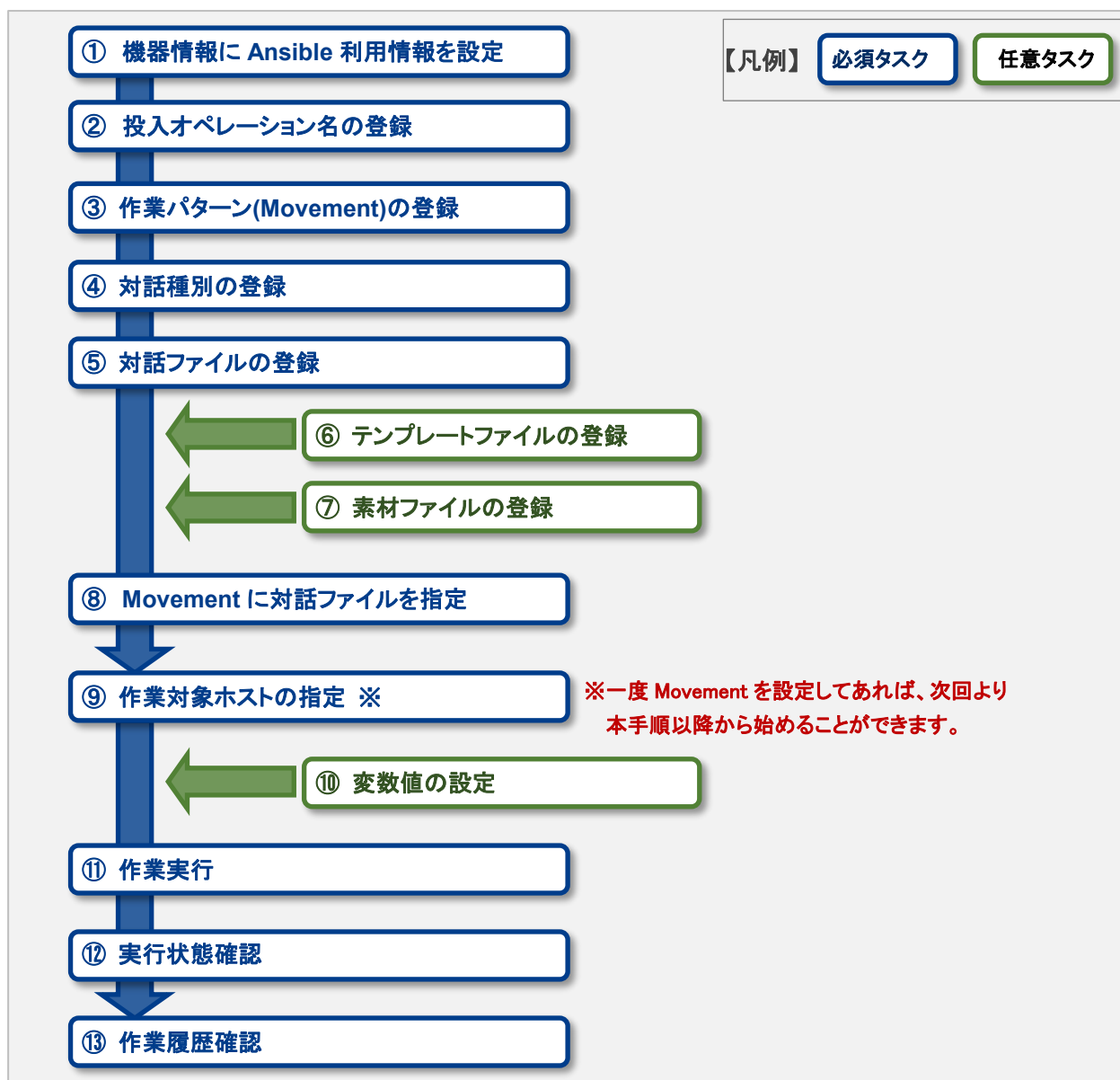
Ansible-Legacy Role コンソールの作業状態確認の画面から、実行した作業の状態がリアルタイムで表示されます。また、作業の緊急停止や、実行ログ、エラーログを監視することができます。
詳細は [5.3.13 作業状態確認](#) を参照してください。

⑫ **作業履歴確認**

Ansible-Legacy Role コンソールの作業管理の画面から、実行した作業の一覧が表示され履歴が確認できます。
詳細は [5.3.14 作業管理](#) を参照してください。

4.1.3 Ansible-Pioneer 作業フロー

以下は、Ansible-Pioneer で作業を実行するまでの流れです。



● 作業フロー詳細と参照先

① 機器情報に Ansible 利用情報を設定

astroll 基本コンソールの機器一覧の画面から、各機器に対して Ansible 利用情報を設定します。
詳細は [5.1.1 機器一覧](#) を参照してください。

② 投入オペレーション名の登録

astroll 基本コンソールの投入オペレーション一覧の画面から、作業用の投入オペレーション名を登録します。
詳細は [5.1.3 投入オペレーション一覧](#) を参照してください。

③ **作業パターン(Movement)の登録**

Ansible-Pioneer コンソールの Movement 一覧の画面から、作業用の Movement を登録します。
詳細は [5.3.1 Movement 一覧](#) を参照してください。

④ **対話種別の登録**

Ansible-Pioneer コンソールの対話種別リストの画面から、対話種別を登録します。
OS 種別ごとの差異を対話ファイルごとに定義し、同一目的の対話ファイルを対話種別として纏めて機器差分を吸収(抽象化)します
詳細は [5.3.4 対話種別リスト\(Ansible-Pioneer のみ\)](#) を参照してください。

⑤ **対話ファイルの登録**

Ansible-Pioneer コンソールの対話ファイル素材集の画面から、対話種別と OS 種別の組み合わせに対して対話ファイルを登録します。
詳細は [5.3.5 対話ファイル素材集\(Ansible-Pioneer のみ\)](#) を参照してください。

⑥ **テンプレートファイルの登録(必要に応じて実施)**

Ansible-Pioneer コンソールのテンプレート管理の画面から、対話ファイル内で定義している template モジュールの template ファイルと template 埋め込み変数の登録／更新／廃止を行います。
詳細は [5.3.7 テンプレート管理\(Ansible-Legacy/Ansible-Pioneer\)](#) を参照してください。

⑦ **素材ファイルの登録(必要に応じて実施)**

Ansible-Pioneer コンソールのファイル管理の画面から、作業対象サーバに配置するファイルを登録します。
詳細は [5.3.8 ファイル管理](#) を参照してください。

⑧ **Movement に対話ファイルを指定**

Ansible-Pioneer コンソールの Movement 詳細の画面から、登録した Movement に対話ファイルに対応した対話種別リストを指定します。
詳細は [5.3.6 Movement 詳細](#) を参照してください。

⑨ **作業対象ホストの指定**

Ansible-Pioneer コンソールの作業対象ホストの画面から、作業対象ホストを指定します。
詳細は [5.3.11 作業対象ホスト](#) を参照してください。

⑩ **変数値の設定**

Ansible-Pioneer コンソールの代入値管理の画面から、Movement に登録した Playbook 内で定義した変数の値を設定します。変数を利用していない場合、設定は不要です。
詳細は [5.3.12 代入値管理](#) を参照してください。

⑪ **作業実行**

Ansible-Pioneer コンソールの作業実行の画面から、実行日時、投入オペレーションを選択して設定して処理の実行を指示します。
詳細は [5.3.15 作業実行](#) を参照してください。

⑫ **作業状態確認**

Ansible-Pioneer コンソールの作業状態確認の画面では、実行した作業の状態がリアルタイムで表示されます。また、作業の緊急停止や、実行ログ、エラーログを監視することができます。
詳細は [5.3.13 作業状態確認](#) を参照してください。

⑬ 作業履歴確認

Ansible-Pioneer コンソールの作業管理の画面では、実行した作業の一覧が表示され履歴が確認できます。

詳細は [5.3.14 作業管理](#) を参照してください。

■登録画面項目一覧凡例

次項に記載の登録画面項目一覧表の内容について説明します。

① 項目	② 説明	③ 入力 必須	④ 入力形式	⑤ 制約事項

①項目

- ・サブメニュー内の項目名です

②説明

- ・項目に対する説明です

③入力必須

- ・○: 項目に対する内容の入力が必須の項目
- ・- : 項目に対する内容の入力が任意の項目

④入力形式

- ・手動入力: 手動での入力が必要な項目
- ・自動入力: 自動で内容が入力される項目
- ・チェックボックス: チェックボックス形式の項目
- ・ボタン: ラジオボタン形式の項目
- ・リスト選択: リストボックス形式の項目

⑤制約事項

- ・項目に対する制約事項(文字数制限など)です

5 Ansible driver 機能・操作方法説明

本章では、Ansible driver で利用する各コンソールの機能について説明します。

5.1 基本コンソール

本節では、astroll 基本コンソールでの操作について記載します。

本作業は astroll 基本コンソールマニュアルを参照して、astroll 基本コンソール画面内で作業を実施してください。

5.1.1 機器一覧

(1) [機器一覧]では、作業対象ホストの情報を登録／更新／廃止を行います。

本書では、Ansible driver の動作に必要なとなる機器一覧の項目(赤枠)について説明します。

他の項目についての説明は、「利用手順マニュアル_astroll_基本コンソール」を参照してください。

The screenshot shows the 'astroll 基本コンソール' (Basic Console) interface. On the left, a sidebar menu lists various functions, with '機器一覧' (Device List) highlighted by a red rectangle. The main area displays a table for managing hosts. The table has columns for '停止' (Stop), '管理システム障害' (Management System Failure), 'HW機器識別' (HW Device Identification), 'ホスト名' (Host Name), 'IPアドレス' (IP Address), 'EtherWakeOnLan' (containing 'MACアドレス' and 'ネットワークデバイス名'), and 'ログインユーザID'. Below the table, there are search filters for each column, a 'フィルタ' (Filter) button, a 'フィルタクリア' (Filter Clear) button, and a checked 'オートフィルタ' (Auto Filter) checkbox. At the bottom of the main area, there are four buttons: '一覧/更新' (List/Update), '登録' (Registration), '全件ダウンロードとファイルアップロード編集' (Full Download and File Upload Edit), and '変更履歴' (Change History).

図 5.1-1 サブメニュー画面(機器一覧)

(2) 「登録」-「登録開始」ボタンより、機器情報の登録を行います。

図 5.1-2 登録画面(機器一覧 - 共通項目)

図 5.1-3 登録画面(機器一覧 - Ansible 利用情報)

(3) 登録画面の共通項目一覧は以下のとおりです。
Web 画面のカラム名の後ろに赤のアスタリスク(*)が付いているカラムが必須入力になりますが、Ansible driverを利用する場合には、Ansible 利用情報も必須入力になります。
未入力で作業実行した場合、想定外エラーとなる場合があります。

表 5.1-1 登録画面項目一覧(機器一覧)

項目		説明	入力必須	入力形式	制約事項
管理システム項番		登録情報を識別する一意のIDが自動入力されます	-	自動入力	-
ホスト名		ホスト名を記入します	○	手動入力	最大長 128 バイト
IP アドレス		IP アドレス(xxx.xxx.xxx.xxx 形式)を記入します	○	手動入力	最大長 15 バイト
EtherWake OnLan	MAC アドレス	MAC アドレスを記入します	-	手動入力	最大長 17 バイト
	ネットワークデバイス名	ネットワークデバイス名を記入します	-	手動入力	最大長 32 バイト
ログインユーザ ID		ログインユーザ ID を記入します	○	手動入力	最大長 30 バイト
ログインパスワード	管理	astroll でパスワードを管理する場合「●」を選択します	○	リスト選択	-
	ログインパスワード	パスワードを指定します	○	手動入力	最大長 30 バイト
ssh 認証鍵ファイル		ssh 認証鍵ファイルを指定して鍵認証する場合のファイルを入力します。 認証方式が鍵方式の場合で ssh 認証鍵ファイルを指定する場合に必須入力となります。	-	ファイル選択	最大サイズ 10K バイト

項目		説明	入力必須	入力形式	制約事項
Legacy/Role 利用情報	認証方式	Ansible からホストへ接続する際の認証方式を選択します。 ・パスワード方式の場合、ログインパスワードの管理は、「●」である必要があります。 ・鍵方式の場合、ログインユーザでの sudo 権限を /etc/sudoers に設定しておく必要があります。また、公開鍵を交換しておく必要があります。	○	リスト選択	説明欄記載のとおり
	WinRM 接続情報	ポート番号	-	手動入力	説明欄記載のとおり
		サーバ証明書	-	ファイル選択	最大サイズ 10K バイト
Pioneer 利用情報	プロトコル	対象機器にログインする際のプロトコル(ssh/telnet)を選択します。	○	リスト選択	-
	OS 種別	対象機器の OS を選択します。 OS 種別マスタで登録されている OS 種別がリスト表示されます。	○	リスト選択	-
接続オプション		(ssh 接続の場合) /etc/ansible.cfg/ssh_args に設定している ssh オプション以外のオプションを設定したい場合、設定したいオプションを入力します。 (telnet 接続の場合) telnet 接続時のオプションを設定したい場合、設定したいオプションを入力します。	-	手動入力	最大長 512 バイト
インベントリファイル追加オプション		astroll が設定しないインベントリファイルのオプションを設定したい場合に追加するオプションを入力します (Exp) Python のバージョンが 2.7 以降で https のサーバ証明書の検証を行わない場合。 ansible_winrm_server_cert_validation=ignore	-	手動入力	最大長 512 バイト
備考		自由記述欄です。	-	手動入力	最大長 4000 バイト

5.1.2 紐付対象メニュー

- (1) [紐付対象メニュー]では、代入値自動登録設定で連携する CMDB のメニューを登録／更新／廃止を行います。

※CMDB をカスタマイズしたときに、連携できるオプションの機能です。デフォルトでは利用しません。

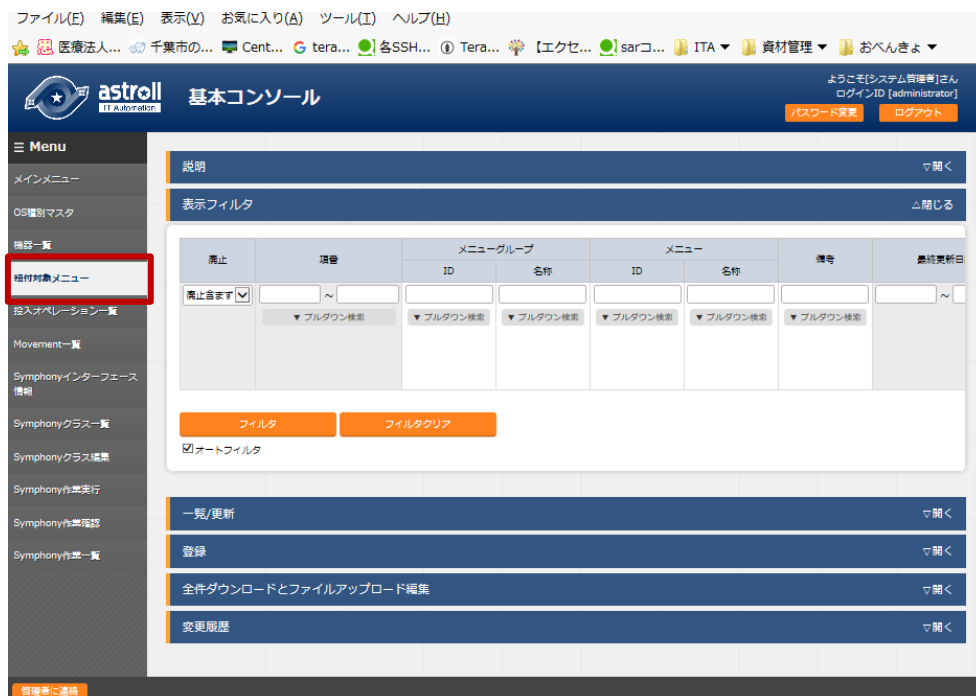


図 5.1-4 サブメニュー画面(紐付対象メニュー)

登録方法の詳細は、関連マニュアルの『利用手順マニュアル_astroll_基本コンソール』をご参照下さい。

5.1.3 投入オペレーション一覧

- (1) [投入オペレーション一覧]画面では、オーケストレータで実行する作業対象ホストに対するオペレーションを管理します。作業は astroll 基本コンソール内メニューより選択します。

The screenshot displays the 'astroll 基本コンソール' interface. On the left, a vertical menu lists various functions, with '投入オペレーション一覧' (Input Operation List) highlighted by a red rectangular box. The main content area is titled '投入オペレーション一覧' and contains a table for managing operations. The table has the following columns: 'No.', 'オペレーション名', '実行予定日', 'オペレーションID', '最終実行日', '備考', '最終更新日時', and '最終更新者'. Below the table, there are two orange buttons labeled 'フィルタ' and 'フィルタのリセット', and a checkbox labeled 'オートフィルタ'. At the bottom of the page, there are four links: '一覧/更新', '登録', '全件ダウンロードとファイルアップロード機能', and '変更履歴'.

図 5.1-5 サブメニュー画面(投入オペレーション一覧)

登録方法の詳細は、関連マニュアルの『利用手順マニュアル_astroll_基本コンソール』をご参照下さい。

5.2 Ansible 共通コンソール

本節では、Ansible 共通コンソールでの操作について記載します。

5.2.1 インタフェース情報

- (1) [インタフェース情報]では、astroll システム・Ansible driver サーバと Ansible サーバが共有するディレクトリのパスおよび Ansible サーバへの接続インタフェース情報を登録／更新／廃止を行います。
※なお、インストール時に必要な値が設定されるため、基本的には設定は必要ありません。



図 5.2-1 サブメニュー画面(インタフェース情報)

- (2) 「一覧」-「更新」ボタンより、オペレーション情報の登録を行います。

更新	データリレイストレージ/パス(ITA)	データリレイストレージ/パス(ANS)	Symphonyインスタンスデータリレイストレージ/パス(ANS)	プロトコル	ホスト	ポート	オプション/パラメータ	Access_key_id
1	/astroll/data_re	/astroll/data_re	/astroll/data_re	https	astroll-it-autom	443	-vvvv	/AccessKeyId

図 5.2-2 登録画面(インタフェース情報)

- (3) インタフェース情報画面の項目一覧は以下のとおりです。
インタフェース情報が未登録または、複数レコード登録されている状態で作業実行した場合、**作業実行は想定外エラーとなります。**

表 5.2-1 登録画面項目一覧(インタフェース情報)

項目	説明	入力必須	入力形式	制約事項
データリレイストレージパス(astroll)※1	astroll システム・Ansible driver サーバから見たディレクトリを入力します。	○	手動入力	最大長 128 バイト
データリレイストレー	Ansible RestAPI サーバから見たディレクトリを入力し	○	手動入力	最大長 128 バイト

項目	説明	入力 必須	入力形式	制約事項
ジパス(ANS)	ます。			
Symphony インスタ ンス データリレイストレー ジパス(ANS)	Symphony 実行時、各 Movement で共有するディレクトリを、Ansible RestAPI サーバで共有するディレクトリで入力します。 astroll システムから見たパスは Symphony インターフェース情報より設定します。Symphony インターフェース情報については「利用手順マニュアル astroll 基本コンソール」を参照して下さい。	○	手動入力	最大長 128 バイト
プロトコル	http / https のどちらかを入力します。	○	手動入力	-
ホスト	Ansible サーバのホスト名(または IP アドレス)を入力します。 HTTPS 通信の場合はホスト名が推奨です。	○	手動入力	最大長 128 バイト
ポート	Ansible サーバの接続ポート(80/443)を入力します。通常は HTTPS(443)です。	○	手動入力	-
オプションパラメータ	ansible-playbook コマンドのオプションパラメータを入力します。デフォルトは -vvvv が入力されています。 -i オプションは astroll が設定します。 -f オプションは「 5.4.6 Movement 一覧 」の並列実行数での設定を推奨します。 --check オプションは「 6.2.1 作業実行 」のドライランボタンの使用を推奨します。	-	手動入力	最大長 512 バイト
ACCESS_KEY_ID	Ansible サーバ接続時の認証に使用するアクセスキーを入力します。	○	手動入力	最大長 64 バイト
SECRET_ACCESS_KEY	Ansible サーバ接続時の認証に使用するシークレットアクセスキーを入力します。	-	手動入力	最大長 64 バイト
状態監視周期(単位 ミリ秒)	「 5.3.13 作業状態確認 」で表示されるログのリフレッシュ間隔を入力します。通常は 3000 ミリ秒程度が推奨値です。	○	手動入力	最小値 1000 ミリ秒
進行状態表示行数	「 5.3.13 作業状態確認 」での進行ログ・エラーログの最大表示行数を入力します。通常は 1000 行程度が推奨値です。	○	手動入力	-
NULL 連携	代入値自動登録設定でパラメータシートの具体値が NULL(空白)の場合に、代入値管理への登録を NULL(空白)の値で行うか設定します。 代入値自動登録設定メニューの「NULL 連携」が空白の場合この値が適用されます。 ・「有効」の場合、パラメータシートの値がどのような値でも代入値管理への登録が行われます。 ・「無効」の場合、パラメータシートに値が入っている場合のみ代入値管理への登録が行われます。	○	リスト選択	-
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

※1 データリレイストレージパスは、それぞれ異なるサーバで運用される場合、ディレクトリパス名が異なる可能性があるため、別々に管理します。詳細は「システム構成／環境構築ガイド(Ansible driver 編)」を参照してください。

5.2.2 グローバル変数管理

- (1) [グローバル変数管理]では、Playbook や対話ファイルなどで利用するグローバル変数名を登録／更新／廃止を行います。

The screenshot shows the 'astroll Ansible共通' interface. The left sidebar has a 'Menu' section with 'グローバル変数管理' highlighted. The main content area is titled 'グローバル変数管理' and contains a table with the following columns: 廃止, 項目, グローバル変数名, 属性値, 変数名説明, 備考, 最終更新日時, and 最終更新者. Below the table are buttons for 'フィルタ' and 'フィルタクリア', and a checkbox for 'オートフィルタ'. On the right, there are links for '説明', '表示フィルタ', '一覧/更新', '登録', '全件ダウンロードとファイルアップロード編集', and '変更履歴'.

図 5.2-3 サブメニュー画面(グローバル変数管理)

- (2) 「登録」-「登録開始」ボタンより、オペレーション情報の登録を行います。

The screenshot shows the '登録' (Registration) form. It has the following fields: 項目, グローバル変数名* (with a red asterisk indicating it is required), 属性値* (with a red asterisk indicating it is required), 変数名説明, 備考, 最終更新日時, and 最終更新者. There are '戻る' (Back) and '登録' (Register) buttons at the bottom. A note states '※*は必須項目です。' (Note: * indicates required items).

図 5.2-4 登録画面(グローバル変数管理)

(3) グローバル変数管理画面の項目一覧は以下のとおりです。

表 5.2-2 登録画面項目一覧(グローバル変数管理)

項目	説明	入力必須	入力形式	制約事項
グローバル変数名	変数名を入力します。 変数名は、「GBL_****」形式で入力します。 ****: 半角英数字とアンダースコア(_)が利用可能です。(最小値:1 バイト、最大値:128 バイト)	○	手動入力	説明欄記載のとおり
具体値	具体値を入力します。 具体値にファイル埋込変数「CPF_」とテンプレート埋込変数「TPF_」が入力出来ます。変数を記述する場合、Playbook に変数を記述する場合と同様、変数名を{{ }}で囲みます。 入力例) 具体値に TPF_sample を入力する場合 '{{△TPF_sample△}}' △:半角スペース ':任意 各モードで具体値にファイル埋込変数とテンプレート埋込変数が入力されている場合の扱い。 ・Ansible-Legacy ファイル埋込変数とテンプレート埋込変数が各変数として扱われます。 ・Ansible-Legacy-Role ファイル埋込変数のみを変数として扱われます。 テンプレート埋込変数は単なる具体値として扱われます。 ・Ansible-pioneer 単なる具体値として扱われます。	○	手動入力	最大長 1024 バイト
変数名説明	変数に対する説明・コメントを入力します	-	手動入力	最大長 128 バイト
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

5.3 Ansibel-Legacy／Legacy Role／Pioneer コンソール

Ansibel-Legacy／Legacy Role／Pioneer コンソールの操作です。

5.3.1 Movement 一覧

(1) [Movement 一覧]では Movement 名の登録／更新／廃止を行います。

The screenshot shows the 'Ansibel-Legacy' web interface. On the left sidebar, the 'Movement一覧' menu item is highlighted with a red rectangle. The main content area is divided into two sections: '説明' (Description) and '一覧/更新' (List/Update). The '説明' section contains a '表示フィルタ' (Display Filter) table with columns for '廃止' (Delete), 'MovementID', 'Movement名', 'オークストレータ', '通信タイマー', 'ホスト指定形式', and 'Ansible利用情報'. Below this is a 'フィルタ' (Filter) button and a 'フィルタクリア' (Clear Filter) button. The '一覧/更新' section contains a table with columns for '更新' (Update), '廃止' (Delete), 'MovementID', 'Movement名', 'オークストレータ', '通信タイマー', 'ホスト指定形式', 'Ansible利用情報', '実行' (Execute), '変更' (Change), and '最終' (Final). The table has one row with the following data: 1, LegacyMove, Ansible Legacy, IP, 実行, 1, 2019/01/01. Below the table is a 'フィルタ結果件数: 1' (Filter result count: 1) label and an 'Excel出力' (Excel output) button. At the bottom of the page, there are three buttons: '登録' (Register), '全件ダウンロードとファイルアップロード編集' (Download all and edit file upload), and '変更履歴' (Change history).

図 5.3-1 サブメニュー画面(Movement 一覧)

登録方法の詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

5.3.2 プレイブック素材集 (Ansible-Legacy のみ)

- (1) [プレイブック素材集]ではユーザーが作成した Playbook の登録／更新／廃止を行います。
※本メニューは Ansible-Legacy コンソールにのみ存在します。
Playbook の記述など関しては、「[6.1 Playbook \(Ansible-Legacy\) の記述](#)」を参照してください。

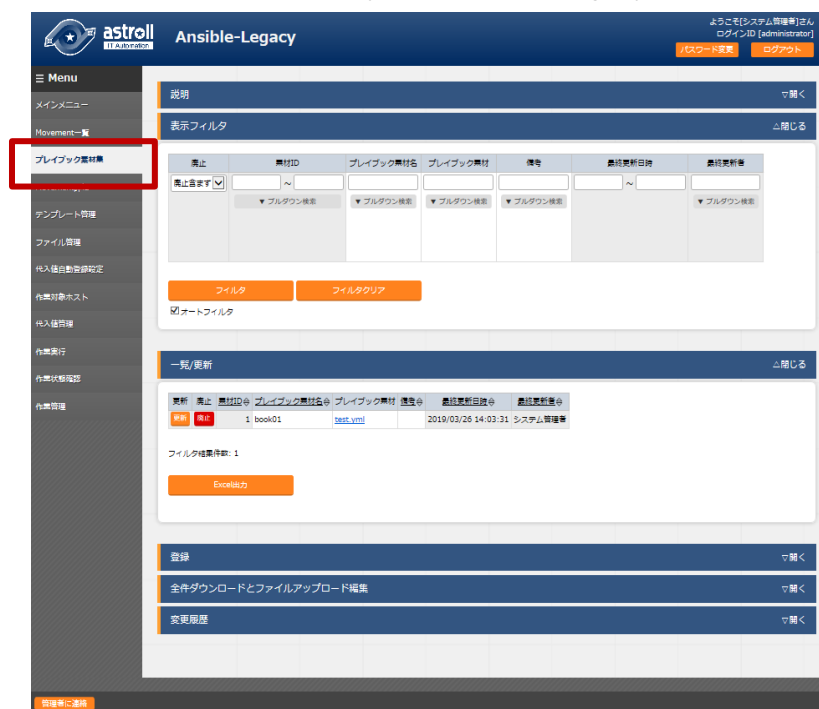


図 5.3-2 サブメニュー画面(プレイブック素材集)

- (2) 「登録」-「登録開始」ボタンより、Playbook の登録を行います。

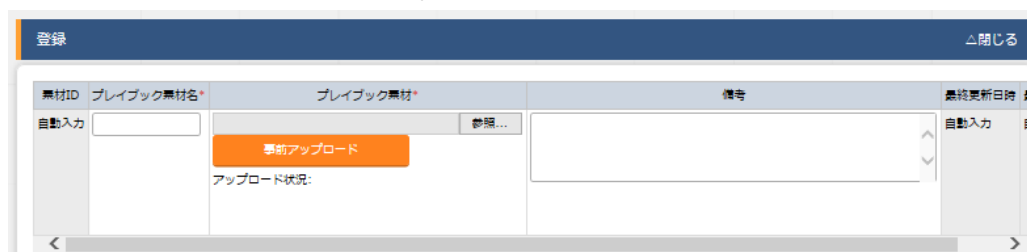


図 5.3-3 登録画面(プレイブック素材集)

(3) 登録画面の項目一覧は以下のとおりです。

表 5.3-1 登録画面項目一覧(プレイブック素材集)

項目	説明	入力 必須	入力形式	制約事項
プレイブック素材名	astroll で管理するプレイブック素材名を入力します。	○	手動入力	最大長 32 バイト
プレイブック素材	作成した Playbook ファイルをアップロードします。	○	ファイル選択	最大サイズ 20M バイト
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

「登録」の前に、「プレイブック素材」を「事前アップロード(①)」してください。「アップロード状況(②)」に Playbook のファイル名が表示されたのを確認してから、「登録」ボタンを押してください。



内部の処理で Playbook ファイル内に定義している変数を抽出します。抽出した変数は、「[5.3.12 代入値管理](#) 代入値自動登録設定」で具体値の登録が可能になります。
抽出するタイミングはリアルタイムではありませんので、「[5.3.10 代入値自動登録設定](#)」や「[5.3.12 代入値管理](#)」で変数が扱えるまでに時間がかかる※4 場合があります。

※4 「本書: 7.2 起動周期の変更」の起動周期に依存します。

5.3.3 ロールパッケージ管理 (Ansible-Legacy Role のみ)

(4) ユーザーが作成したロールパッケージファイルの登録／更新／廃止を行います。

※本メニューは Ansible-Legacy Role コンソールにのみ存在します。

ロールパッケージファイルは、「roles」のある階層のディレクトリを zip にて圧縮したものを登録してください。ロールパッケージディレクトリ構成などは「[6.3 ロールパッケージ \(Ansible-Legacy Role\) の記述](#)」を参照してください。

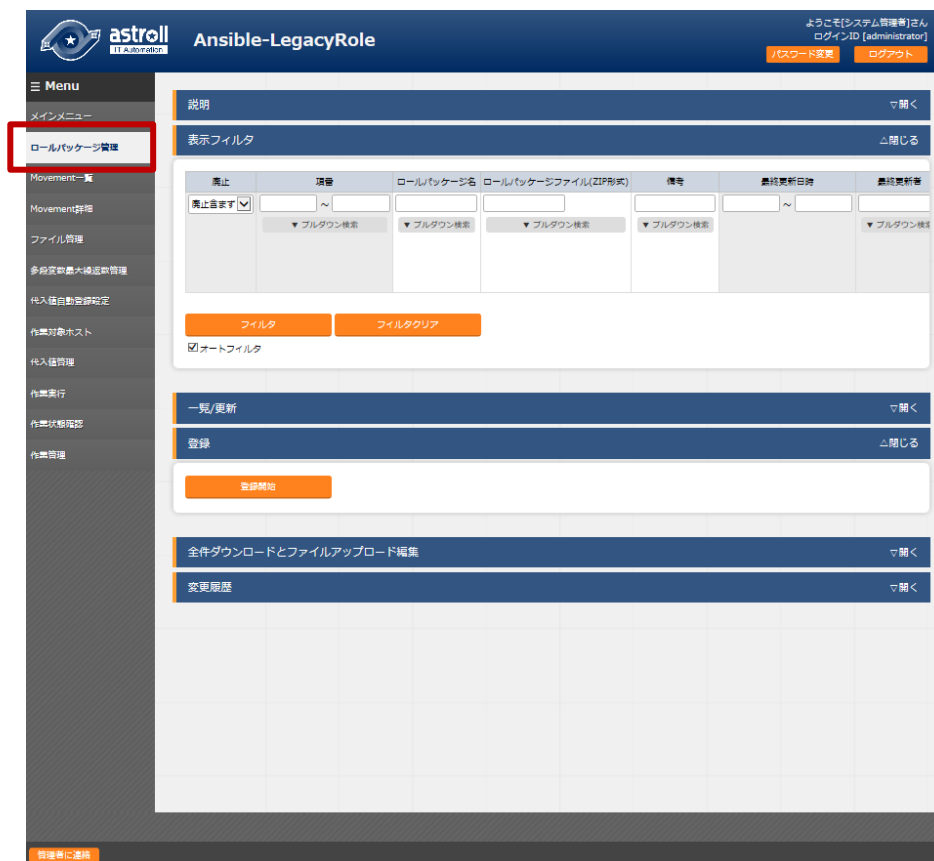


図 5.3-4 サブメニュー画面 (ロールパッケージ管理)

登録方法の詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

5.3.4 対話種別リスト(Ansible-Pioneer のみ)

- (1) [対話種別リスト]では、対話種別の登録／更新／廃止を行います。
 ※本メニューは Ansible-Pioneer コンソールにのみ存在します。
 Ansible-Pioneer では、「OS 種別」ごとの差異を対話ファイルごとに定義し、同一目的の対話ファイルを「対話種別」として纏めて機器差分を吸収(抽象化)します。

図 5.3-5 サブメニュー画面(対話種別リスト)

- (2) 「登録」-「登録開始」ボタンより、オペレーション情報の登録を行います。

図 5.3-6 登録画面(対話種別リスト)

- (3) 登録画面の項目一覧は以下のとおりです。

表 5.3-2 登録画面項目一覧(対話種別リスト)

項目	説明	入力必須	入力形式	制約事項
対話種別名	対話種別名を入力します	○	リスト選択	最大長 32 バイト
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

5.3.5 対話ファイル素材集(Ansible-Pioneer のみ)

- (1) [対話ファイル素材集]では、ユーザーが作成した対話ファイルの登録／更新／廃止を行います。
※本メニューは、Ansible-Pioneer コンソールにのみ存在します。
対話ファイルの記述などに関しては、「5.3.4 対話種別リスト(Ansible-Pioneer のみ)」を参照してください。
「対話種別」と「OS 種別」の組み合わせごとに対話ファイルを登録します。
1つの「対話種別」で複数の OS に対応させたい場合は、同じ「対話種別」で、「OS 種別」それぞれについて対話ファイルを登録してください。

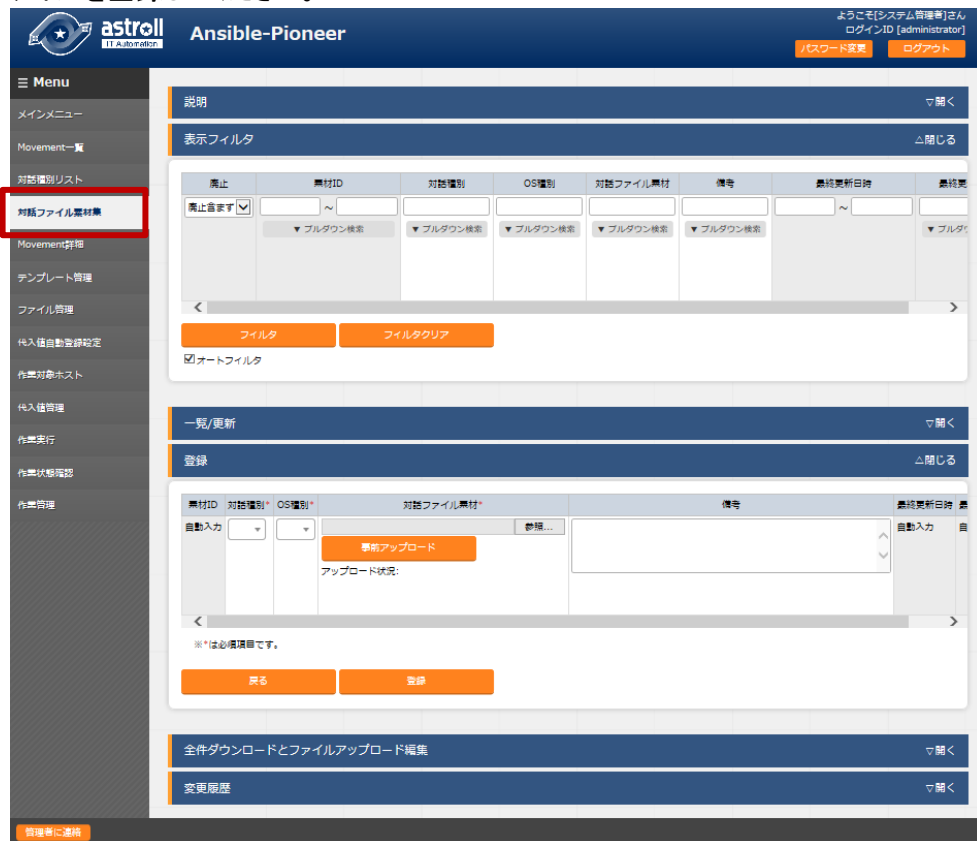


図 5.3-7 サブメニュー画面(対話ファイル素材集)

- (2) 「登録」-「登録開始」ボタンより、対話ファイル素材の登録を行います。



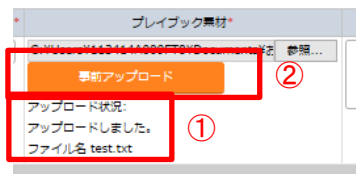
図 5.3-8 登録画面(対話ファイル素材集)

(3) 登録画面の項目一覧は以下のとおりです。

表 5.3-3 登録画面項目一覧(対話ファイル素材集)

項目	説明	入力 必須	入力形式	制約事項
対話種別	対話種別リストに登録されている対話種別が表示されます。 登録する対話ファイルの対話種別を選択します。	○	リスト選択	-
OS 種別	OS 種別マスタに登録されている OS 種別が表示されます。 登録する対話ファイルの OS 種別を選択します。	○	リスト選択	-
対話ファイル素材	対話種別と OS 種別に対応する対話ファイルをアップロードします。	○	ファイル 登録	最大サイズ 20M バイト
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

「登録」の前に、「対話ファイル」を「事前アップロード(①)」してください。「アップロード状況(②)」に対話ファイルのファイル名が表示されたのを確認してから、「登録」ボタンを押してください。



内部の処理で対話ファイル内に定義している変数を取り出します。取り出した変数は、「[5.3.10 代入値自動登録設定](#)」や「[5.3.12 代入値管理](#)」で具体値の登録が可能になります。

抜出するタイミングはリアルタイムではないので、「[5.3.10 代入値自動登録設定](#)」や「[5.3.12 代入値管理](#)」で変数が扱えるまでに時間がかかる※6場合があります。

※6 「本書：7.2 起動周期の変更」の起動周期に依存します。

5.3.6 Movement 詳細

(1) [Movement 詳細]では、Movement で実行する素材の登録／更新／廃止を行います。

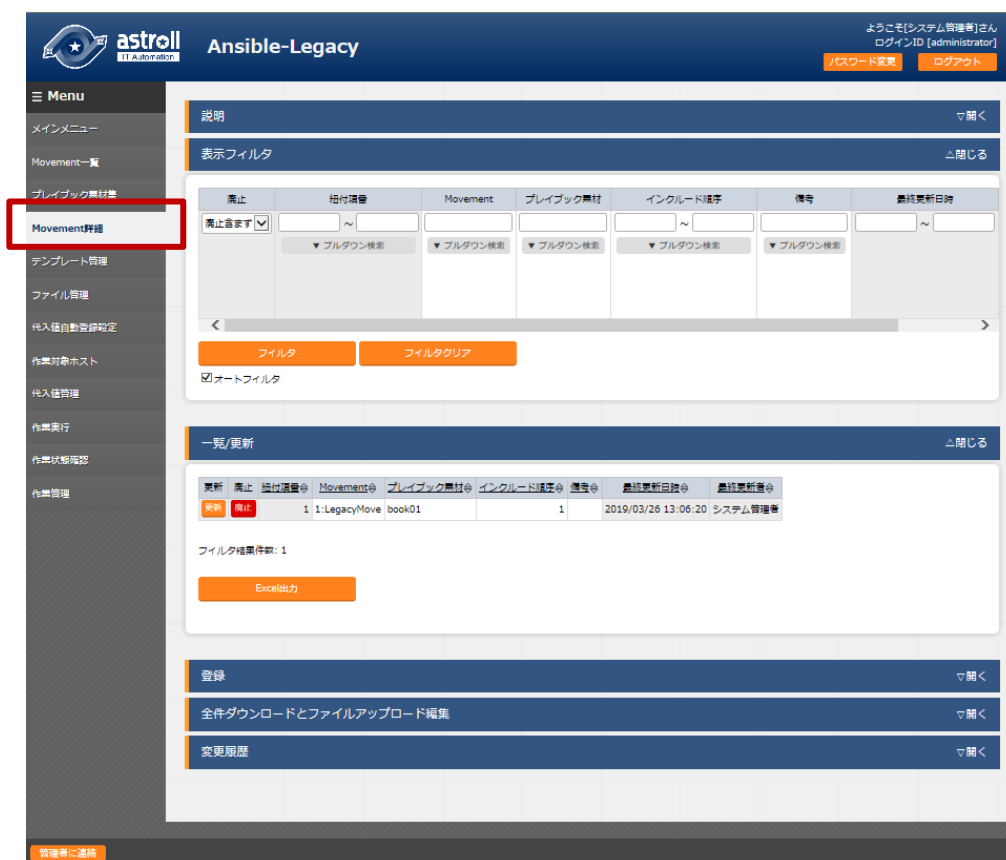


図 5.3-9 サブメニュー画面(Movement 詳細)

※画面は Ansible-Legacy のものです。

(2) 「登録」-「登録開始」ボタンより、Movement 詳細の登録を行います。



図 5.3-10 登録画面(Movement 詳細)

(3) 登録画面の項目一覧は以下のとおりです。

- Ansible-Legacy の場合

表 5.3-4 登録画面項目一覧(Movement 詳細 Ansible-Legacy の場合)

項目	説明	入力 必須	入力形式	制約事項
Movement	Movement 一覧で登録した Movement が表示されます。 Movement を選択します。	○	リスト選択	-
プレイブック素材	「5.3.2 プレイブック素材集 (Ansible-Legacy のみ)」で登録したプレイブック素材が表示されます。プレイブック素材を選択します。	○	リスト選択	-
インクルード順序	プレイブック素材の実行順序(1～:一意値)を入力します。 入力されたインクルード順序(昇順)でプレイブック素材が実行されます。	○	手動入力	半角整数
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

- Ansible-Legacy Role の場合

表 5.3-5 登録画面項目一覧(Movement 詳細 Ansible-Legacy の場合)

項目	説明	入力 必須	入力形式	制約事項
Movement	Ansible-Legacy と同様。	○	リスト選択	-
ロールパッケージ名	ロールパッケージ管理で登録したロールパッケージが表示されます。実行するロールパッケージを選択します。	○	リスト選択	-
ロール名	ロールパッケージ名で選択したロールパッケージに含まれているロール名が表示されます。実行するロールパッケージ内のロールを選択します。	○		-
インクルード順序	Ansible-Legacy と同様。	○	手動入力	半角整数
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

【注意】

同一 Movement に複数のロールパッケージを登録しないでください。

複数のロールパッケージが登録されている Movement を作業実行した場合、想定外エラーとなります。

- Ansible-Pioneer の場合

表 5.3-6 登録画面項目一覧(Movement 詳細 Ansible-Pioneer の場合)

項目	説明	入力 必須	入力形式	制約事項
Movement	Ansible-Legacy と同様。	○	リスト選択	-
対話種別	「5.3.4 対話種別リスト (Ansible-Pioneer のみ)」で登録した対話種別が表示されます。実行する対話種別を選択します。 ホスト毎に OS 種別と対話種別に関連付く対話ファイルが実行対象となります。	○	リスト選択	-
インクルード順序	Ansible-Legacy と同様。	○	手動入力	半角整数
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

5.3.7 テンプレート管理(Ansible-Legacy/Ansible-Pioneer)

- (1) [テンプレート管理]では、Playbook で定義している template モジュールや ios_config モジュールなどのパラメータで使用する Jinja2 テンプレートファイルと template 埋め込み変数の登録／更新／廃止を行います。

※本メニューは、Ansible-Legacy と Ansible-Pioneer コンソールに存在します。

Ansible-Legacy と Ansible-Pioneer では、テンプレート管理でテンプレートモジュールを登録しておくことで、Playbook 内で定義している template モジュールで使用する template ファイルを template 埋め込み変数で指定することが出来ます。

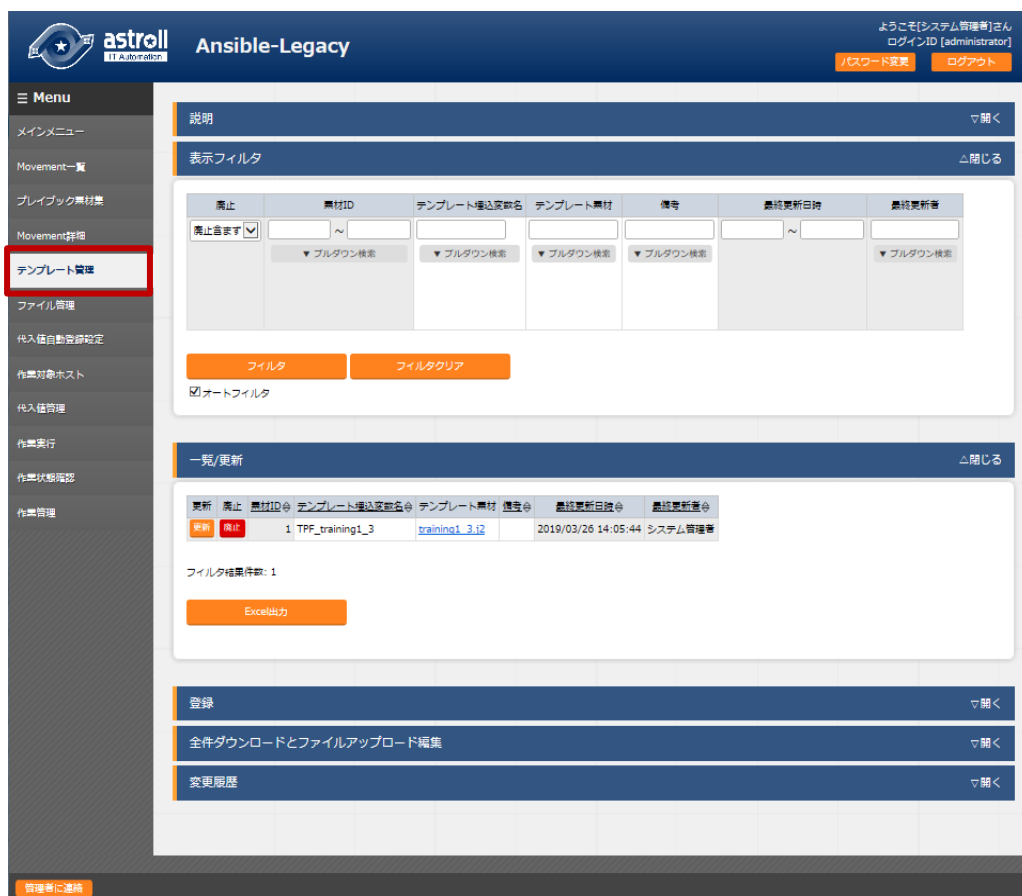


図 5.3-11 サブメニュー画面(テンプレート管理)

登録方法の詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

5.3.8 ファイル管理

- (1) [ファイル管理]では、Playbook 内で定義している各モジュールで使用するファイルとファイル埋め込み変数の登録／更新／廃止を行います。

※本メニューは、Ansible-Legacy/ Ansible-Legacy Role/ Ansible-Pioneer コンソールに存在します。ファイル管理でファイル素材を登録しておくことで、Playbook 内で定義している各モジュールで使用するファイルをファイル埋め込み変数名で指定することが出来ます。

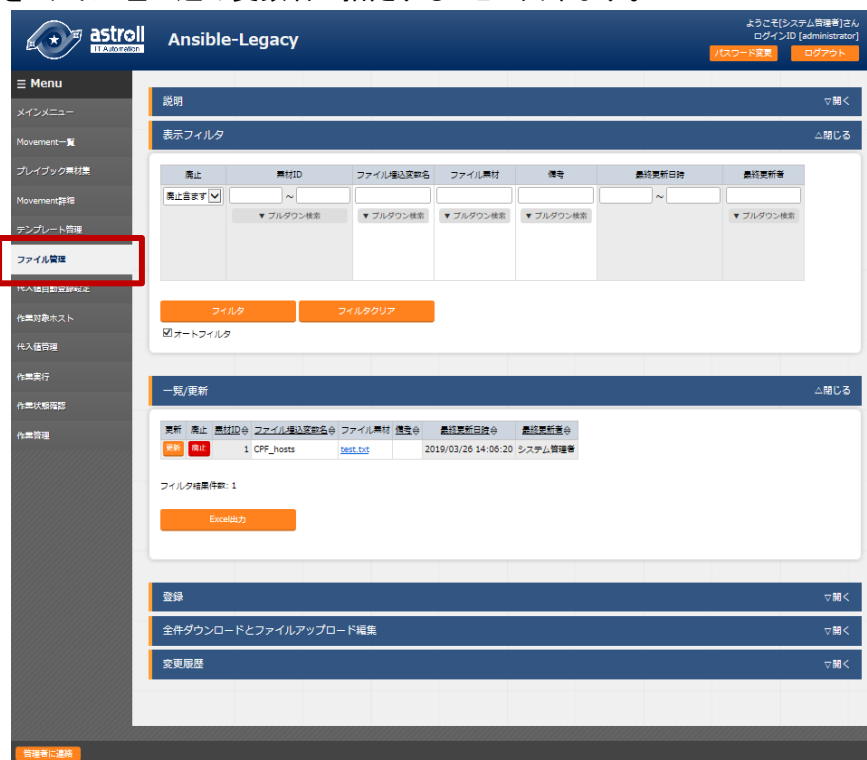


図 5.3-12 サブメニュー画面(ファイル管理)

登録方法の詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

5.3.9 多段変数最大繰返数管理 (Ansible-Legacy Role のみ)

- (2) [多段変数最大繰返数管理]では、「5.3.3 ロールパッケージ管理 (Ansible-Legacy Role のみ)」で登録したロールパッケージで定義されている多段変数内で繰返配列定義されているメンバー変数の配列の最大繰返数の更新が行えます。変更したいメンバー変数の更新ボタンをクリックし最大繰返数を更新します。

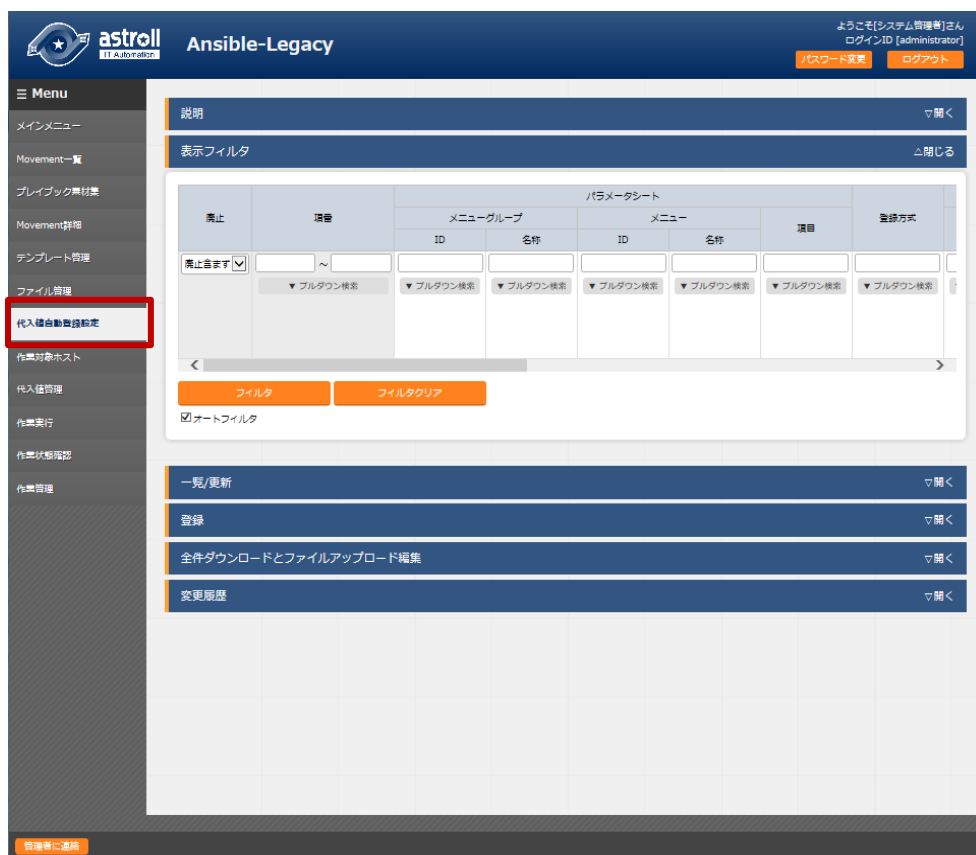


図 5.3-13 サブメニュー画面 (多段変数最大繰返数管理)

登録方法の詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

5.3.10 代入値自動登録設定

- (1) 基本コンソールの「紐付対象メニュー」で連携対象とした CMDB のオペレーションとホスト毎の項目の設定値を紐付ける Movementと変数の登録／更新／廃止を行います。登録した情報は内部の処理により代入値管理と作業対象ホストに反映されます。

※CMDB をカスタマイズしたときに、連携できるオプションの機能。デフォルトでは利用しません。

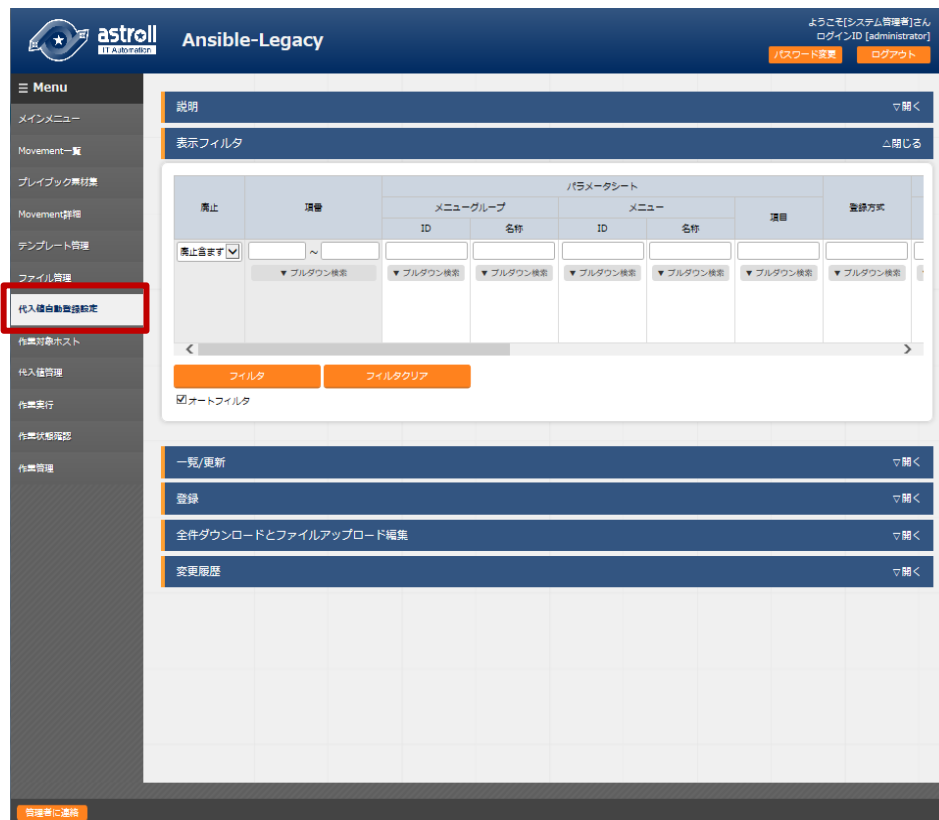


図 5.3-14 サブメニュー画面(代入値自動登録設定)

※画面は Ansible-Legacy Role のものです

登録方法の詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

5.3.11 作業対象ホスト

(1) [作業対象ホスト]では、オペレーションに関連付く Movement とホストの登録／更新／廃止を行います。

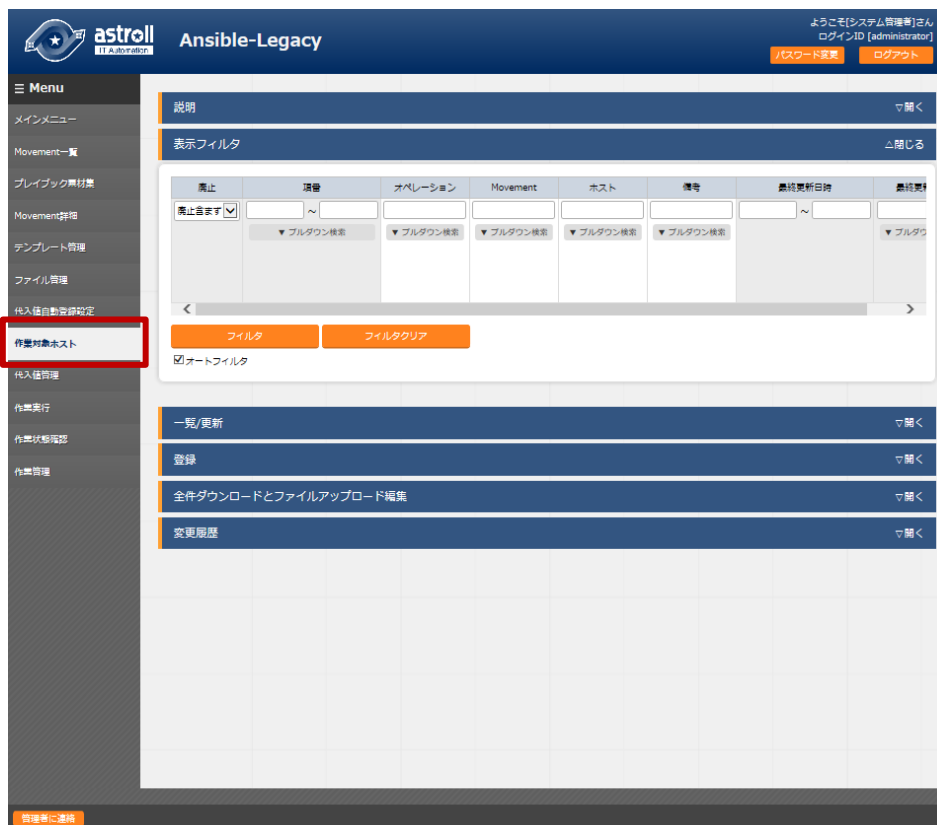


図 5.3-15 サブメニュー画面(作業対象ホスト)

登録方法の詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

5.3.12 代入値管理

- (1) 変数への代入値の登録／更新／廃止を行います。
- オペレーションごとに、対象の Movement で利用される Playbook やテンプレートファイル内の変数「VAR_」に代入する具体値をメンテナンス(閲覧/登録/更新/廃止)できます。
- また、読替表の定義により「VAR_」以外の変数「LCA_」に対して代入する具体値をメンテナンスできます。詳しくは「6.5 読替表の記述」を参照してください。
- 登録した変数の情報は作業実行時にホスト変数ファイル(host_vars/配下)に出力されます。

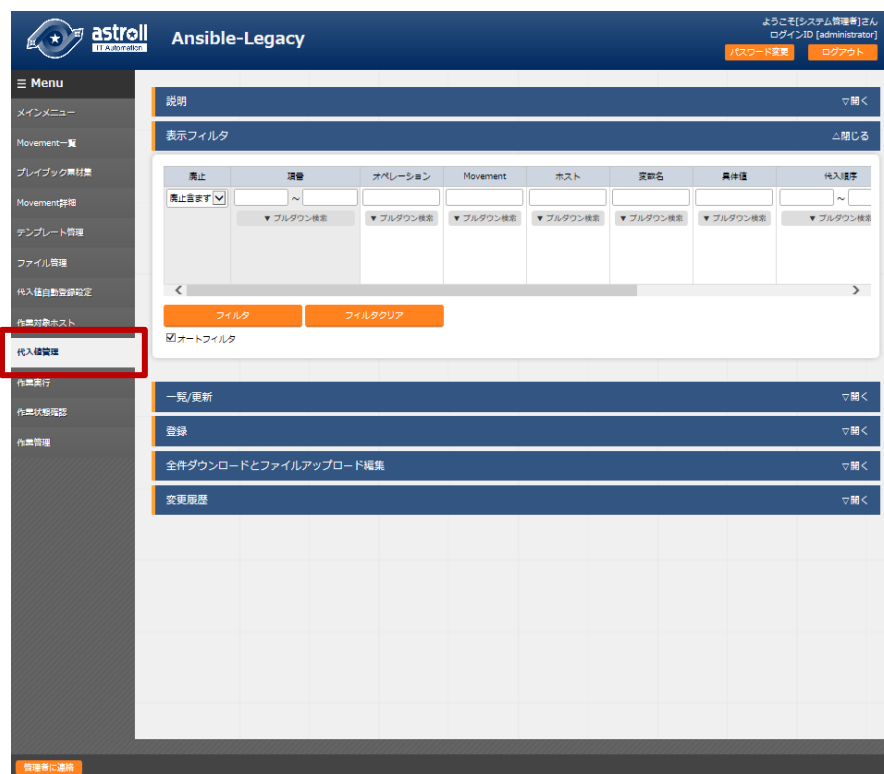


図 5.3-16 サブメニュー画面(代入値管理)
※画面は Ansible-Legacy Role のものです。

登録方法の詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

① 代入順序の入力

Ansible-Legacy では、代入順序が未入力の場合は、通常変数として扱います。

代入順序が入力されている場合は、複数具体値変数として扱います。複数具体値変数の場合は複数の具体値が必要ない場合(具体値が 1 個でよい)でも代入順序は入力してください。

Ansible-Legacy Role では、変数名またはメンバー変数名を選択することで、複数具体値変数の場合のみ代入順序が入力可能となります。複数具体値変数の場合に入力してください。

Ansible-Pioneer では、代入順序が未入力の場合は、通常変数として扱います。

代入順序が入力されている場合は、複数具体値変数として扱います。複数具体値変数の場合は、複数の具体値が必要ない場合(具体値が 1 個でよい)でも代入順序を入力してください。

各モードとも、特定の複数具体値変数に対して代入順序が連続していなくても問題ありません。

Exp)

代入値管理の登録

ホスト	変数	具体値	代入順序
HOST_A	VAR_std	value1	
HOST_A	VAR_list_a	value2	10
HOST_A	VAR_list_b	value3	100
HOST_A	VAR_list_b	value4	200

HOST_A のホスト変数ファイルへの出力内容

```
VAR_std: value1
VAR_list_a:
- value2
VAR_list_b:
- value3
- value4
```

② ホスト変数ファイルへの出力

代入値管理で登録した変数の具体値はホスト変数ファイルへ出力されます。

Ansible-Legacy と Ansible-Pioneer では、作業実行時に Playbook または対話ファイルで使用している変数の具体値が代入値管理に登録されていないと作業実行が想定外エラーとなります。

Ansible-Legacy Role では、代入値管理で具体値を登録した変数のみが作業実行時にホスト変数ファイルへ出力されます。多段変数も同様に具体値を登録しているメンバー変数のみとなります。

Exp)

変数定義

```
VAR_users:
  - name: alice
    authorized:
      - /tmp/alice/onekey.pub
  mysql:
    password: mysql-
password
  hosts:
    - "127.0.0.1"
    - "localhost"
  - name: bob
略
```

代入値管理の登録

ホスト	変数	メンバー変数	具体値	代入順序
HOST_A	VAR_users:	[0].name	value1	
HOST_A	VAR_users	[1].authorized	value2	

HOST_A のホスト変数ファイルへの出力内容

```
VAR_users:
- name :value1
- .authorized: value2
```

③ デフォルト値チェックオプション

複数ロール間でデフォルト値が一致していない変数に対して具体値の登録した場合に、警告メッセージを表示して登録させないパラメータを「astroll 管理コンソール システム設定」で設定することが出来ます。このパラメータはデフォルトでは未登録です。必要に応じて登録して下さい。

システム設定に登録する内容は以下の通りです。尚、システム設定については「利用手順マニュアル_astroll_管理コンソール」を参照下さい。

表 5.3-7-2 システム設定登録内容

項目	入力値	入力必須
識別 ID	ANSIBLE_DEF_VAL_CHK	○
項目名	任意の文字列	-
設定値	1: パラメータ有効 1 以外またはレコード未登録 : パラメータ無効	○
備考	任意の文字列	

5.3.13 作業状態確認

(1) 作業の実行状態を監視します。

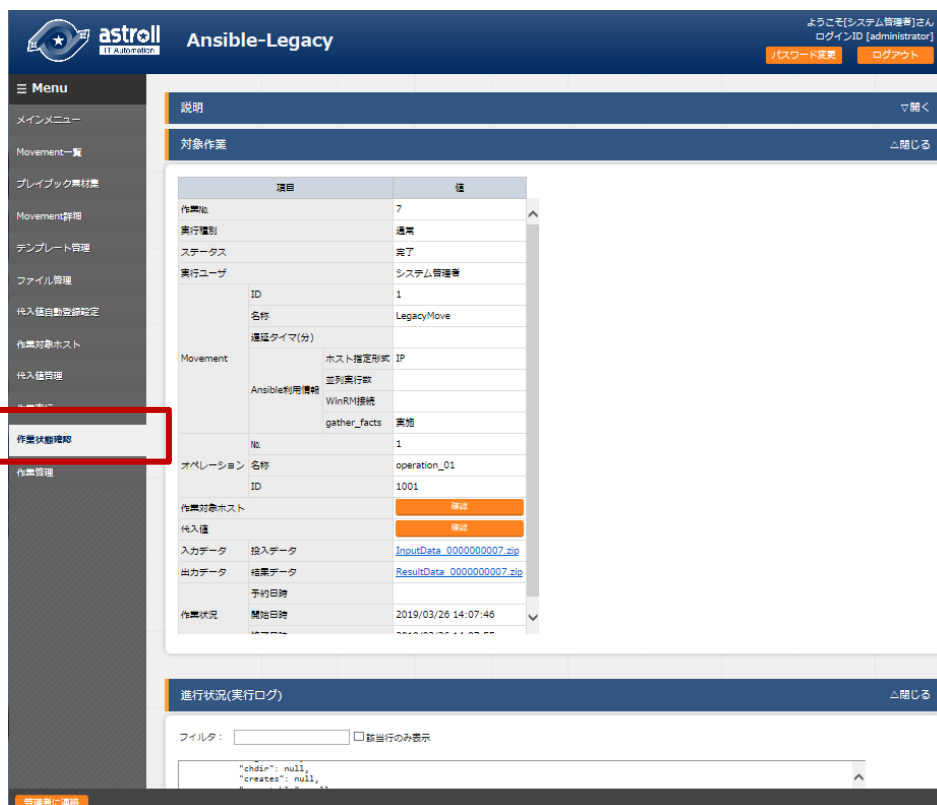


図 5.3-17 サブメニュー画面(作業状態確認)

① 実行状態表示

実行状況に即し、「ステータス」が表示されます。

また、実行ログ、エラーログに実行状況の詳細が表示されます。

「実行種別」には、ドライランの場合は「ドライラン」、それ以外は「通常」が表示されます。

ステータスが想定外エラーで終了した場合、Web コンテンツの登録不備が原因であれば、エラーログにメッセージが表示されます。

また、「5.2.1 インタフェース情報」の登録不備等で、Ansible RestAPI との通信に失敗した場合にはエラーログにメッセージが表示されません。この場合は、アプリケーションログにエラー情報が記録されます。必要に応じてアプリケーションログを確認ください。

「実行ユーザ」には、作業実行メニューより「実行」ボタンまたは「ドライラン」ボタンを押下した際のログインユーザが表示されます。

② 作業対象ホスト確認

「確認」ボタンで「5.4.11 作業対象ホスト」が表示され、作業対象のオペレーションと Movement に絞り込んだホストが表示されます。

③ 代入値確認

「確認」ボタンで「5.4.12 代入値管理」が表示され、作業対象のオペレーションと Movement に絞り込んだ代入値が表示されます。

④ 緊急停止/予約取り消し

「緊急停止」ボタンで構築作業を停止させることができます。

また、実行前の「予約実行」の作業の場合は、「予約取消」ボタンが表示されます。「予約取消」ボタンで予約実行が取り消せます。

⑤ ログ検索

実行ログ、エラーログは、フィルタリングができます。各ログのフィルタのテキストボックスに検索したい文字列を入力し、「該当行のみ表示」のチェックボックスをチェックすることで該当する行だけが表示されます。

実行ログ、エラーログのリフレッシュ表示間隔と最大表示行数を、「5.2.1 インタフェース情報」の「状態監視周期(単位ミリ秒)」と「進行状態表示行数」で設定できます。

⑥ 投入データ

実行した Playbook などをダウンロードすることができます。

⑦ 結果データ

実行ログ、エラーログなどをダウンロードすることができます。

5.3.14 作業管理

- (1) 作業の履歴を閲覧できます。
条件を指定し「フィルタ」ボタンをクリックすると、作業一覧テーブルとグラフを表示します。

それぞれのグラフにマウスカーソルを合わせると、グラフのダウンロードボタンが右上に表示されます。
表示されたグラフのダウンロードボタンをクリックすることで、グラフのダウンロードが可能です。

「作業状態確認」ボタンで、「5.3.13 作業状態確認」に遷移し、実行状態の詳細を見ることができます。

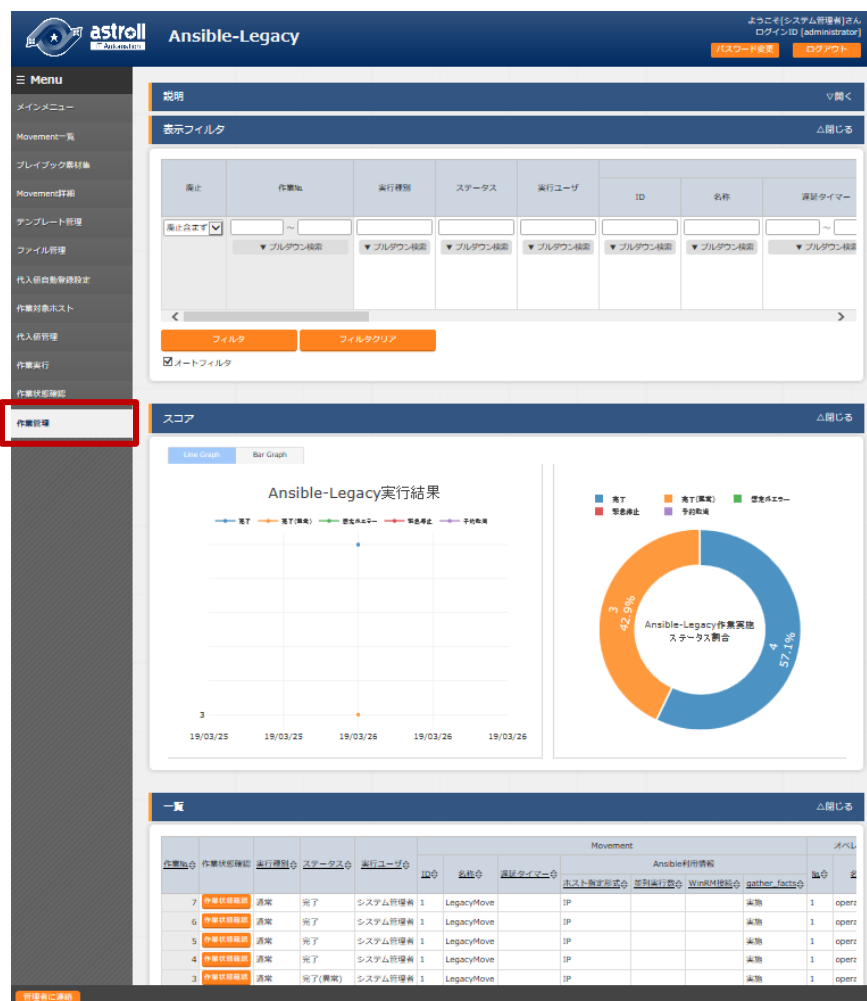


図 5.3-18 サブメニュー画面(作業管理)

5.3.15 作業実行

- (1) 作業の実行を指示します。Movement 一覧、オペレーション一覧からそれぞれラジオボタンで選択し、実行ボタンを押すと、「5.3.13 作業状態確認」に遷移し、実行されます。

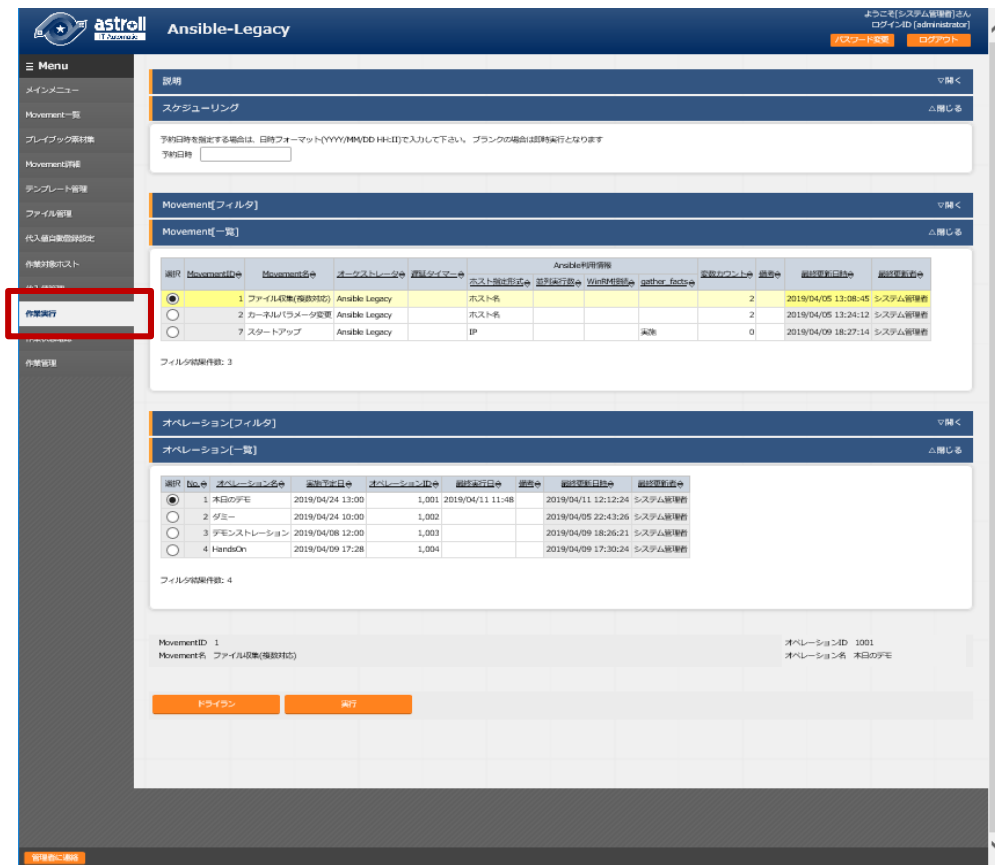


図 5.3-19 サブメニュー画面(作業実行)

※画面は Ansible-Legacy のものです。

① ドライラン

「ドライラン」ボタンをクリックすると、実際に対象機器に対して構築作業をせず、ドライランを行うことができます。ドライランを行った場合の、モード毎の動作は以下のとおりです。

Driver	動作
Ansible-Legacy	Ansible-Playbook コマンドの—check パラメータを指定し Playbook を実行します。
Ansible-Legacy Role	Ansible-Playbook コマンドの—check パラメータを指定し role を実行します。
Ansible-Pioneer	対象機器への接続チェックのみを行います。

② 予約日時の指定

「予約日時」を入力することで、実行を予約することができます。

「予約日時」には、未来の日時のみ登録可能です。

6 構築コード記述方法

6.1 Playbook（Ansible-Legacy）の記述

詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

6.2 対話ファイル（Ansible-Pioneer）の記述

対話ファイルの Ansible-Pioneer では astroll 独自モジュールを Ansible に組込んでいます。
対話ファイルは astroll 独自書式となります。
文字コードは、UTF-8 で作成してください。

(2) 対話ファイルの構成

対話ファイルは 2 種類のセクションにより構成されます。

セクション名	用途
Conf	timeout パラメータによりタイムアウト値を指定します。 タイムアウト値:1~3600(単位:秒)
exec_list	3 種類の対話コマンドにより作業対象ホストの構築を行います。

対話ファイルの先頭に timeout パラメータを記述。以降に対話コマンドを記述します。
コメントは Ansible の基本書式と同様の記述が出来ます。

```
Exp)
# コメント
conf:
△△timeout: 10
exec_list:
※△:半角スペース
timeout:の記述の前に半角スペース 2 文字を付与してください。
```

(3) 対話コマンド

対話コマンドは以下の 4 種類があります。

モジュール	用途
exec	作業対象ホストにコマンドを投入します。
expect	作業対象ホストが標準出力に出力する内容より、期待する文字列(プロンプト)の出力を待ち合せます。
state	作業対象ホストにコマンドを投入し、標準出力にプロンプトを出力するまでの標準出力の内容を外部 Shell で解析し結果判定をします。
command	作業対象ホストにコマンドを投入する前後において、繰り返しや条件分岐を行うことができます。

① expect モジュール

作業対象ホストが標準出力に出力する内容より、期待する文字列(プロンプト)の出力を待ち合せます。期待する文字列は正規表記で記述できます。期待する文字列を受取ると次へ進みます。また、timeout パラメータで指定された時間内に受取れない場合は対話ファイルを異常終了します。

```
Exp) telnet 接続でパスワード入力のプロンプトを待ち合せます。
△△-△expect:△'Password'
※△:半角スペース
- expect: の記述の前に半角スペース 2 文字を付与してください。
待ち合わせる文字列をコーテーションで囲むことを推奨します。
```

② exec モジュール

作業対象ホストにコマンドを投入します。
exec モジュールと expect モジュールは対で使用します。

```
Exp) telnet 接続でパスワード入力のプロンプトを待ち合せてパスワードを投入します。
△△-△expect:△'Password'
△△△△exec:△itapassword
※△:半角スペース
- exec: の記述の前に半角スペース 4 文字を付与してください。
```

③ state モジュール

作業対象ホストにコマンドを投入し、標準出力にプロンプトを出力するまでの標準出力の内容を外部 Shell で解析し結果判定をします。

state モジュールの書式

パラメータ	必須/ 任意	説明
△△-△state:△xxx	必須	投入するコマンドを指定します。
△△△△prompt:△xxx	必須	待受けプロンプトを指定します。正規表記で記述できます。
△△△△shell:△xxx	任意	作成した shell で結果を確認する場合に、shell ファイル名を指定します。 作成した shell の exit コードが 0 の場合は正常、他は異常と判定します。 デフォルトの shell で結果を確認する場合、本パラメータは不要となります。デフォルトの shell は parameter(-)で指定された文字列で標準出力の内容を grep します。マッチする行が 1 行でもあれば正常とし、マッチする行がなければ異常と判定します。また、parameter を指定しなかった場合、異常と判定されます。コマンドの結果(標準出力)を stdout_file で指定したファイルに退避したい目的で使用する場合、ignore_errors で「yes」を指定して下さい。
△△△△parameter: △△△△△△-△xxx △△△△△△-△xxx	任意	投入するコマンドの結果(標準出力)を検索する文字列を指定します。 shell を指定している場合、作成した shell の実行時パラメータとなります。複数ある場合は検索文字列を列挙します。
△△△△stdout_file:△xxx	任意	投入するコマンドの結果(標準出力)を退避するファイルです。 shell:を指定している場合は、このファイルからコマンドの結果を取得してください。

パラメータ	必須/ 任意	説明
△ △ △ △ success_exit: △ xxx	任意	検索結果が正常の場合に対話ファイルを正常終了する場合に「yes」を指定します。「no」の場合は正常の場合は次に進みます。デフォルトは「no」。
△ △ △ △ ignore_errors: △ xxx ※△:半角スペース	任意	検索結果が異常でも次に進む場合に「yes」を指定します。「no」の場合は、異常の場合に対話ファイルを異常終了とします。デフォルトは「no」。

Exp1)

hosts ファイルを cat し、表示結果を parameter 値で grep している。139.0.0.1、lalhost を含む行あれば正常と判定し次に進みます。行がなければ異常と判定し対話ファイルを異常終了します。

exec_list:

- state: cat /etc/hosts
- prompt: root@{{ __loginhostname__ }}
- parameter:
 - 139.0.0.1
 - lalhost

Exp2)

hosts ファイルを cat し、表示結果を parameter 値で grep している。139.0.0.1、lalhost を含む行あれば正常と判定しますが success_exit: yes の設定により対話ファイルを正常終了します。行がなければ異常と判定し対話ファイルを異常終了します。

exec_list:

- state: cat /etc/hosts
- prompt: root@{{ __loginhostname__ }}
- parameter:
 - 139.0.0.1
 - lalhost

success_exit: yes

Exp3)

hosts ファイルを cat し、表示結果を parameter 値で grep している。139.0.0.1、lalhost を含む行あれば正常と判定し次に進みます。行がなければ異常と判定しますが ignore_errors: yes の設定により次に進みます。

exec_list:

- state: cat /etc/hosts
- prompt: root@{{ __loginhostname__ }}
- parameter:
 - 139.0.0.1
 - lalhost

ignore_errors: yes

Exp4)

hosts ファイルを cat し、ユーザー作成の shell で表示結果を parameter 値で grep している。139.0.0.1、lalhost を含む行あれば正常と判定し次に進みます。行がなければ異常と判定し対話ファイルを異常終了します。

exec_list:

```
- state: cat /etc/hosts
  prompt: root@{{ __loginhostname__ }}
  shell: /tmp/grep.sh
  stdout_file: /tmp/stdout.txt
  parameter:
    - 139.0.0.1
    - lalhost
```

ユーザー作成 shell(/tmp/grep.sh)

```
#!/bin/bash
STDOUT=/tmp/STDOUT.tmp
STDERR=/tmp/STDERR.tmp
cat /tmp/stdout.txt|grep $1|grep $2 | wc -l >${STDOUT} 2>${STDERR}
RET=$?
if [ $RET -ne 0 ]; then
    EXIT_CODE=$RET
else
    if [ -s ${STDERR} ]; then
        EXIT_CODE=1
    else
        CNT=`cat ${STDOUT}`
        if [ ${CNT} -eq 0 ]; then
            EXIT_CODE=1
        else
            EXIT_CODE=0
        fi
    fi
fi
/bin/rm -rf ${STDOUT} ${STDERR} >/dev/null 2&>1
exit ${EXIT_CODE}
```

Exp5)

hosts ファイルを cat し、表示結果を stdout_file で指定したファイルに保存し次に進みます。デフォルトの shell は parameter の設定がないと異常と判定します。次に進める為に ignore_errors: yes を設定します。

exec_list:

```
- state: cat /etc/hosts
  prompt: root@{{ __loginhostname__ }}
  stdout_file: {{ __symphony_workflowdir__ }}/hosts
  ignore_errors: yes
```

④ command モジュール

作業対象ホストにコマンドを投入する前後において、繰り返しや条件分岐を行うことができます。

command モジュールの書式

パラメータ	必須/ 任意	説明
△△-△command:△xxx	必須	投入するコマンドを指定します。
△△△△prompt:△xxx	必須	待受けプロンプトを指定します。正規表記で記述できます。
△△△△timeout:△xxx	任意	コマンドを送ってからのプロンプト待ちタイマを指定します。 省略されている場合は、conf->timeout を使用します。
△△△△register:△xxx	任意	コマンドを送信後に標準出力の情報を退避する任意の文字列です。 with_items でループしている場合は、最後のコマンド送信後の標準出力の情報を退避。この変数で条件判定が出来ます。(使用できるのは条件判定のみ) ただし、変数名ごとに標準出力の情報を退避することはできません。 前の情報を上書きします。
△△△△with_items: △△△△△-△xxx △△△△△△-△xxx	任意	with_items に設定する変数は複数具体値変数です。 with_items に設定する変数の具体値数は同じでなくても良いです。 同じでない場合は、各変数の具体値数の最大値数でループします。 具体値が不足している変数の具体値は空として扱います。 ただし、prompt または timeout の場合は上記に当てはまりません。 prompt、timeout で具体値が不足していると、エラーになります。 各変数のスコープは item.X(X は 0 から 99)とします。 変数(item.X)の適用範囲は register/when 以外です。 prompt、timeout で with_items を利用する場合の変数名は下記の通りにしてください。 prompt: {{△VAR_prompt_XXX△}} timeout: {{△VAR_timeout_XXX△}} (△は半角スペース。XXX は任意の半角英数字とアンダースコア)
△△△△when: △△△△△△-△xxx △△△△△△△-△xxx	任意	command 実行前の条件判定です。 条件にマッチして command 実行します。 条件にマッチしていなければ次の command 行に移ります。 条件式 変数定義判定 VAR_xx is define 変数が定義されている true VAR_xx is undefine 変数が未定義 true ※define/undefine は astroll の変数(VAR_xx)のみ指定可能 変数具体値判定 VAR_xx/register 変数 比較演算子 文字列 VAR_xx/register 変数 比較演算子 VAR_xx VAR_xx/register 変数 match(正規表記文字列/VAR_xx) VAR_xx/register 変数 no match(正規表記文字列/VAR_xx) ※比較演算子は「=」、「!=」、「>」、「>=」、「<」、「<=」 ※比較演算子の「>」、「>=」、「<」、「<=」は数値を想定しています。 and/or による複合条件
△△△△exec_when: △△△△△△△-△xxx △△△△△△△△-△xxx	任意	ループ毎の条件判定です。(continue 条件) 条件にマッチしていれば該当ループの command を実行します。 マッチしていなければ次のループへ移ります。

パラメータ	必須/ 任意	説明
		<p>条件式</p> <p>変数定義判定</p> <p>VAR_xx is define 変数が定義されている true</p> <p>VAR_xx is undefine 変数が未定義 true</p> <p>※define/undefine は astroll の変数(VAR_xx)のみ指定可能</p> <p>変数具体値判定</p> <p>VAR_xx/register 変数 比較演算子 文字列</p> <p>VAR_xx/register 変数 比較演算子 VAR_xx</p> <p>VAR_xx/register 変数 match(正規表記文字列/VAR_xx)</p> <p>VAR_xx/register 変数 no match(正規表記文字列/VAR_xx)</p> <p>※比較演算子は「==」、「!=」、「>」、「>=」、「<」、「<=」</p> <p>※比較演算子の「>」、「>=」、「<」、「<=」は数値を想定しています。</p> <p>and/or による複合条件</p>
<p>△△△△failed_when:</p> <p>△△△△△-△xxx</p> <p>△△△△△-△xxx</p> <p>※△:半角スペース</p>	任意	<p>command 実行後(ループ毎)の stdout の内容に対する条件判定です。</p> <p>条件にマッチしていれば正常とします。</p> <p>マッチしていなければ異常とし、対話ファイルを異常終了させます。</p> <p>条件式</p> <p>変数具体値判定</p> <p>stdout 比較演算子 文字列</p> <p>stdout 比較演算子 VAR_xx</p> <p>stdout match(正規表記文字列/VAR_xx)</p> <p>stdout no match(正規表記文字列/VAR_xx)</p> <p>※比較演算子は「==」、「!=」、「>」、「>=」、「<」、「<=」</p> <p>※比較演算子の「>」、「>=」、「<」、「<=」は数値を想定しています。</p> <p>and/or による複合条件</p>

```

Exp1)
conf:
    timeout: 30

exec_list:
# プロンプト以外の文字列で待合せが必要な場合は、expect/exec の組合せです。
# パスワードが必要な場合
- expect: 'password:'
  exec:  '{{ __loginpassword__ }}'

# VAR_hosts_make という astroll 変数がホスト変数ファイルに記載されている場合、
# hosts ファイルを cat します。記載されていない場合は、スキップします。
- command: cat /etc/hosts
  prompt: root@{{ __loginhostname__ }}
  when:
    - VAR_hosts_make is define

```

```

Exp2)
conf:
    timeout: 30

exec_list:
# プロンプト以外の文字列で待合せが必要な場合は、expect/exec の組合せです。
# パスワードが必要な場合
- expect: 'password:'
  exec:   '{{ __loginpassword__ }}'

# VAR_hosts_make という astroll 変数がホスト変数ファイルに記載されている場合、
# hosts ファイルを cat します。記載されていない場合は、スキップします。
# cat により、標準出力された hosts ファイルの内容を result_stdout に退避します。
- command: cat /etc/hosts
  prompt: root@{{ __loginhostname__ }}
  register: result_stdout
  when:
    - VAR_hosts_make is define

# VAR_hosts_make という astroll 変数がホスト変数ファイルに記載されている場合、
# command 実行します。記載されていない場合は、スキップします。
# with_items の複数具体値変数に設定されている具体値数分 command 実行します。
# ループ毎の条件判定として、hosts ファイルに「ip アドレス ホスト名」が該当しない場合
# command 実行します。
# hosts ファイルの最終行に echo による、「IP アドレス ホスト名」を追記します。

- command: 'echo {{ item.0 }} {{ item.1 }} >> /etc/hosts'
  prompt: 'root@{{ __loginhostname__ }}'
  when:
    - VAR_hosts_make is define
  with_items:
    - '{{ VAR_hosts_ip }}'      # item.0
    - '{{ VAR_hosts_name }}'   # item.1
  exec_when:
    - result_stdout no match('{{ item.0 }} *{{ item.1 }}')

```

```

Exp3)
conf:
    timeout: 30

exec_list:
# プロンプト以外の文字列で待合せが必要な場合は、expect/exec の組合せです。
# パスワードが必要な場合
- expect: 'password:'
  exec: '{{ __loginpassword__ }}'

# with_items の複数具体値変数に設定されている具体値数分 command 実行します。
# 自動起動設定を実行します。
- command: 'systemctl enable {{ item.0 }}'
  prompt: 'root@{{ __loginhostname__ }}'
  with_items:
    - '{{ VAR_service_name_list }}' # item.0

# with_items の複数具体値変数に設定されている具体値数分 command 実行します。
# サービスの起動を実行します。
- command: 'systemctl start {{ item.0 }}'
  prompt: 'root@{{ __loginhostname__ }}'
  with_items:
    - '{{ VAR_service_name_list }}' # item.0

# with_items の複数具体値変数に設定されている具体値数分 command 実行します。
# サービスのステータスを標準出力します。
# 標準出力された結果の内容に、item.1 の正規表現がある場合、正となります。
# 例えば、VAR_service_status_list の具体値を running と設定し、サービスが起動している場合、
# 「Active: active (running)」の runnig が一致するので正となります。(次のループに移ります)
# そうでない場合は、異常と判断し、対話ファイルは異常終了となります。
- command: 'systemctl status {{ item.0 }}'
  prompt: 'root@{{ __loginhostname__ }}'
  with_items:
    - '{{ VAR_service_name_list }}' # item.0
    - '{{ VAR_service_status_list }}' # item.1
  failed_when:
    - stdout match('{{ item.1 }}')

```



```

Exp4)
conf:
    timeout: 30

exec_list:
# プロンプト以外の文字列で待合せが必要な場合は、expect/exec の組合せです。
# パスワードが必要な場合
- expect: 'password:'
  exec:  '{{ __loginpassword__ }}'

# with_items の複数具体値変数に設定されている具体値数分 command 実行します。
# command に「{{ item.0 }}」のみの記述をする場合は、ダブルクォーテーションで囲みます。
# prompt や timeout で with_items を利用する場合、具体値数に注意が必要です。
# prompt→command→prompt→command→prompt ... (以下ループ) となり、command 数+1
# 設定する必要があります。(timeout も同様)
- command: "{{ item.0 }}"
  prompt: '{{ item.1 }}'
  timeout: '{{ item.2 }}'
  with_items:
    - '{{ VAR_command_list }}' # item.0
    - '{{ VAR_prompt_list }}' # item.1
    - '{{ VAR_timeout_list }}' # item.2

```

```

Exp5)
conf:
    timeout: 30

exec_list:
# プロンプト以外の文字列で待合せが必要な場合は、expect/exec の組合せです。
# パスワードが必要な場合
- expect: 'password:'
  exec:  '{{ __loginpassword__ }}'

# with_items の複数具体値変数に設定されている具体値数分 command 実行します。
# 代入値管理の具体値に{{ item.X }} を設定することができます。その際は対話ファイルに記載する
# item.X より具体値に記載する item.X の数値が大きくなるようにしてください。
# 今回の例で実行する command は「systemctl status ky_pioneer_execute-workflow.service」
- command: "{{ item.0 }}"
  prompt: 'root@{{ __loginhostname__ }}'
  with_items:
    - '{{ VAR_command_list }}' # item.0
    - '{{ VAR_service_name_list }}' # item.1

```

変数名	具体値
VAR_command_list	systemctl status {{ item.1 }}
VAR_service_name_list	ky_pioneer_execute-workflow.service

(4)

正規表記

下記のコマンド及びパラメータに記述された文字列は正規表記で評価されます。

- expect モジュール
- state モジュールの prompt パラメータ
- command モジュールの prompt パラメータ

ですので正規表記での記述が可能です。また、記述された文字列にメタ文字「(){}など」を含む場合、メタ文字の前にエスケープ文字「¥」を挿入する必要があります。

Exp1)

以下のような文字列を待ち受ける場合、赤字がメタ文字となります。

XAMPP Developer Files [Y/n] exec_list:

メタ文字の前にエスケープ文字「¥」を挿入する必要があります。

XAMPP Developer Files ¥[Y¥/n¥] exec_list:

6.3 ロールパッケージ（Ansible-Legacy Role）の記述

詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

6.4 astrollreadme（Ansible-Legacy Role のみ）の記述

詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

6.5 読替表（Ansible-Legacy Role のみ）の記述

詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

6.6 BackYard コンテンツ

詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

(1) 作業インスタンス履歴削除

astroll 基本コンソールの投入オペレーション一覧に登録されてるオペレーションで実施予定日が一定期間経過しているオペレーション ID を「5.3.11 作業対象ホスト」と「5.3.12 代入値管理」で使用しているレコードがある場合、経過日数に応じて削除(物理削除または廃止)します。

下記ファイルに経過日数を登録しています。この日数に応じてレコードが削除(物理削除または廃止)されます。

●経過日数登録ファイル

~/ita-root/confs/backyardconfs/ansible_driver/keep_day_length.txt

●ファイルフォーマット

p1,p2 p1:廃止までの日数 p2:物理削除までの日数

Exp)

廃止までの日数: 30 日 物理削除までの日数: 60 日の場合

30,60

6.7 Ansible 利用ガイドライン astroll 追加ルール

詳細は、関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』をご参照下さい。

7 運用上の注意点

astroll システムを活用する操作はクライアント PC のブラウザ画面からのユーザー利用による入力だけでは無く、システム運用・保守による操作もあります。

用意している運用・保守の操作は次のとおりです。

- ログレベルの変更
- メンテナンス

7.1 ログレベルの変更

astroll システム のプロセスのログレベルの変更方法は次のとおりです。

ログレベルを変更できるプロセスには、通常の独立監視プロセスと、RestAPI のプロセスがあります。

(1) 通常の独立監視プロセス

① RHEL 6.x の場合の対象ファイル

~/ita-root/backyards/ansible_driver/

(Ansible 操作)

```
ky_legacy_checkcondition-workflow
ky_legacy_execute-workflow
ky_legacy_role_checkcondition-workflow
ky_legacy_role_execute-workflow
ky_pioneer_checkcondition-workflow
ky_pioneer_execute-workflow
```

(変数自動登録)

```
ky_legacy_varsautolistup-workflow
ky_legacy_role_varsautolistup-workflow
ky_pioneer_varsautolistup-workflow
```

(代入値自動登録設定)

```
ky_legacy_valautostup-workflow
ky_legacy_role_valautostup-workflow
ky_pioneer_valautostup-workflow
```

(構成管理 DB 連携)

```
ky_ansible_cmdbmenuanalysis-workflow
```

※ 上記ファイルへのリンクファイルが /etc/init.d に作成されています。これらは削除しないでください。

NORMAL レベル

「LOG_LEVEL='NORMAL'」を有効にします。

```
# ログ出力レベル
#  DEBUG :解析レベルでログ出力
#  NORMAL:クリティカルな場合のみログ出力
#LOG_LEVEL='DEBUG'
```

DEBUG レベル

「LOG_LEVEL='DEBUG'」を有効にします。

```
# ログ出力レベル
#  DEBUG :解析レベルでログ出力
#  NORMAL:クリティカルな場合のみログ出力
LOG_LEVEL='DEBUG'
```

② RHEL 7.x の場合の対象ファイル

/etc/systemd/system/

(Ansible 操作)

```
ky_legacy_role_execute-workflow.service
ky_legacy_role_checkcondition-workflow.service
ky_legacy_execute-workflow.service
ky_legacy_checkcondition-workflow.service
ky_pioneer_execute-workflow.service
ky_pioneer_checkcondition-workflow.service
```

(変数自動登録)

```
ky_legacy_role_varsautolistup-workflow.service
ky_legacy_varsautolistup-workflow.service
ky_pioneer_varsautolistup-workflow.service
```

(代入値自動登録設定)

```
ky_legacy_role_valautostup-workflow.service
ky_legacy_valautostup-workflow.service
ky_pioneer_valautostup-workflow.service
```

(構成管理 DB 連携)

```
ky_ansible_cmdbmenuanalysis-workflow
```

NORMAL レベル

NORMAL を設定にします。

```
ExecStart=/astroll/ita-root/backyards/common/ky_loopcall-php-procedure.sh
/usr/local/bin/php /usr/local/bin/php /astroll/ita-
root/backyards/ansible_driver/ky_pioneer_varsautolistup-workflow.php
/astroll/ita-root/logs/backyardlogs 60 NORMAL > /dev/null 2>&1
```

DEBUG レベル

DEBUG を設定にします。

```
ExecStart=/astroll/ita-root/backyards/common/ky_loopcall-php-procedure.sh
/usr/local/bin/php /usr/local/bin/php /astroll/ita-
root/backyards/ansible_driver/ky_pioneer_varsautolistup-workflow.php
/astroll/ita-root/logs/backyardlogs 60 DEBUG > /dev/null 2>&1
```

※ ログレベル変更は、プロセス再起動(Restart)後に有効になります。再起動については次節「[7.3 メンテナンス方法について](#)」を参照してください。

③ RHEL 6.x と RHEL 7.x で共通の対象ファイル

~/ita-root/backyards/ansible_driver/ky_ansible_dataautoclean-workflow.sh

「②RHEL 6.x の場合の対象ファイル」の設定と同様です。

7.2 起動周期の変更

astroll システム のプロセスの起動周期の変更方法は次のとおりです。
ただし、例外を除き起動周期はデフォルト値の使用をしてください。
起動周期を変更できるプロセスには、通常の独立監視プロセスのみです。

(1) 通常の独立監視プロセス

① RHEL 6.x の場合の対象ファイル

~/ita-root/backyards/ansible_driver/

(Ansible 操作)

ky_legacy_checkcondition-workflow
ky_legacy_execute-workflow
ky_legacy_role_checkcondition-workflow
ky_legacy_role_execute-workflow
ky_pioneer_checkcondition-workflow
ky_pioneer_execute-workflow

(変数自動登録)

ky_legacy_varsautolistup-workflow
ky_legacy_role_varsautolistup-workflow
ky_pioneer_varsautolistup-workflow

(代入値自動登録設定)

ky_legacy_valautostup-workflow
ky_legacy_role_valautostup-workflow
ky_pioneer_valautostup-workflow

(構成管理 DB 連携)

ky_ansible_cmdbmenuanalysis-workflow

上記ファイルへのリンクファイルが /etc/init.d に作成されています。これらは削除しないでください。

・起動周期の設定

INTERVAL で起動周期を設定します。(単位:秒)

```
# 無限ループのインターバル(0 以上を指定、180 未満にすること)
INTERVAL=60
```

② RHEL 7.x の場合の対象ファイル

/etc/systemd/system/

(Ansible 操作)

```
ky_legacy_role_execute-workflow.service
ky_legacy_role_checkcondition-workflow.service
ky_legacy_execute-workflow.service
ky_legacy_checkcondition-workflow.service
ky_pioneer_execute-workflow.service
ky_pioneer_checkcondition-workflow.service
```

(変数自動登録)

```
ky_legacy_role_varsautolistup-workflow.service
ky_legacy_varsautolistup-workflow.service
ky_pioneer_varsautolistup-workflow.service
```

(代入値自動登録設定)

```
ky_legacy_role_valautostup-workflow.service
ky_legacy_valautostup-workflow.service
ky_pioneer_valautostup-workflow.service
```

(構成管理 DB 連携)

```
ky_ansible_cmdbmenuanalysis-workflow
```

・起動周期の設定

ExecStart の 5 番目のパラメータで設定します。(単位:秒)

```
ExecStart=/astroll/ita-root/backyards/common/ky_loopcall-php-procedure.sh
/usr/local/bin/php /usr/local/bin/php /astroll/ita-
root/backyards/ansible_driver/ky_pioneer_varsautolistup-workflow.php /astroll/ita-
root/logs/backyardlogs 60 NORMAL > /dev/null 2>&1
```

7.3 メンテナンス方法について

7.3.1 Ansible driver 独立型プロセスの起動/停止/再起動

ky_legacy_checkcondition-workflow を例示します。

① RHEL 6.x の場合

●プロセス起動

```
$ service ky_legacy_checkcondition-workflow start ↵
```

●プロセス停止

```
$ service ky_legacy_checkcondition-workflow stop ↵
```

●プロセス再起動

```
$ service ky_legacy_checkcondition-workflow restart ↵
```

② RHEL 7.x の場合

●プロセス起動

```
$/usr/bin/systemctl start ky_legacy_checkcondition-workflow ↵
```

●プロセス停止

```
$/usr/bin/systemctl stop ky_legacy_checkcondition-workflow ↵
```

●プロセス再起動

```
$/usr/bin/systemctl restart ky_legacy_checkcondition-workflow ↵
```

同様に、各対象ファイル名に置き換えて起動/停止/再起動を行ってください。

8 トラブルシューティング

関連マニュアルの『利用手順マニュアル astroll_Ansible 系ドライバー共通』にも、共通の QA 事項について記載をしています。そちらも併せてご参照ください。

No	内容
Q-1	機器一覧のユーザーに root ユーザー以外を設定したホストに対して legacy の作業実行をしたが、Ansible のコマンドでエラーが発生する。
A-1	astroll で Ansible のコマンドを実行する場合に sudo パラメータのデフォルトを yes に設定しています。 作業対象ホスト側で該当ユーザーに対する sudo 権限を設定してください。 sudo 権限を与えられない場合は sudo パラメータの設定を Playbook に追記してください。
Q-2	機器一覧のプロトコルを telnet に設定したホストに対して pioneer の作業実行をしたが、作業対象ホストへの接続でエラーとなる。
A-2	対話ファイルに記述しているパスワードおよびパスワードプロンプトの記述が正しいか確認してください。

9 付録

9.1 Ansible 実行時に使用される投入データと astroll メニューの紐づけ

各 astroll メニューより情報を抜出して Ansible 実行に必要な[投入データ]を作ります。

[投入データ]は ZIP 形式で[作業状態確認](#)よりダウンロードが可能です。

各種データと astroll メニューの関係性は以下の通りです。

9.1.1 Ansible-Legacy 投入データ

【上位ディレクトリ】

child_playbooks	ユーザーが作成した Playbook が格納されているディレクトリ		
	{	Ansible-Legacy プレイブック素材集	プレイブック素材
		Ansible-Legacy Movement 詳細	インクルード順序
template_files	実行する Playbook 内で使用する template ファイルを格納するディレクトリ		
	{	Ansible-Legacy テンプレート管理	テンプレート素材
		Ansible-Legacy Movement 詳細	インクルード順序
copy_files	作業対象サーバに配置するファイルを格納されているディレクトリ		
	{	Ansible-Legacy ファイル管理	ファイル素材
		Ansible-Legacy Movement 詳細	インクルード順序
host_vars	作業実行の対象となるホストの情報と各種変数の定義ファイルが格納されるディレクトリ		
	{	Ansible 共通 グローバル変数管理	変数名/具体値
		Ansible-Legacy 代入値管理	変数名/具体値
		Ansible-Legacy template 管理	テンプレート素材
		Ansible-Legacy ファイル管理	ファイル変数名
		Ansible-Legacy Movement 詳細	インクルード順序
		Ansible-Legacy インターフェース情報	データリレイストレージパス (astroll)
		Ansible-Legacy インターフェース情報	Symphony インスタンスデータリレイストレージパス(ANS)
		基本コンソール 機器一覧	プロトコル
		基本コンソール 機器一覧	ログインユーザ ID
		基本コンソール 機器一覧	ログインパスワード
		基本コンソール 機器一覧	ホスト名

ssh_key_files	認証方式が鍵方式の場合、指定した ssh 認証鍵ファイルが格納されるディレクトリ		
	{	基本コンソール 機器一覧	ssh 認証鍵ファイル
winrm_ca_files	WinRM 接続する場合、接続情報を定義したファイルが格納されるディレクトリ		
	{	基本コンソール 機器一覧	WinRM 接続情報
AnsibleExecOption.txt	AnsiblePlaybook 実行時のパラメータ		
	{	Ansible 共通 インターフェース情報 オプションパラメータ	並列実行数
		Ansible-Legacy Movement 一覧	
hosts	作業実行の対象となるホストの情報が記載されているファイル		
	{	基本コンソール 機器一覧	ホスト名
		基本コンソール 機器一覧	IP アドレス
		基本コンソール 機器一覧	ログインユーザ ID
		基本コンソール 機器一覧	ログインパスワード
		基本コンソール 機器一覧	接続オプション
			※ansible_ssh_extra_args のパラメータ
		基本コンソール 機器一覧	ssh 認証鍵ファイル
		基本コンソール 機器一覧	サーバー証明書
		基本コンソール 機器一覧	インベントリファイル追加オプション
playbook.yml	Playbook やホストの情報を全て呼び出し、Ansible を実行するファイル。		
	{	Ansible-Legacy プレイブック素材集	プレイブック素材
		Ansible-Legacy Movement 詳細	インクルード順序
		Ansible-Legacy Movement 詳細	gather_facts

投入データを用いて以下のコマンドで Ansible を直接実行することも可能です。

ansible-playbook (オプション) -i hosts playbook.yml

9.1.2 Ansible-Pioneer 投入データ

【上位ディレクトリ】

template_files	実行する Playbook 内で使用する template ファイルを格納するディレクトリ		
	{	Ansible-Pioneer テンプレート管理 Ansible-Pioneer Movement 詳細	テンプレート素材 インクルード順序
copy_files	作業対象サーバに配置するファイルを格納されているディレクトリ		
	{	Ansible-Pioneer ファイル管理 Ansible-Pioneer Movement 詳細	ファイル素材 インクルード順序
ssh_key_files	認証方式が鍵方式の場合、指定した ssh 認証鍵ファイルが格納されるディレクトリ		
	{	基本コンソール 機器一覧	ssh 認証鍵ファイル
winrm_ca_files	W イン RM 接続する場合は接続情報を定義したファイルが格納されるディレクトリ		
	{	基本コンソール 機器一覧	WinRM 接続情報
host_vars	作業実行の対象となるホストの情報と各種変数の定義ファイルが格納されるディレクトリ		
	{	Ansible 共通 インターフェース情報	データリレイストレージパス (astroll)
		Ansible 共通 インターフェース情報	Symphony インスタンスデータリレイストレージパス(ANS)
		Ansible 共通 グローバル変数管理	変数名/具体値
		Ansible-Pioneer 代入値管理	変数名/具体値
		Ansible-Pioneer template 管理	テンプレート素材
		Ansible-Pioneer Movement 詳細	インクルード順序
		Ansible-Pioneer ファイル管理	ファイル変数名
		Ansible-Pioneer Movement 詳細	インクルード順序
		基本コンソール 機器一覧	ログインパスワード
		基本コンソール 機器一覧	ホスト名
		基本コンソール 機器一覧	接続オプション
		基本コンソール 機器一覧	プロトコル
		基本コンソール 機器一覧	ログインユーザ ID
dialog_files	ユーザーが作成した Playbook が格納されているディレクトリ		
	{	Ansible-Pioneer 対話ファイル素材集 Ansible-Pioneer Movement 詳細	対話ファイル素材 インクルード順序

AnsibleExecOption.txt	AnsiblePlaybook 実行時のパラメータ		
	<div> <div>{</div> <div>Ansible 共通</div> <div>インターフェース情報</div> <div>オプションパラメータ</div> <div>}</div> </div>		
hosts	作業実行の対象となるホストの情報が記載されているファイル		
	<div> <div>{</div> <div>基本コンソール</div> <div>機器一覧</div> <div>基本コンソール</div> <div>機器一覧</div> <div>基本コンソール</div> <div>機器一覧</div> <div>基本コンソール</div> <div>機器一覧</div> <div>基本コンソール</div> <div>機器一覧</div> <div>基本コンソール</div> <div>機器一覧</div> <div>}</div> </div>		
			<div> <div>ホスト名</div> <div>IP アドレス</div> <div>ログインユーザ ID</div> <div>ログインパスワード</div> <div>ssh 認証鍵ファイル</div> <div>インベントリファイル追加オプション</div> </div>
playbook.yml	Playbook やホストの情報を全て呼び出し、Ansible を実行するファイル。		
	<div> <div>{</div> <div>Ansible-Pioneer</div> <div>インターフェース情報</div> <div>データリレイストレージパス (astroll)</div> <div>}</div> </div>		

投入データを用いて以下のコマンドで Ansible を直接実行することも可能です。
ansible-playbook (オプション) -i hosts playbook.yml

9.1.3 Ansible-LegacyRole 投入データ

【上位ディレクトリ】

copy_files	作業対象サーバに配置するファイルを格納されているディレクトリ		
	Ansible-LegacyRole	ファイル管理	ファイル素材
	Ansible-LegacyRole	Movement 詳細	インクルード順序
roles	ユーザーが作成した role が格納されているディレクトリ		
	Ansible-LegacyRole	ロールパッケージ管理	ロールパッケージファイル (ZIP 形式)
ssh_key_files	認証方式が鍵方式の場合、指定した ssh 認証鍵ファイルが格納されるディレクトリ		
	基本コンソール	機器一覧	ssh 認証鍵ファイル
winrm_ca_files	WinRM 接続する場合は接続情報を定義したファイルが格納されるディレクトリ		
	基本コンソール	機器一覧	WinRM 接続情報
host_vars	作業実行の対象となるホストの情報と各種変数の定義ファイルが格納されるディレクトリ		
	Ansible 共通	インターフェース情報	データリレイストレージパス (astroll)
	Ansible 共通	インターフェース情報	Symphony インスタンスデータリレイストレージパス (ANS)
	Ansible 共通	グローバル変数管理	変数名/具体値
	Ansible-LegacyRole	代入値管理	変数名/具体値
	Ansible-LegacyRole	template 管理	テンプレート素材
	Ansible-LegacyRole	Movement 詳細	インクルード順序
	Ansible-LegacyRole	ファイル管理	ファイル変数名
	Ansible-LegacyRole	Movement 詳細	インクルード順序
	基本コンソール	機器一覧	プロトコル
	基本コンソール	機器一覧	ログインユーザ ID
	基本コンソール	機器一覧	ログインパスワード
	基本コンソール	機器一覧	ホスト名

AnsibleExecOption.txt	AnsiblePlaybook 実行時のパラメータ		
	Ansible 共通	インターフェース情報	オプションパラメータ
	Ansible-LegacyRole	Movement 一覧	並列実行数
hosts	作業実行の対象となるホストの情報が記載されているファイル		
	基本コンソール	機器一覧	ホスト名
	基本コンソール	機器一覧	IP アドレス
	基本コンソール	機器一覧	ログインユーザ ID
	基本コンソール	機器一覧	ログインパスワード
	基本コンソール	機器一覧	接続オプション
			※ansible_ssh_extra_args のパラメータ
	基本コンソール	機器一覧	ssh 認証鍵ファイル
	基本コンソール	機器一覧	サーバー証明書
	基本コンソール	機器一覧	インベントリファイル
			追加オプション
site.yml	Playbook やホストの情報を全て呼び出し、Ansible を実行するファイル。		
	Ansible-Legacy	プレイブック素材集	プレイブック素材
	Ansible-Legacy	Movement 詳細	インクルード順序
	Ansible-Legacy	Movement 詳細	gather_facts

投入データを用いて以下のコマンドで Ansible を直接実行することも可能です。
 ansible-playbook (オプション) -i hosts site.yml