



IT Automation

収集機能・比較機能 【実習編】

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

Exastro IT Automation Version 1.9
Exastro developer

目次

はじめに

- [\(1\) 本書について](#)
- [\(2\) 作業環境](#)
- [\(3\) シナリオ](#)

1. 実習 1 【収集機能】ターゲットホストのOS情報を収集する

[実習 1 全体図](#)

- [1.1 ターゲットホストの登録](#)
- [1.2 オペレーションの登録](#)
- [1.3 Movementの登録](#)
 - [1.3.1 ヘッダーセクションとgather_facts](#)
- [1.4 Playbookの登録](#)
 - [1.4.1 YAMLファイルと収集用ディレクトリ](#)
- [1.5 Movement-Playbook紐付](#)
- [1.6 作業対象ホストの登録](#)
- [1.7 収集値を登録するパラメータシートの作成](#)
- [1.8 収集項目値管理の登録](#)
- [1.9 収集インターフェース情報の登録](#)
- [1.10 作業実行](#)
- [1.11 収集結果の確認](#)

2. 実習 2 【比較機能】実習 1 で収集した値と期待値を比較する

[実習 2 全体図](#)

- [2.1 オペレーションの登録](#)
 - [2.2 期待値用パラメータシートの作成](#)
 - [2.3 期待値の登録](#)
 - [2.4 比較定義の登録](#)
 - [2.5 比較実行](#)
- [【参考】比較定義詳細](#)
- [【参考】\(1\) 比較定義の登録](#)

目次

[【参考】 \(2\) 比較定義詳細の登録](#)

[【参考】 \(3\) 比較実行](#)

3. 実習3 【収集機能】 ターゲットホストのSSL証明書ファイルを収集する

[実習3全体図](#)

[3.1 ターゲットホストの登録](#)

[3.2 オペレーションの登録](#)

[3.3 Movementの登録](#)

[3.4 Playbookの登録](#)

[3.4.1 ファイルの収集用ディレクトリ](#)

[3.5 Movement-Playbook紐付](#)

[3.6 ファイル名の登録](#)

[3.7 代入値自動登録設定の登録](#)

[3.8 収集値を登録するパラメータシートの作成](#)

[3.9 収集項目値管理の登録](#)

[3.10 収集インターフェース情報の登録](#)

[3.11 作業実行](#)

[3.12 収集結果の確認](#)

4. 実習4 【比較機能】 実習3で収集したSSL証明書ファイルを、異なる日時に収集したファイルと比較する

[実習4全体図](#)

[4.1 オペレーションの登録](#)

[4.2 差分有りSSL証明書の用意](#)

[4.3 ファイル名の登録](#)

[4.4 作業実行](#)

[4.5 収集結果の確認](#)

[4.6 比較定義の登録](#)

[4.7 比較実行](#)

はじめに



(1) 本書について

本書について

本書では「収集機能」と「比較機能」について、実習形式で作業を進め、理解を深めていただけます。



Menu

DASHBOARD

編集

メインメニュー

システム設定

メニュー グループ 管理

メニュー 管理

ロール 管理

ユーザ 管理

ロール・メニュー 紐付 管理

ロール・ユーザ 紐付 管理

シーケンス 管理

SSO 基本 情報 管理

SSO 属性 情報 管理

バージョン 確認

Ansible-Legacy

Ansible-Pioneer

Ansible-LegacyRole

メニュー グループ

管理 コンソール

基本 コンソール

エクスポート / イン…

Symphony

Conductor

メニュー 作成

入力 用

代入 値自動登録 用

参照 用

比較

ホスト グループ 管理

Ansible 共通

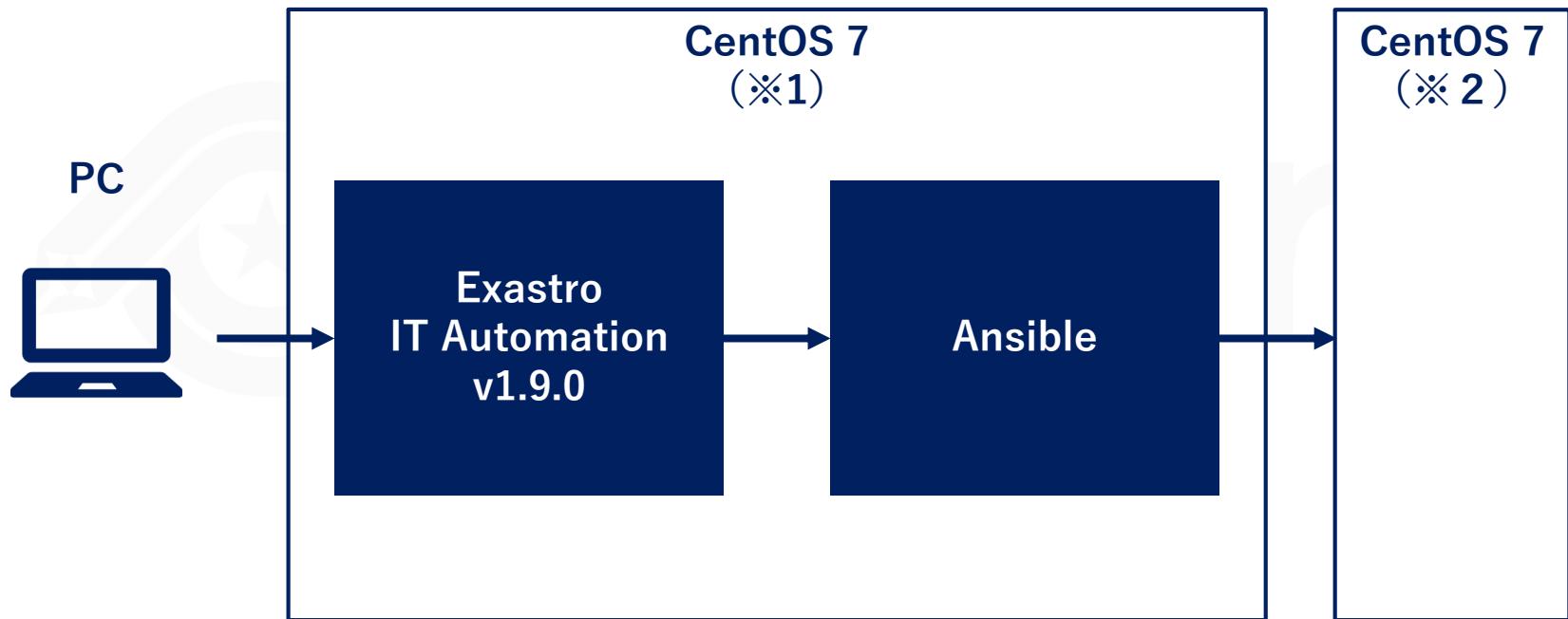
収集機能に関するメニュー

- 収集インターフェース情報
- 収集項目値管理

(2) 作業環境

■ 作業環境

本書で使用する作業環境は以下の通りです。



※1 ITAはRHEL7系およびRHEL8系のOSで導入いただけます。

※2 Ansibleの動作対象となれるOSであれば、問題なくご利用いただけます。

(3) シナリオ

■ パラメータの収集・比較とファイルの収集・比較

- 実習1と2ではパラメータの収集・比較、実習3と4ではファイルの収集・比較を解説しています。
- 具体的な収集対象は「パラメータ：OS情報」「ファイル：SSL証明書ファイル」となります。

	収集機能	比較機能
パラメータの 収集・比較	実習1 ターゲットホストのOS情報を 収集する	実習2 実習1で収集した値と期待値 を比較する
ファイルの 収集・比較	実習3 ターゲットホストのSSL証明 書ファイルを収集する	実習4 実習3で収集したSSL証明書 ファイルを、異なる日時に収 集したファイルと比較する

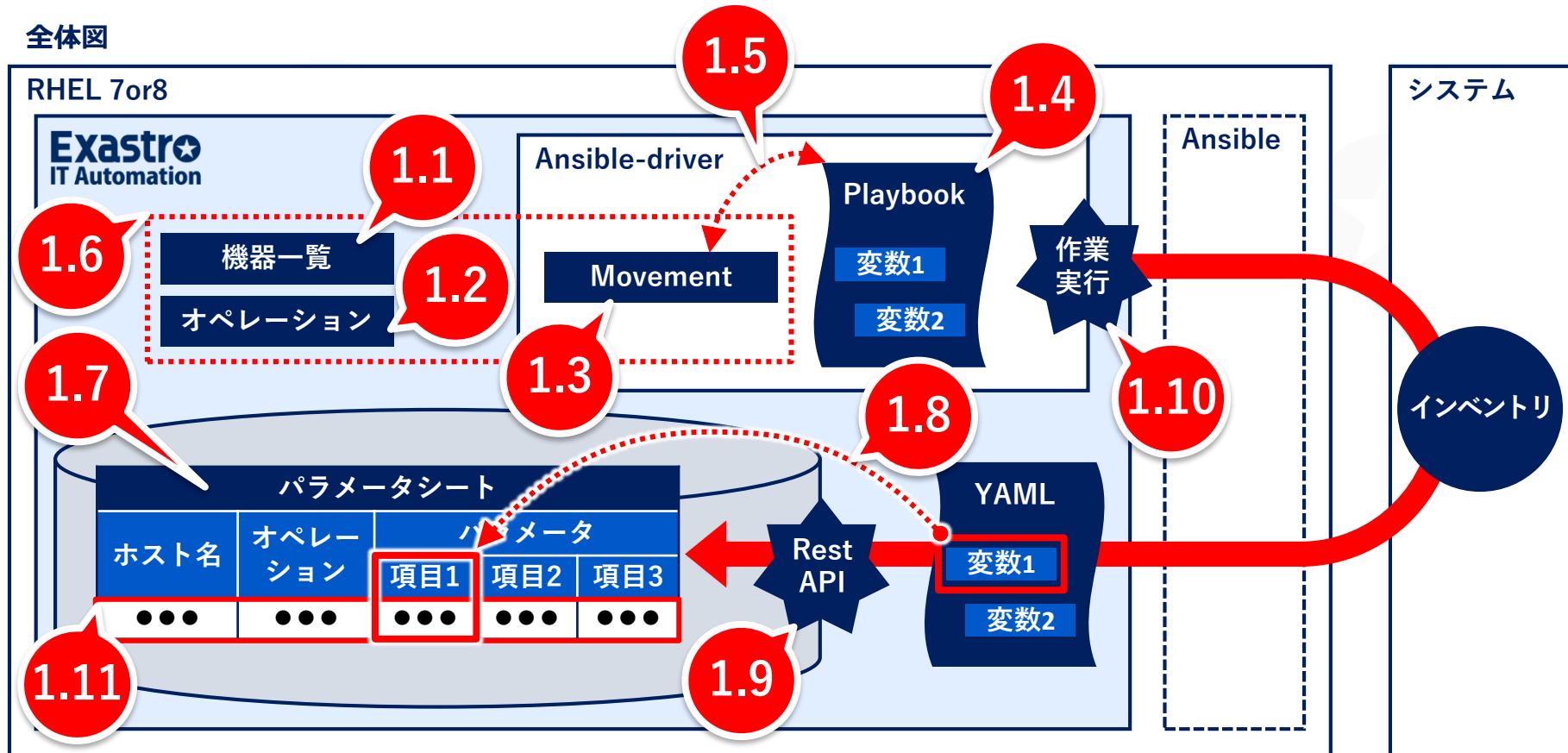
1. 実習 1 【収集機能】ターゲットホストのOS情報 を収集する

実習 1 全体図

実習 1 の作業の流れ

- 数字は本書の章番号です。
- 各種設定を行ってから作業実行し、インベントリ（OS情報）を収集してパラメータシートへ自動登録します。

全体図



1.1 ターゲットホストの登録

■ ターゲットホストの接続情報を登録する

「機器一覧」から登録します。

メニュー：基本コンソール > 機器一覧

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

The screenshot shows the 'Machine List' registration screen. The fields highlighted with red boxes are: 'HW機器種別' (selected as 'SV'), 'ホスト名' (entered as 'targethost'), 'IPアドレス' (entered as '192.0.2.1'), 'ログインユーザID' (selected as 'root'), 'ログインパスワード' (selected as '管理'), and '認証方式' (selected as 'パスワード認証').

HW 機器種別	ホスト名 (任意の名称)	IPアドレス (ご使用の環境 に応じて)	ログイン ユーザID (ご使用 の環境 に応じて)	ログインパスワード		Ansible利用情報 Legacy/Role利用 情報
				管理	ログイン パスワード (ご使用の環境 に応じて)	
					認証方式	
SV	targethost	192.0.2.1	root	●	*****	パスワード認証

1.2 オペレーションの登録

今回のオペレーションを登録する

オペレーションはITAの自動作業一式を指します。

この後、関連するすべてのデータをオペレーションに紐付けていきます。

メニュー：基本コンソール > オペレーション一覧

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、 [登録] ボタンを押下する。

No.	オペレーションID	オペレーション名*	実施予定日時*	アクセス権	設定
自動入力	自動入力	GatherFacts1	2021/04/22 17:09	アクセス許可ロール	設定

オペレーション名 (任意の名称)	実施予定時間 (任意の時間)
GatherFacts1	2021/04/22 17:09

実施予定時間は情報として入力できるようになっているもので、タイマーではありません。

オペレーション名は任意の名称です。

1.3 Movementの登録

Ansible-LegacyのMovementを登録する

Movementとは、ITAの自動作業の最小単位で、ジョブを表します。

この後Playbookに紐付けて、OS情報を収集するMovementとなります。

メニュー : Ansible-Legacy > Movement一覧

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、 [登録] ボタンを押下する。

MovementID	Movement名*	遅延タイマー	ホスト指定形式*	Ansible利用情報	ヘッダーセクション
自動入力	GatherFacts		IP		<pre>- hosts: all remote_user: "{{ __loginuser__ }}" gather_facts: yes become: yes</pre>

Movement名 (任意の名称)	Ansible利用情報	
	ホスト指定形式	ヘッダーセクション
GatherFacts	IP	<pre>- hosts: all remote_user: "{{ __loginuser__ }}" gather_facts: yes become: yes</pre>

gather_factsを有効にします。
※ 詳細は『1.3.1 ヘッダーセクションとgather_facts』参照)

1.3.1 ヘッダーセクションとgather_facts

gather_factsを有効にする

- ITAのデフォルトの設定では、AnsibleのPlaybookのヘッダーセクションのgather_factsが無効になっています。
- この実習ではgather_factsを使ってOS情報を取得するため、ヘッダーセクションに下記のように入力して、有効にしておきます。
- デフォルト値から変更する必要がない場合、ヘッダーセクションは空欄でOKです。

デフォルト

```
- hosts: all
  remote_user: "{{ __loginuser__ }}"
  gather_facts: no
  become: yes
```



gather_factsをyesにする

```
- hosts: all
  remote_user: "{{ __loginuser__ }}"
  gather_facts: yes
  become: yes
```

ヘッダーセクションにすべての行を記入し、変更する項目の設定値を変更します。

1.4 Playbookの登録（1/2）

■ 作業実行用Playbookを作成する

- 取得したOS情報を記載したYAMLファイルを生成する、という内容のPlaybookを作成します。
- YAMLファイルと収集用ディレクトリについては、[『1.4.1 YAMLファイルと収集用ディレクトリ』](#)を参照してください。

```
- name: make yaml file
blockinfile:
  create: yes
  mode: 644
  insertbefore: EOF
  marker: ""
  dest: "{{ __parameter_dir__ }}/{{ inventory_hostname }}/gatherfacts.yml"
  content: |
    ansible_architecture      : {{ ansible_architecture }}
    ansible_bios_version      : {{ ansible_bios_version }}
    ansible_default_ipv4_address : {{ ansible_default_ipv4.address }}
    ansible_default_ipv4_interface : {{ ansible_default_ipv4.interface }}
    ansible_default_ipv4_network : {{ ansible_default_ipv4.network }}
    ansible_distribution        : {{ ansible_distribution }}
    ansible_distribution_file_path : {{ ansible_distribution_file_path }}
    ansible_distribution_file_variety : {{ ansible_distribution_file_variety }}
    ansible_distribution_major_version: {{ ansible_distribution_major_version }}
    ansible_distribution_release   : {{ ansible_distribution_release }}
    ansible_distribution_version   : {{ ansible_distribution_version }}
    ansible_machine              : {{ ansible_machine }}
    ansible_memtotal_mb          : {{ ansible_memtotal_mb }}
    ansible_nodename              : {{ ansible_nodename }}
    ansible_os_family             : {{ ansible_os_family }}
    ansible_pkg_mgr               : {{ ansible_pkg_mgr }}
    ansible_processor_cores       : {{ ansible_processor_cores }}
delegate_to: 127.0.0.1
```

ファイル名：GatherFacts.yml

1.4 Playbookの登録 (2/2)

Ansible-LegacyにPlaybookを登録する

作成した作業実行用Playbookを登録します。

メニュー : Ansible-Legacy > Playbook素材集

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、 [登録] ボタンを押下する。



Playbook素材名 (任意の名称)	Playbook素材
GatherFacts	GatherFacts.yml

1.4.1 YAMLファイルと収集用ディレクトリ (1/2)

YAMLファイルを収集用のディレクトリに生成する

- ITAでは、収集結果ファイルはYAML形式に指定されているため、YAMLファイルを生成する必要があります。
- 生成されたYAMLファイルは、ITAの予約変数で指定された収集用ディレクトリに格納します。

GatherFacts.yml 7行目

```
dest: '{{ __parameter_dir_ }}/{{ inventory_hostname }}/gatherfacts.yml'
```

このディレクトリが指定されている

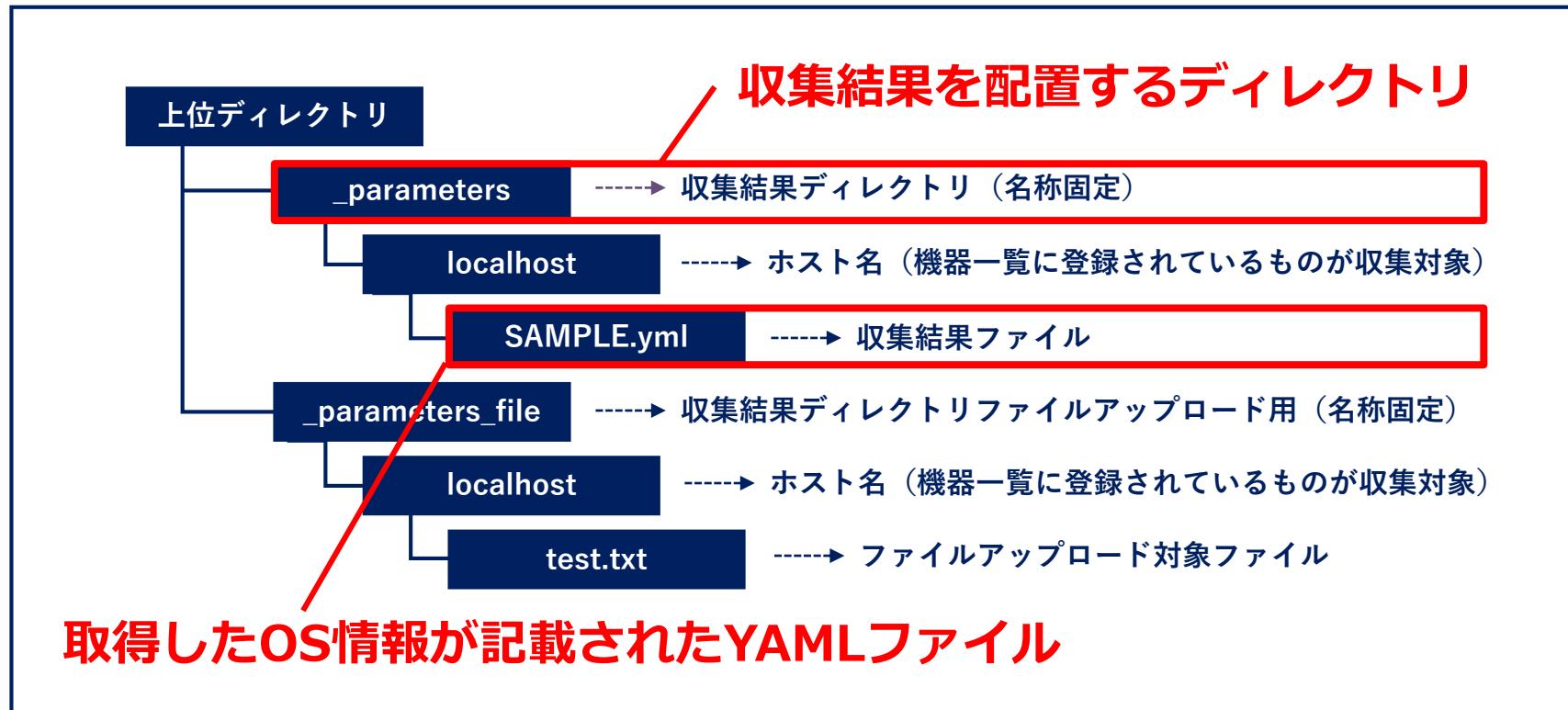
パスの変数

	ITA予約変数 ↓	変数指定内容
ソースファイルの格納先	__parameter_dir__	作業結果ディレクトリ配下の「_parameters」のパス
収集したファイルの格納先	__parameters_file_dir__	作業結果ディレクトリ配下の「_parameters_file」のパス

1.4.2 YAMLファイルと収集用ディレクトリ (2/2)

収集用のディレクトリは、下記のようなファイルの階層構造になっています。

ファイルの階層構造



1.5 Movement-Playbook紐付

MovementとPlaybookを紐付ける

登録したMovementとPlaybookを紐付けます。

メニュー : Ansible-Legacy > Movement-Playbook紐付

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、 [登録] ボタンを押下する。

紐付項目番	Movement*	Playbook素材*	インクルード順序*	アクセス権
自動入力	1:GatherFacts ▾	GatherFacts ▾	1	設定 アクセス許可ロール

Movement	Playbook素材	インクルード順序
GatherFacts	GatherFacts	1

インクルード順序とは、複数のPlaybookをMovementに紐付けた際に作業実行される順番です。
今回はPlaybookが1つだけなので1とします。

1.6 作業対象ホストの登録

■ オペレーション・Movement・作業対象ホストを紐付ける

登録したオペレーション・Movement・作業対象ホストを紐付けます。

メニュー：Ansible-Legacy > 作業対象ホスト

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

項目番号	オペレーション*	Movement*	ホスト*
7	6:GatherFacts1	4:GatherFacts	2:targethost
	オペレーション	Movement	ホスト
	GatherFacts1	GatherFacts	targethost

1.7 収集値を登録するパラメータシートの作成 (1/4)

■ 収集した値を登録するパラメータシートを作成する

[Gathered Facts] メニューを作成します。これがパラメータシートとなり、収集した値が自動登録されます。

メニュー：メニュー作成 > メニュー定義・作成

- ① 各項目で下表のように選択または入力する。
- ② [作成] ボタンを押下する。



1.7 収集値を登録するパラメータシートの作成 (2/4)

1.基本情報

メニュー名（任意の名称）	作成対象	表示順序
Gathered Facts	パラメータシート（ホスト/オペレーションあり）	1

2.対象メニューグループ

入力用	代入値自動登録用	参照用
入力用（デフォルト）	代入値自動登録用（デフォルト）	参照用（デフォルト）

3.項目

項目名（任意の名称）	入力方式	最大バイト数（任意の値）
ansible_architecture	文字列（單一行）	128
ansible_bios_version	文字列（單一行）	128
ansible_default_ipv4 > address (※)	文字列（單一行）	128
ansible_default_ipv4 > interface (※)	文字列（單一行）	128
ansible_default_ipv4 > network (※)	文字列（單一行）	128
ansible_distribution	文字列（單一行）	128
ansible_distribution_file_path	文字列（單一行）	128
ansible_distribution_file_variety	文字列（單一行）	128
ansible_distribution_major_version	文字列（單一行）	128
ansible_distribution_release	文字列（單一行）	128

1.7 収集値を登録するパラメータシートの作成 (3/4)

項目名（任意の名称）	入力方式	最大バイト数（任意の値）
ansible_distribution_version	文字列（單一行）	128
ansible_machine	文字列（單一行）	128
ansible_memtotal_mb	文字列（單一行）	128
ansible_nodename	文字列（單一行）	128
ansible_os_family	文字列（單一行）	128
ansible_pkg_mgr	文字列（單一行）	128
ansible_processor_cores	文字列（單一行）	128

※ カラムグループ作成

ansible_default_ipv4 > address
ansible_default_ipv4 > interface
ansible_default_ipv4 > network

[ansible_default_ipv4] カラムグループを作成し、その中に [address] [interface] [network] の3カラムを入れ込みます。

ansible_default_ipv4	address	interface	network
	文字列(單一行)	文字列(單一行)	文字列(單一行)
	最大バイト数* 128	最大バイト数* 128	最大バイト数* 128
	正規表現	正規表現	正規表現
	初期値	初期値	初期値
	<input type="checkbox"/> 必須 <input type="checkbox"/> 一意制約	<input type="checkbox"/> 必須 <input type="checkbox"/> 一意制約	<input type="checkbox"/> 必須 <input type="checkbox"/> 一意制約
	説明	説明	説明
	備考	備考	備考

1.7 収集値を登録するパラメータシートの作成 (4/4)



作成されたメニュー

[Gathered Facts] メニューが作成された。



[登録開始] ボタンを押下すると、作成した項目が確認できる。

The screenshot shows the 'Register' screen with three tables of collected facts:

No	パラメータ	ansible_architecture ansible_bios_version ansible_default_ipv4 address interface network
1	ansible_distribution ansible_distribution_file_path ansible_distribution_file_variety ansible_distribution_major_version	
2	ansible_machine ansible_memtotal_mb ansible_nodename ansible_os_family ansible_pkg_mgr ansible_processor_cores	

* *は必須項目です。

1.8 収集項目値管理の登録（1/3）

収集項目値管理を登録する

- 収集項目 (FROM) のYAMLファイル名・変数名と、パラメータシート (TO) のメニュー名・項目名を紐付けます。
- 下表の通り、各変数と項目を 1 セットとして登録します。

メニュー : Ansible共通 > 収集項目値管理

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、 [登録] ボタンを押下する。

収集項目 (FROM)			パラメータシート (TO)	
ID 自動入力	パース形式 YAML	PREFIX(ファイル名) gatherfacts	変数名 ansible_architectu	メンバ変数 2100011611:代入値自動登録用:2:Gathered Facts
			メニュー名 メニュー	項目 パラメータ/ansible_architecture

1.8 収集項目値管理の登録 (2/3)

収集項目 (FROM)			パラメータシート (TO)	
バース形式	PREFIX (ファイル名)	変数名	メニュー グループ :メニュー	項目
YAML	gatherfacts	ansible_architecture	代入値自動登録用: Gathered Facts	パラメータ/ ansible_architecture
YAML	gatherfacts	ansible_bios_version	代入値自動登録用: Gathered Facts	パラメータ/ ansible_bios_version
YAML	gatherfacts	ansible_default_ipv4_address	代入値自動登録用: Gathered Facts	パラメータ/ ansible_default_ipv4/address
YAML	gatherfacts	ansible_default_ipv4_interface	代入値自動登録用: Gathered Facts	パラメータ/ ansible_default_ipv4/interface
YAML	gatherfacts	ansible_default_ipv4_network	代入値自動登録用: Gathered Facts	パラメータ/ ansible_default_ipv4/network
YAML	gatherfacts	ansible_distribution	代入値自動登録用: Gathered Facts	パラメータ/ ansible_distribution
YAML	gatherfacts	ansible_distribution_file_path	代入値自動登録用: Gathered Facts	パラメータ/ ansible_distribution_file_path
YAML	gatherfacts	ansible_distribution_file_variety	代入値自動登録用: Gathered Facts	パラメータ/ ansible_distribution_file_variety
YAML	gatherfacts	ansible_distribution_major_version	代入値自動登録用: Gathered Facts	パラメータ/ ansible_distribution_major_version
YAML	gatherfacts	ansible_distribution_release	代入値自動登録用: Gathered Facts	パラメータ/ ansible_distribution_release

1.8 収集項目値管理の登録（3/3）

収集項目 (FROM)			パラメータシート (TO)	
バース形式	PREFIX (ファイル名)	変数名	メニュー グループ :メニュー	項目
YAML	gatherfacts	ansible_machine	代入値自動登録用: Gathered Facts	パラメータ/ ansible_machine
YAML	gatherfacts	ansible_memtotal_mb	代入値自動登録用: Gathered Facts	パラメータ/ ansible_memtotal_mb
YAML	gatherfacts	ansible_nodename	代入値自動登録用: Gathered Facts	パラメータ/ ansible_nodename
YAML	gatherfacts	ansible_os_family	代入値自動登録用: Gathered Facts	パラメータ/ ansible_os_family
YAML	gatherfacts	ansible_pkg_mgr	代入値自動登録用: Gathered Facts	パラメータ/ ansible_pkg_mgr
YAML	gatherfacts	ansible_processor_cores	代入値自動登録用: Gathered Facts	パラメータ/ ansible_processor_cores

1.9 収集インターフェース情報の登録

収集インターフェース情報を登録する

収集した値をITAのパラメータシートに登録する際のREST APIアクセスで必要になるため、RESTユーザー／パスワードを実行権限のあるユーザーで登録します。

メニュー： Ansible共通 > 収集インターフェース情報

- ① [フィルタ] を押下する。
- ② 「一覧」に1行だけ表示されるので、 [更新] ボタンを押して下表のように入力し、 [登録] ボタンを押下する。

履歴	更新	ID	ホスト名	IP	RESTユーザー	RESTパスワード	REST方式	プロトコル	ポート	アクセス権
履歴	更新	1	localhost	127.0.0.1	administrator	*****	IP	http	80	アクセス許可ロール
履歴	更新	1	localhost	127.0.0.1	administrator	*****	IP	http	80	アクセス許可ロール



ID	ホスト名*	IP*	RESTユーザー	RESTパスワード	REST方式*	プロトコル*	ポート*
1	localhost	127.0.0.1	administrator	*****	IP	http	80

RESTユーザー	RESTパスワード
実行権限のあるユーザー	そのユーザーのパスワード

1.10 作業実行 (1/2)

作業実行する

Movementとオペレーションを選択し、作業実行します。

メニュー : Ansible-Legacy > 作業実行

- ① Movement[一覧] から登録したMovementを選択する。
- ② オペレーション[一覧]から登録したオペレーションを選択する。
- ③ [実行] ボタンを押下する。

The screenshot shows the Ansible-Legacy interface with three main sections:

- Movement[一覧]**: A table listing Movements. One row is highlighted with a red border, showing details like MovementID 1, MovementName GatherFacts, and Ansible Legacy.
- オペレーション[フィルタ]**: A header for the Operations list.
- オペレーション[一覧]**: A table listing Operations. One row is highlighted with a red border, showing details like OperationID 1, OperationName GatherFacts1, and Ansible Legacy.

At the bottom left, there is a summary table:

MovementID	オペレーション名
1	GatherFacts

At the bottom right, there is a summary table:

Movement[一覧]	オペレーション[一覧]
GatherFacts	GatherFacts1

Buttons at the bottom include "ドライラン" (Dry Run) and "実行" (Execute), with "実行" being highlighted by a red box.

1.10 作業実行（2/2）

作業のステータスを確認する

[作業状態確認] メニューで、ステータスが完了になれば、作業完了です。

メニュー：Ansible-Legacy > 作業状態確認



The screenshot shows a software interface for managing Ansible tasks. On the left is a vertical menu bar with various options like 'Menu', 'Movement一覧', 'Playbook素材集', etc., and '作業状態確認' which is currently selected and highlighted in blue. The main window has a title bar '対象作業'. Below it is a table with several rows of task parameters. The 'Status' row is highlighted with a red box. The table columns are '項目' (Item) and '値' (Value). Other visible rows include '作業No.' (Job No.), '実行種別' (Execution Type), '実行エンジン' (Execution Engine), '呼出元Symphony' (Calling from Symphony), '呼出元Conductor' (Calling from Conductor), '実行ユーザ' (Execution User), 'ID' (ID), '名称' (Name), 'Movement' (Movement), '遅延タイム(分)' (Delay Time (min)), 'Ansible利用情報' (Ansible Usage Information), 'ホスト指定形式' (Host Specification Format), 'IP' (IP), 'オペレーション' (Operation), 'No.' (No.), and '名称' (Name).

項目	値
作業No.	133
実行種別	通常
ステータス	完了
実行エンジン	Ansible Engine
Ansible engin virtualenv パス	
呼出元Symphony	
呼出元Conductor	
実行ユーザ	システム管理者
ID	20
名称	GatherFacts
Movement	
遅延タイム(分)	
Ansible利用情報	
ホスト指定形式	IP
WinRM接続	
No.	15
オペレーション	
名称	GatherFacts1

1.11 収集結果の確認（1/2）

■ 収集状況を確認する

収集の成功/失敗を確認します。

メニュー： Ansible-Legacy > 作業管理

- ① [filtrer] ボタンを押下する。
- ② 一覧 > 収集状況 > ステータス に、以下のように表示される。
 - 収集済み : 収集成功
 - 収集済み（通知あり） : 登録/更新中に不備があった場合
 - 対象外 : 収集失敗
 - 収集エラー : 登録したオペレーションかターゲットホストの情報に不備があった場合

The screenshot shows two parts of the Ansible-Legacy interface. On the left is a table with columns: History, Job No., Operation Status Confirmation, Execution Type, Status, and Engine. The 'Status' column contains 'Completed' and 'Ansible Engine'. On the right is a detailed view of a row from the table, titled 'Collection Status'. It has columns: Status (highlighted with a red box) and Collection Log, which shows 'Completed' and 'CollectData_0000000060.log'.

履歴	作業No.◆	作業状態確認	実行種別◆	ステータス◆	実行エンジン◆
履歴	60	作業状態確認	通常	完了	Ansible Engine

収集状況

ステータス◆	収集ログ
収集済み	CollectData_0000000060.log

1.11 収集結果の確認（2/2）

パラメータシートを確認する

値がパラメータシートに登録されているか確認します。

メニュー：入力用（もしくは参照用） > Gathered Facts

- ① [filtrat] ボタンを押下する。
- ② 一覧を表示し、作成した項目に値が入っているか確認する。

履歴	複製	更新	廃止	No	ホスト名	オペレーション				パラメ		
						ID	オペレーション名	基準日時	実施予定日時	最終実行日時	ansible_architecture	ansible_bios_version
履歴	複製	更新	廃止	1	targethost	6	GatherFacts1	2021/07/28 16:47	2021/06/10 09:56	2021/07/28 16:47	x86_64	1.11.0-2.e17
-タ												
ansible_default_ipv4				ansible_distribution		ansible_distribution_file_path						
ansible_default_ipv4 > address		ansible_default_ipv4 > interface		ansible_default_ipv4 > network		CentOS						
192.0.2.1		eth0		192.0.2.0		/etc/redhat-release						
ansible_distribution_file_variety				ansible_distribution_major_version		ansible_distribution_release		ansible_distribution_version				
RedHat		7		Core		7.8						
ansible_machine		ansible_memtotal_mb		ansible_nodename		ansible_os_family		ansible_pkg_mgr		ansible_processor_cores		
x86_64		1771		demo.localdomain		RedHat		yum		1		

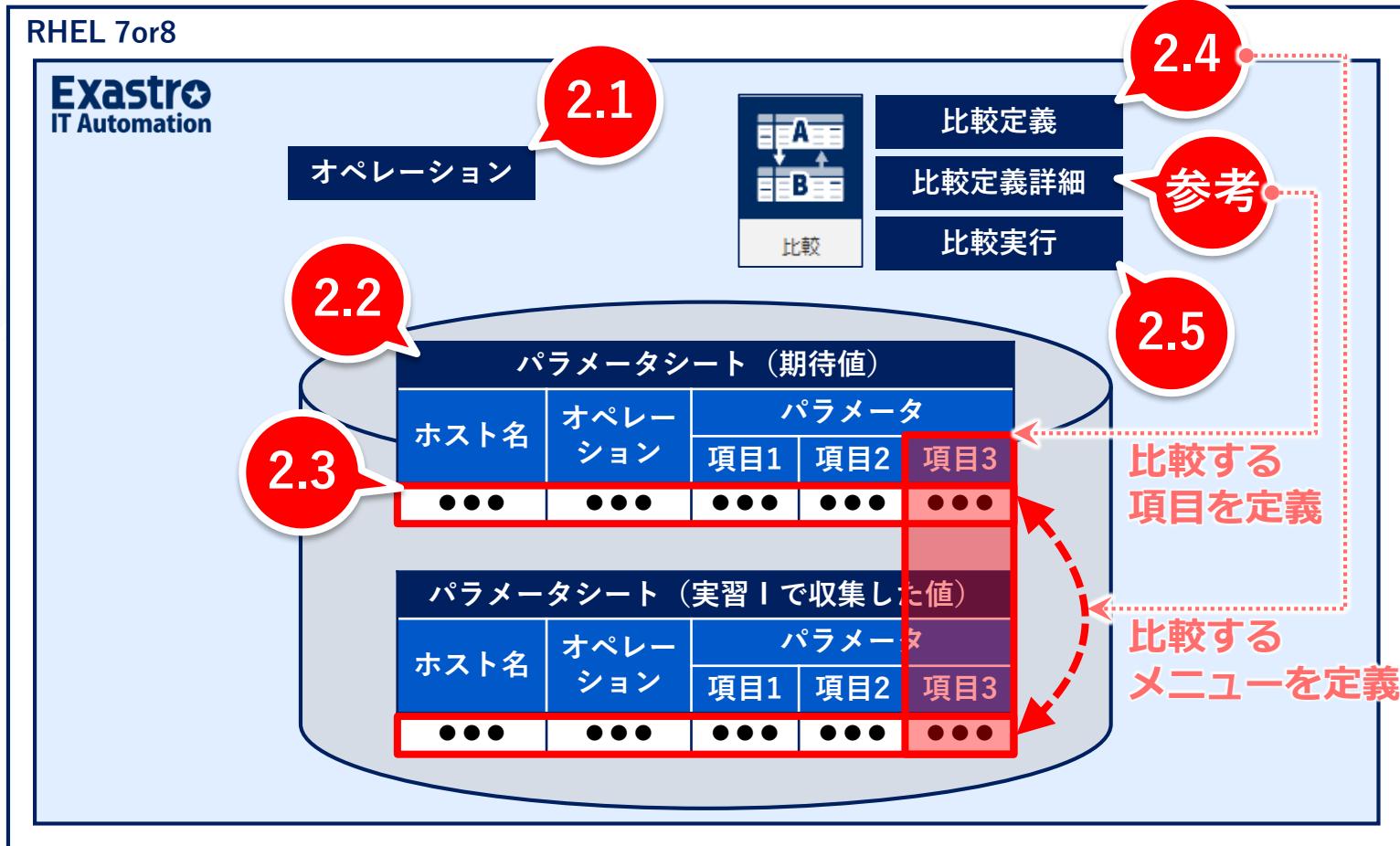
2. 実習 2 【比較機能】 実習 1 で収集した値と期待値を比較する

実習 2 全体図

実習 2 の作業の流れ

- ITAに期待値を登録し、実習 1 で収集した値と比較します。

全体図



2.1 オペレーションの登録

| オペレーションを登録する

比較用のオペレーションを登録します。

メニュー：基本コンソール > オペレーション一覧

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

No.	オペレーションID	オペレーション名*	実施予定日時*	アクセス権	
				設定	アクセス許可ロール
10	10	GatherFacts2	2021/10/01 09:25	設定	

オペレーション名 (任意の名称)	実施予定時間 (任意の時間)
GatherFacts2	2021/10/01 09:25

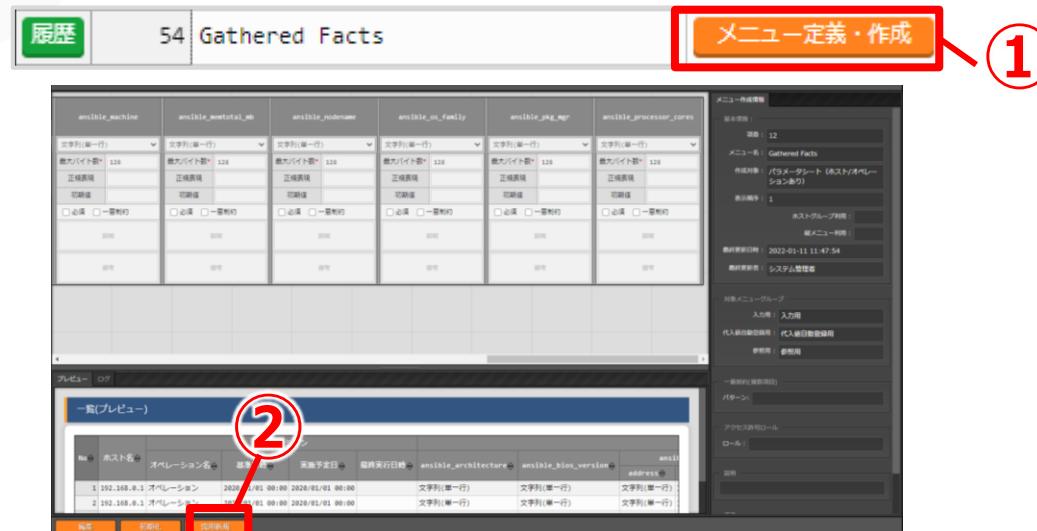
2.2 期待値用パラメータシートの作成（1/3）

期待値登録用のパラメータシートを作成する

実習1で作成した[Gathered Facts]メニューを複製し、「メニュー名」「表示順序」だけ変更します。

メニュー：メニュー作成 > メニュー定義一覧

- ① 「フィルタ」を押下して「一覧/更新」に[Gathered Facts]メニューを表示したら、「メニュー定義・作成」ボタンを押下する。
- ② メニュー定義の閲覧画面が表示されたら、[流用新規]ボタンを押下する。
- ③ 「メニュー名」「表示順序」だけ空の状態で複製されるので、下表のように入力する。
- ④ [作成]ボタンを押下する。



2.2 期待値用パラメータシートの作成 (2/3)

メニュー作成情報

メニュー名*: 3

表示順序*: 3

オペレーション

作成

メニューの表示順は任意です。

2.2 期待値用パラメータシートの作成 (3/3)



作成されたメニュー

[OS情報] メニューが作成された。

[登録開始] ボタンを押下すると、[Gathered Facts] メニューと同じ項目が確認できる。

No	ansible_architecture	ansible_bios_version	ansible_default_ipv4
自動入力			address interface network

ansible_distribution	ansible_distribution_file_path	ansible_distribution_file_variety	ansible_distribution_major_version

ansible_machine	ansible_memtotal_mb	ansible_nodename	ansible_os_family	ansible_pkg_mgr	ansible_processor_cores

2.3 期待値の登録

期待値を登録する

作成した [OS情報] メニューに期待値を登録します。実習 1 で収集した値と差分を出しておきたいので、「ansible_default_ipv4_address」だけ異なる値を入力しておきます。

メニュー：入力用 > OS情報

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。



以降省略

No	ホスト名*	オペレーション			パラメータ		
		オペレーション*	ansible_architecture	ansible_bios_version	ansible_default_ipv4 > address		
1	targethost	2021/10/01 09:25_10:GatherFacts2	i86_64	1.11.0-2.el7	192.0.2.2		

ホスト名	オペレーション	パラメータ/項目名	
		ansible_default_ipv4_address	その他の項目
targethost	GatherFacts2	実習 1 で収集した値と異なる値を入力してください	実習 1 で収集した値を入力してください

2.4 比較定義の登録

比較対象となる2つのメニューを選択する

期待値として登録した値と、実習1で収集した値を比較します。

メニュー：比較 > 比較定義

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

NO	比較定義名*	比較対象メニュー1*	比較対象メニュー2*	全件一致
2	OS情報	2100011611:代入値自動登録用:100:OS情報	2100011611:代入値自動登録用:91:Gathered Facts	<input checked="" type="radio"/>

比較定義名 (任意の名称)	比較対象メニュー1	比較対象メニュー2	全件一致
OS情報	代入値自動登録用:8:OS情報	代入値自動登録用:Gathered Facts	<input checked="" type="radio"/>

対象メニューの項目すべてを比較するので
「●」を選択する。

※ 項目を限定する場合は『【参考】 比較定義詳細』参照

2.5 比較実行 (1/2)

定義した比較を実行する

実習 1 で収集した値と期待値を比較します。

メニュー：比較 > 比較実行

- ① 比較実行 で、下表のように選択または入力し、 [比較] ボタンを押下する。
- ② 比較結果が表示される。

比較実行

比較定義 : 2:OS情報【100:OS情報 - 91:Gathered Fact... 基準日1 : 基準日2 : 対象ホスト : 選択

出力内容 : 全件出力 差分のみ

比較

比較定義	基準日 1	基準日 2	出力内容
OS情報-Gathered Facts	空欄	空欄	全件出力

「比較結果」に差分のあるレコードのみを出力する場合は「差分のみ」を選択する。

2.5 比較実行 (2/2)



比較結果が表示される

差分があれば「差分あり」と表示される。

比較項目番号	結果	リスト名	メニュー名称	No.	オペレーション名	基準日	パラメータ/ansible_architecture	パラメータ/ansible_bios_version	パラメータ/ansible_default_ipv4/ansible_default_ipv4 > address
1	差分あり	targethost OS情報	OS情報	1	GatherFacts2	2021/10/01 09:25 x86_64	1.11.0-2.el7		192.0.2.2
2	差分あり	targethost Gathered Facts 1	Gathered Facts 1		GatherFacts1	2021/07/28 16:47 x86_64	1.11.0-2.el7		192.0.2.1

Excel出力

CSV出力

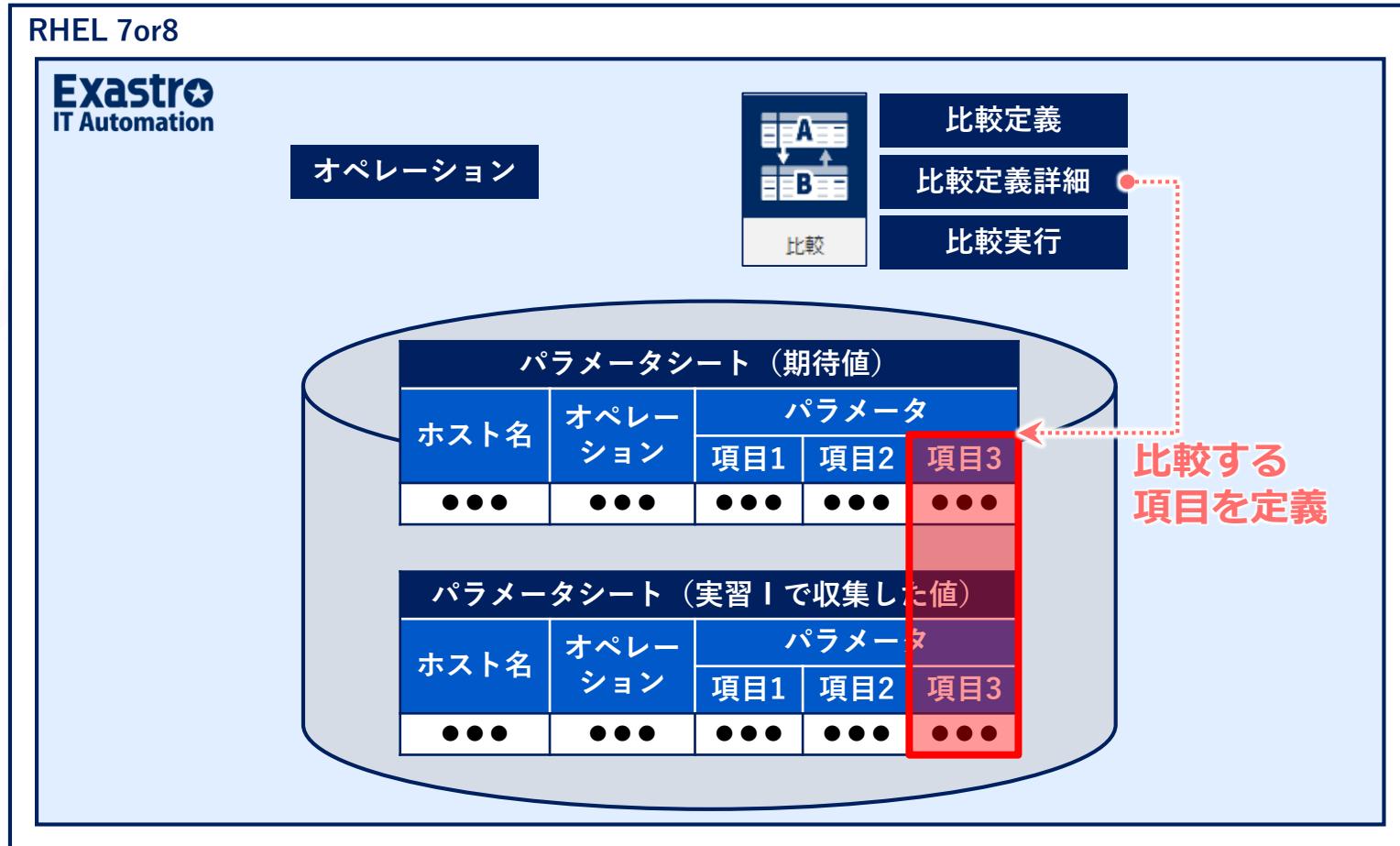
差分のあるaddressの値だけ赤色で出力される。

【参考】 比較定義詳細

パラメータシートの項目を限定して比較する

- 項目を限定して比較する場合、[比較定義詳細] メニューを利用します。

全体図



【参考】 (1) 比較定義の登録

比較対象となる2つのメニューを選択する

比較するメニューを選択します。項目を限定して比較するので「全件一致」はOFFにします。

メニュー：比較 > 比較定義

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

No	比較定義名*	比較対象メニュー1*	比較対象メニュー2*	全件一致
自動入力	IPアドレス	2100011611:代入値自動登録用:2:Gathered Facts ▼	2100011611:代入値自動登録用:8:OS情報 ▼	

※*は必須項目です。

比較定義名 (任意の名称)	比較対象メニュー1	比較対象メニュー2	全件一致
IPアドレス	代入値自動登録用:8:OS情報	代入値自動登録用:Gathered Facts	-

「●」は選択しない。

【参考】 (2) 比較定義詳細の登録

比較対象となる項目を選択する

選択したメニューの中から、比較する項目を選択します。

メニュー：比較 > 比較定義詳細

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

No	比較定義名*	表示項目名*	対象カラム1
1	IPアドレス [100:OS情報-91:Gathered Facts] ▼	IPアドレス	2100011611:代入値自動登録用:100:OS情報:129:パラメータ/ansible_default_ipv4/ansible_default_ip...

対象カラム2	表示順
2100011611:代入値自動登録用:91:Gathered Facts:109:/パラメータ/ansible_default_ipv4/ansible_defau...	1

比較定義名	表示項目名	対象カラム1	対象カラム2	表示順
IPアドレス [Gathered Facts- OS情報]	IPアドレス	代入値自動登録用: Gathered Facts:パラメータ/ ansible_default_ipv4/address	代入値自動登録用: OS情報:パラメータ/ ansible_default_ipv4/address	1

【参考】 (3) 比較実行 (1/2)

定義した比較を実行する

比較定義詳細を設定したら、比較を実行します。

メニュー：比較 > 比較実行

- ① 比較実行で、下表のように選択または入力し、[比較] ボタンを押下する。
- ② 比較結果が表示される。

比較実行

比較定義 : 3:IPアドレス【100:OS情報 - 91:Gathered … 基準日1 : 基準日2 : 対象ホスト : 選択

出力内容 : 全件出力 差分のみ

比較

比較定義	基準日1	基準日2	出力内容
IPアドレス【OS情報-Gathered Facts】	空欄	空欄	全件出力

【参考】 (3) 比較実行 (2/2)



比較結果が表示される

比較結果							
比較項目番号	結果	ホスト名	メニュー名称	No	オペレーション名	基準日	IPアドレス
1	差分あり	targethost	OS情報	1	GatherFacts2	2021/10/01 09:25	192.0.2.2
2	差分あり	targethost	Gathered Facts	1	GatherFacts1	2021/07/28 16:47	192.0.2.1

Excel出力

CSV出力

指定した項目だけ表示される。

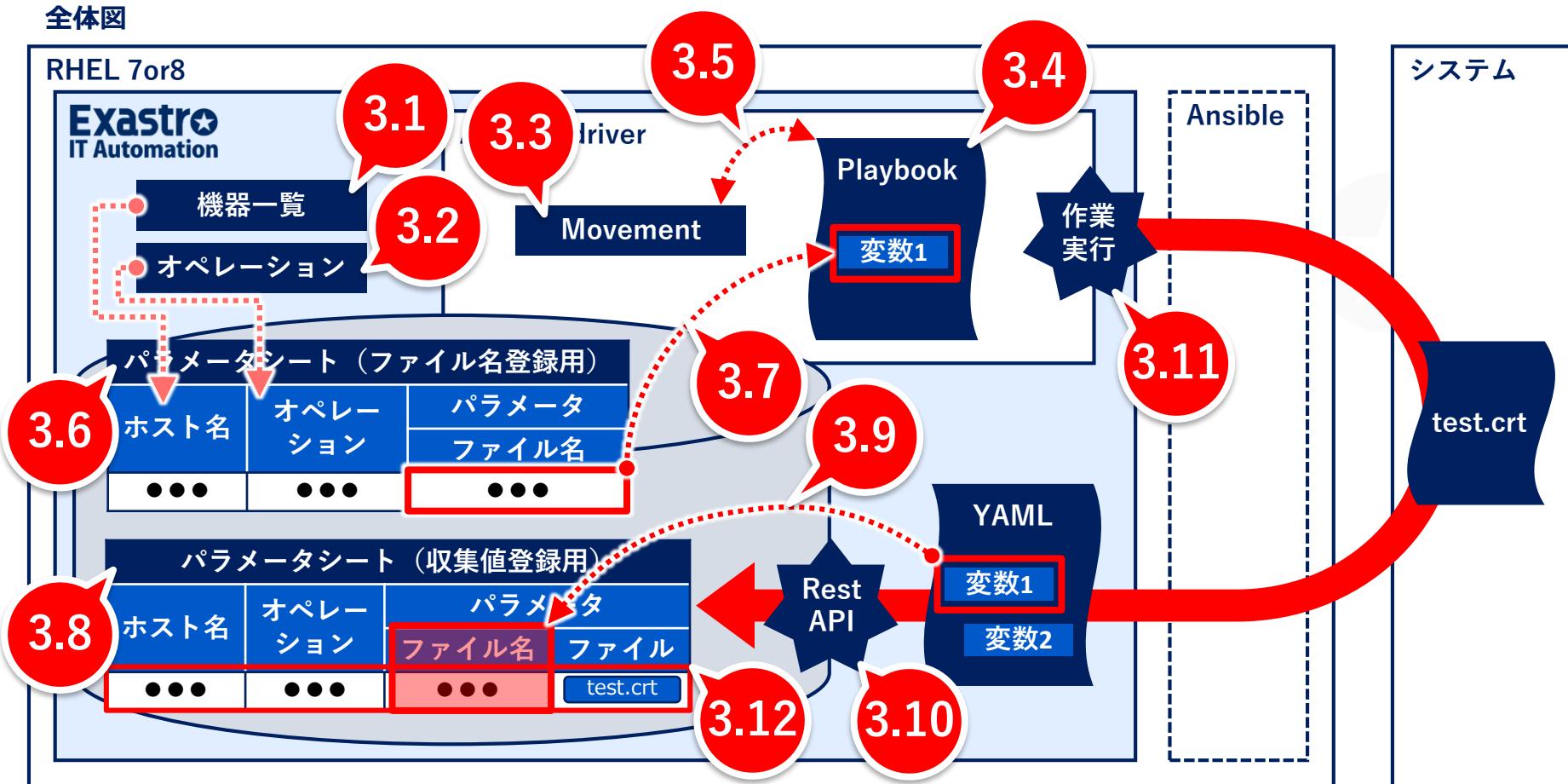
3. 実習3 【収集機能】 ターゲットホストのSSL証明書ファイルを収集する

実習 3 全体図

実習 3 の作業の流れ

- 実習 1 と全体の流れは同じですが、収集対象がファイルになります。
- パラメータシートから収集したファイルがダウンロードできるようになります。

全体図



3.1 ターゲットホストの登録

■ ターゲットホストの接続情報を登録する

- 実習 1 と同じホストを利用する場合は、この手順は不要です。

メニュー：基本コンソール > 機器一覧

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

HW 機器種別	ホスト名 (任意の名称)	IPアドレス (ご使用の環境 に応じて)	ログイン ユーザID (ご使用 の環境 に応じて)	ログインパスワード		Ansible利用情報 Legacy/Role利用 情報 認証方式
				管理	ログイン パスワード (ご使用の環境 に応じて)	
				●	*****	
SV	targethost	192.0.2.1	root	●	*****	パスワード認証

3.2 オペレーションの登録

今回のオペレーションを登録する

今回のオペレーションを登録します。

メニュー：基本コンソール > オペレーション一覧

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

No.	オペレーションID	オペレーション名*	実施予定日時*	アクセス権	
				設定	アクセス許可ロール
8	8	getSSL1	2021/04/23 17:10	設定	

オペレーション名 (任意の名称)	実施予定時間 (任意の時間)
getSSL1	2021/04/23 17:10

3.3 Movementの登録

Ansible-LegacyのMovementを登録する

この後Playbookに紐付けて、SSL証明書を取得してくるジョブとなります。

メニュー : Ansible-Legacy > Movement一覧

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

MovementID	Movement名*	遅延タイマー	Ansible利用情報	WinRM接続	ヘッダーセクション
自動入力	getSSL	IP			

Movement名 (任意の名称)	Ansible利用情報
	ホスト指定形式
getSSL	IP

3.4 Playbookの登録

作業実行用Playbookを作成する

- 「SSL証明書ファイル取得のためのYAMLファイルを作成 → SSL証明書ファイルを収集ディレクトリにコピーする」という内容です。
- ファイルの収集用ディレクトリについては、[『3.4.1 ファイルの収集用ディレクトリ』](#)を参照してください。

```
- name: make yaml file
blockinfile:
  create: yes
  mode: 644
  insertbefore: EOF
  marker: ""
  dest: "{{ __parameter_dir__ }}/{{ inventory_hostname }}/getSSL.yml"
  content: |
    SSL_file_name      : {{ VAR_ssl_name }}
    SSL_file           : {{ VAR_ssl_name }}
delegate_to: 127.0.0.1

- name: get SSL file
  fetch:
    src: /etc/pki/tls/certs/{{ VAR_ssl_name }}
    dest: "{{ __parameters_file_dir__ }}/{{ inventory_hostname }}/"
    flat: yes
```

ファイル名 : **getSSL.yml**

3.4 Playbookの登録

Ansible-LegacyにPlaybookを登録する

作成した作業実行用Playbookを登録します。

メニュー : Ansible-Legacy > Playbook素材集

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、 [登録] ボタンを押下する。



Playbook素材名 (任意の名称)	Playbook素材
getSSL	getSSL.yml

3.4.1 ファイルの収集用ディレクトリ (1/2)

■ 収集したファイルは収集用のディレクトリに格納される

- 収集したファイルは、ITAの予約変数で指定された収集用ディレクトリに格納します。

```
dest: "{{ __parameters_file_dir_ }}/{{ inventory_hostname }}/"
```

getSSL.yml
下から2行目

このディレクトリが指定されている

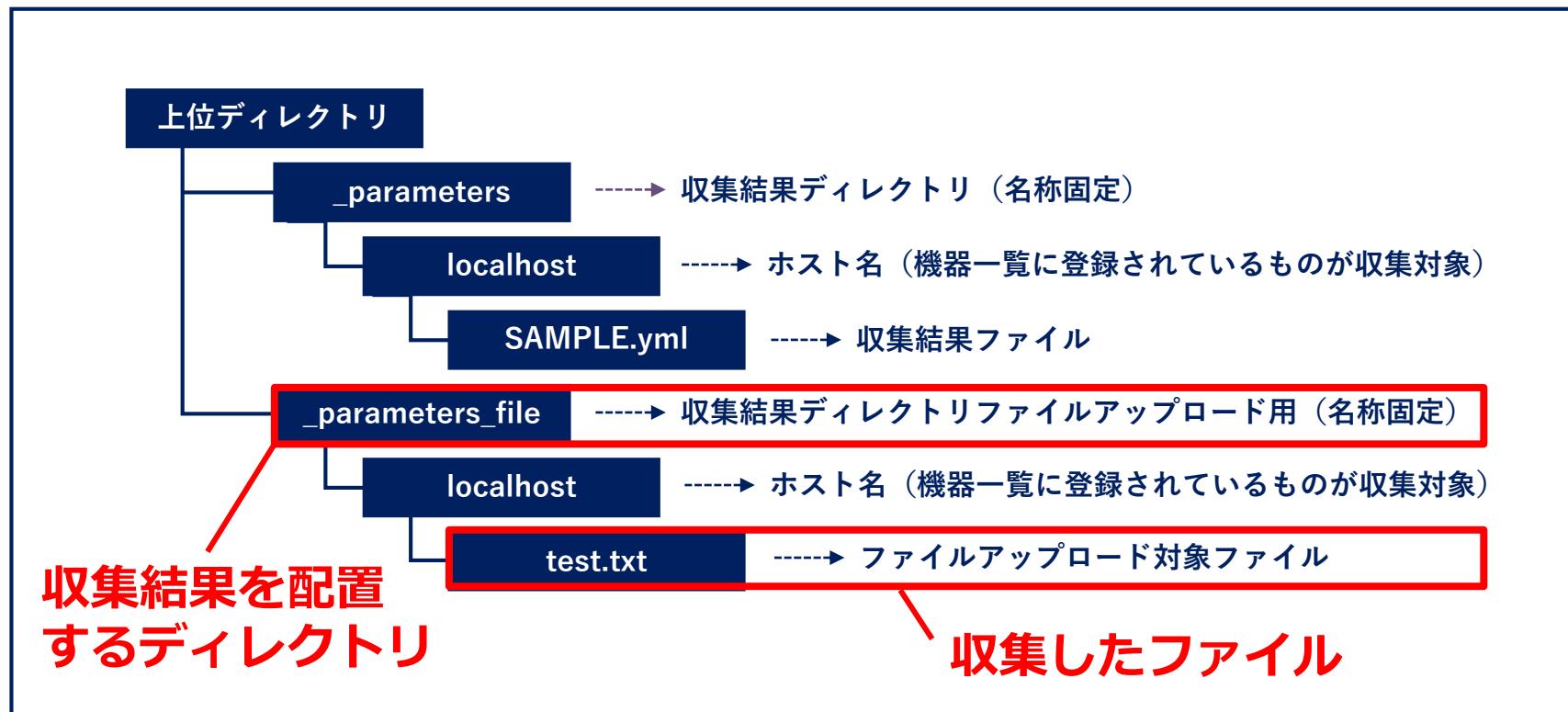
パスの変数

	ITA予約変数	変数指定内容
ソースファイルの格納先	__parameter_dir__	作業結果ディレクトリ配下の「_parameters」のパス
収集したファイルの格納先	__parameters_file_dir__	作業結果ディレクトリ配下の「_parameters_file」のパス

3.4.1 ファイルの収集用ディレクトリ (2/2)

収集用のディレクトリは、下記のようなファイルの階層構造になっています。

ファイルの階層構造



3.5 Movement-Playbook紐付

MovementとPlaybookを紐付ける

登録したMovementとPlaybookを紐付けます。

メニュー : Ansible-Legacy > Movement-Playbook紐付

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、 [登録] ボタンを押下する。

The screenshot shows a registration form for linking a Movement to a Playbook. The form has several fields:

- 紐付項目番**: A dropdown menu showing "自動入力" with "2:getSSL" selected.
- Movement***: A dropdown menu showing "getSSL".
- Playbook素材***: A dropdown menu showing "getSSL".
- インクルード順序***: An input field containing the value "1".
- アクセス権**: A section with two buttons: "設定" (highlighted in orange) and "アクセス許可ロール".

The fields **Movement***, **Playbook素材***, and **インクルード順序*** are highlighted with a red box.

Movement	Playbook素材	インクルード順序
getSSL	getSSL	1

3.6 ファイル名の登録 (1/3)

■ 具体値登録用メニューを作成する

ファイル名 (test.crt) を登録するためのパラメータシートを作成します。

メニュー：メニュー作成 > メニュー定義・作成

- ① 各項目で下表のように選択または入力する。
- ② [作成] ボタンを押下する。

1. 基本情報

メニュー名（任意の名称）	作成対象	表示順序
SSL証明書名	パラメータシート（ホスト/オペレーションあり）	4

2. 対象メニューグループ

入力用	代入値自動登録用	参照用
入力用（デフォルト）	代入値自動登録用（デフォルト）	参照用（デフォルト）

3. 項目

項目名（任意の名称）	入力方式	最大バイト数（任意の値）
ファイル名	文字列（單一行）	128

3.6 ファイル名の登録 (2/3)



作成されたメニュー

[SSL証明書名] メニューが作成された。



[登録開始] ボタンを押下すると、作成した項目が確認できる。

No	ホスト名*	オペレーション	オペレーション*	パラメータ	ファイル名
自動入力	<input type="text"/>				

* *は必須項目です。

戻る 登録

3.6 ファイル名の登録 (3/3)

■ ファイル名を登録する

作成したパラメータシートにファイル名 (test.crt) を登録します。

メニュー : 入力用 > SSL証明書名

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、 [登録] ボタンを押下する。

No	ホスト名*	オペレーション	パラメータ ファイル名
1	targethost	2021/04/23 17:10_8:getSSL1	test.crt

ホスト名	オペレーション	パラメータ ファイル名
targethost	getSSL1	test.crt

3.7 代入値自動登録設定の登録

代入値自動登録設定を登録する

- 収集するファイル名（具体値）を、Playbook内の変数に紐づけます。 ([ファイル名: test.crt] [変数名: VAR_ssl_name])
- ファイル名を別のパラメータシートに登録しておき（[『3.6 ファイル名の登録』](#)で登録済み）、代入値自動登録設定でパラメータシートの項目名とPlaybook内の変数を紐づけておくと、変数の具体値が自動的に設定されます。

メニュー : Ansible-Legacy > 代入値自動登録設定

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

パラメータシート (From)		登録方式	IaC変数 (To)	
メニュー グループ : メニュー	項目		Movement	Value変数 変数名
代入値自動登録用 : SSL証明書名	パラメータ/ ファイル名	Value型	getSSL	VAR_ssl_name

3.8 収集値を登録するパラメータシートの作成（1/3）

■ 収集した値を登録するパラメータシートを作成する

- [SSL証明書] メニューを作成します。
- 「ファイル名」「ファイル」の2つの項目を設置します。「ファイル」からは収集したファイルがダウンロードできるようになります。

メニュー：メニュー作成 > メニュー定義・作成

- ① 各項目で下表のように選択または入力する。
- ② [作成] ボタンを押下する。

1. 基本情報

メニュー名（任意の名称）	作成対象	表示順序
SSL証明書	パラメータシート（ホスト/オペレーションあり）	2

2. 対象メニューグループ

入力用	代入値自動登録用	参照用
入力用（デフォルト）	代入値自動登録用（デフォルト）	参照用（デフォルト）

3.8 収集値を登録するパラメータシートの作成 (2/3)

3.項目

項目名（任意の名称）	入力方式	最大バイト数（任意の値）
ファイル名	文字列（單一行）	128
ファイル	ファイルアップロード	1000000

「項目」は以下を入力

The screenshot shows a configuration dialog for a parameter. The fields are annotated as follows:

- 項目名 (Item Name): Points to the "項目名" field at the top left.
- 入力方式 (Input Method): Points to the "入力方式" dropdown menu at the top right.
- 最大バイト数 (Max Bytes): Points to the "最大バイト数" input field containing "128".
- 項目名 (Item Name): Points to the "項目名" field at the top right.
- 入力方式 (Input Method): Points to the "入力方式" dropdown menu at the top left.
- ファイル最大バイト数 (File Max Bytes): Points to the "ファイル最大バイト数" input field containing "1000000".

3.8 収集値を登録するパラメータシートの作成 (3/3)



作成されたメニュー

[SSL証明書] メニューが
作成された。

The screenshot shows the application's main menu on the left and a dashboard view on the right. The 'SSL証明書' menu item is highlighted with a red box. On the dashboard, there are three buttons at the bottom labeled '入力用' (Input), '代入値自動…' (Automatic value assignment…), and '参照用' (Reference). The '入力用' button is also highlighted with a red box.

[登録開始] ボタンを押下すると、
作成した項目が確認できる。

The screenshot shows a registration form titled '登録'. The form has columns for 'No.', 'ホスト名*', 'オペレーション*', 'オペレーション*', 'ファイル名', and 'ファイル'. The 'ファイル名' and 'ファイル' columns are grouped together and highlighted with a red box. A message at the bottom states 'ファイルを選択' (Select file) and '事前アップロード' (Pre-upload).

3.9 収集項目値管理の登録

収集項目値管理を登録する

- 収集した値をパラメータシートに自動登録するよう設定します。
- 収集項目 (FROM) のYAMLファイル名・変数名と、パラメータシート (TO) のメニュー名・項目名を紐付けます。「ファイル名」「ファイル」の2つ分登録します。

メニュー : Ansible共通 > 収集項目値管理

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、 [登録] ボタンを押下する。

収集項目 (FROM)

パラメータシート (TO)

収集項目 (FROM)			パラメータシート (TO)	
ID	パース形式*	PREFIX(ファイル名)	変数名*	メンバ変数
自動入力	YAML ▼	getSSL	SSL_file_name	メニュー:メニュー 2100011611:代入値自動登録用:5:SSL証明書 ▼
				項目 パラメータ/ファイル名 ▼

収集項目 (FROM)

パラメータシート (TO)

パース形式	PREFIX(ファイル名)	変数名	メニュー:メニュー	項目
YAML	getSSL	SSL_file_name	代入値自動登録用:SSL証明書	パラメータ/ファイル名
YAML	getSSL	SSL_file	代入値自動登録用:SSL証明書	パラメータ/ファイル

3.10 収集インターフェース情報の登録

収集インターフェース情報を登録する

- 収集した値をITAのパラメータシートに登録する際のREST APIアクセスで必要になるため、RESTユーザー／パスワードを実行権限のあるユーザーで登録します。
- 実習1と同じRESTユーザーを使用する場合は、この手順は不要です。

メニュー： Ansible共通 > 収集インターフェース情報

- ① [フィルタ] を押下する。
- ② 「一覧」に1行だけ表示されるので、 [更新] ボタンを押して下表のように入力し、 [登録] ボタンを押下する。

The screenshot shows the 'Ansible共通 > 収集インターフェース情報' screen. At the top, there's a table with columns: 履歴, 更新, ID*, ホスト名*, IP*, RESTユーザー*, REST/パスワード, REST方式*, プロトコル*, ポート*. A row is selected with ID 1, Hostname 'localhost', IP '127.0.0.1', User 'administrator', and Password '*****'. The '更新' (Update) button in the second column is highlighted with a red box. A large blue arrow points down to a detailed update form below.

ID	ホスト名*	IP*	RESTユーザー	RESTパスワード	REST方式*	プロトコル*	ポート*
1	localhost	127.0.0.1	administrator	*****	IP	http	80

Below the table is a detailed update form:

ID	ホスト名*	IP*	RESTユーザー	RESTパスワード	RESTユーザー	RESTパスワード
1	localhost	127.0.0.1	administrator	*****	実行権限のあるユーザー	そのユーザーのパスワード

3.11 作業実行 (1/2)

作業実行する

Movementとオペレーションを選択し、作業実行します。

メニュー : Ansible-Legacy > 作業実行

- ① Movement[一覧] から登録したMovementを選択する。
- ② オペレーション[一覧]から登録したオペレーションを選択する。
- ③ [実行] ボタンを押下する。

The screenshot shows the Exastro UI interface for selecting a movement and operation.

Movement[一覧] (Movement List):

選択	MovementID	Movement名	オーケストレータ	遅延タイマー	Ansible利用情報	アクセス権	最終更新日時	最終更新者
<input type="radio"/>	1	GatherFacts	Ansible Legacy	IP	- hosts: all remote_user: "{{ __loginuser__ }}" gather_facts: yes		2021/04/07 10:48:18	システム管理者
<input checked="" type="radio"/>	2	getSSL	Ansible Legacy	IP			2021/04/08 14:23:18	システム管理者

フィルタ結果件数: 2

オペレーション[フィルタ] (Operation Filter):

オペレーション[一覧] (Operation List):

選択	No.番	オペレーションID	オペレーション名	実施予定期	最終実行日時	アクセス権	最終更新日時	最終更新者
<input type="radio"/>	1	1	GatherFacts1	2021/04/22 17:09	2021/04/14 09:03	アクセス許可	2021/04/14 09:03:28	legacy作業実行プロシージャ
<input checked="" type="radio"/>	2	2	getSSL1	2021/04/23 17:10	2021/04/13 11:37	アクセス許可	2021/04/13 11:37:27	legacy作業実行プロシージャ
<input type="radio"/>	3	3	GatherFacts2	2021/04/16 12:19	2021/04/07 15:51	アクセス許可	2021/04/07 15:51:51	legacy作業実行プロシージャ
<input type="radio"/>	4	4	getSSL2	2021/04/28 12:19	2021/04/13 12:42	アクセス許可	2021/04/13 12:42:21	legacy作業実行プロシージャ
<input type="radio"/>	5	5	GatherFacts101	2021/10/01 09:25		アクセス許可	2021/04/13 09:25:48	システム管理者

フィルタ結果件数: 5

Movement[一覧] | **オペレーション[一覧]**

getSSL	getSSL1
--------	---------

MovementID: 2
Movement名: getSSL

ドライラン 実行

3.11 作業実行（2/2）

作業のステータスを確認する

[作業状態確認] メニューで、ステータスが完了になれば、作業完了です。

メニュー：Ansible-Legacy > 作業状態確認

The screenshot shows the Ansible-Legacy application interface. On the left is a vertical menu bar with the following items:

- メインメニュー
- Movement一覧
- Playbook素材集
- Movement-Playbook紐付
- 変数名一覧
- 代入値自動登録設定
- 作業対象ホスト
- 代入値管理
- 作業実行
- 作業状態確認
- 作業管理

The '作業状態確認' item is highlighted in red. The main content area is titled '対象作業' (Target Job). It displays a table of job details:

項目	値	
作業No.	135	
実行種別	通常	
ステータス	完了	
実行エンジン	Ansible Engine	
Ansible engine virtualenv パス		
呼出元 Symphony		
呼出元 Conductor		
実行ユーザ	システム管理者	
Movement	ID	21
	名称	getSSL
	遅延タイマ(分)	
Ansible利用情報	ホスト指定形式	IP
	WinRM接続	
オペレーション	No.	17
	名称	getSSL1
	ID	17

3.12 収集結果の確認（1/2）

■ 収集状況を確認する

収集の成功/失敗を確認します。

メニュー： **Ansible-Legacy > 作業管理**

- ① [filtrer] ボタンを押下する。
- ② 一覧 > 収集状況 > ステータス に、以下のように表示される。
 - ・ 対象外 : 収集失敗
 - ・ 収集済み : 収集成功
 - ・ 収集済み（通知あり） : 登録/更新中に不備があった場合
 - ・ 収集エラー : 登録したオペレーションかターゲットホストの情報に不備があった場合

履歴	作業No. ▾	作業状態確認	実行種別 ▾	ステータス ▾
履歴	8	作業状態確認	通常	完了

収集状況	
ステータス ▾	収集ログ
収集済み	CollectData_0000000008.log

3.12 収集結果の確認（2/2）

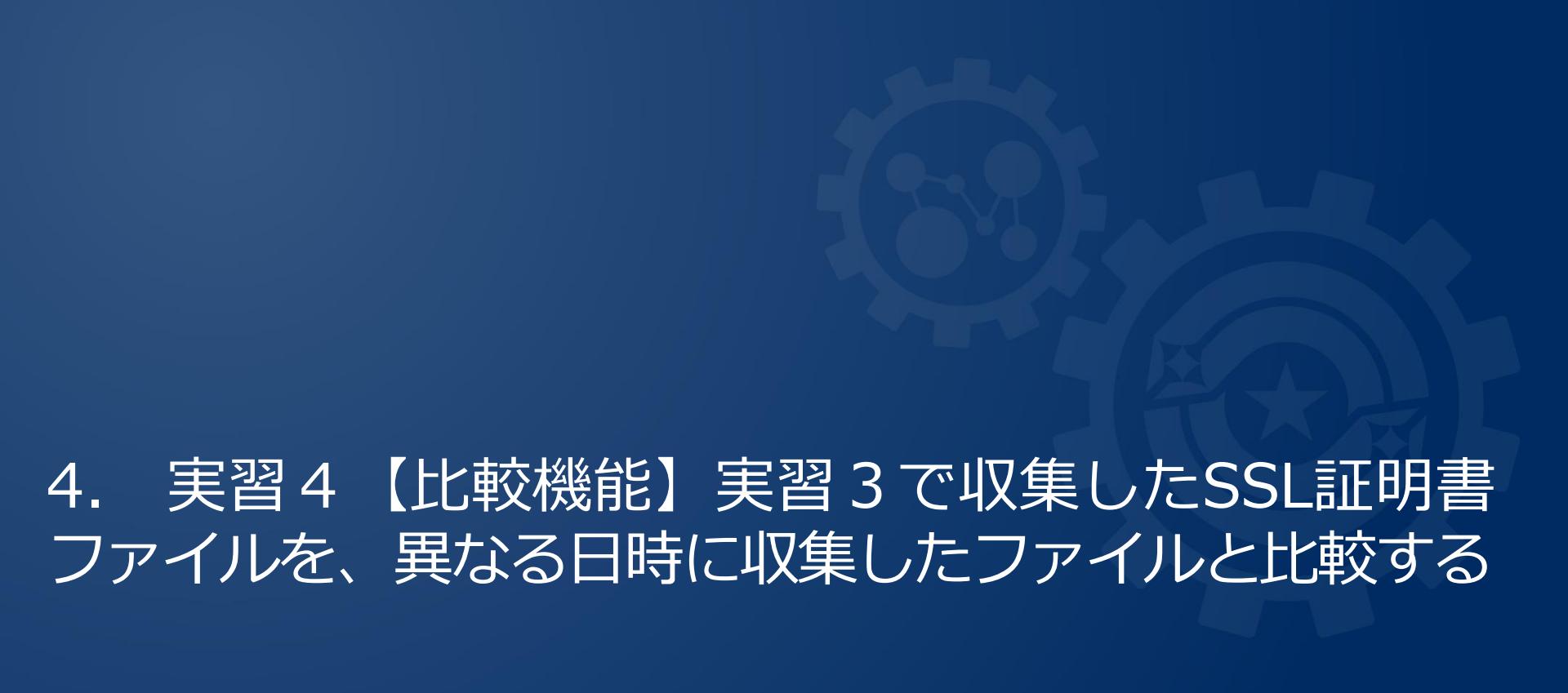
■ パラメータシートを確認する

収集した値がパラメータシートに登録され、ファイルがダウンロードできるか確認します。

メニュー： 入力用（もしくは代入値自動登録用） > SSL証明書

- ① [filtrat] ボタンを押下する。
- ② 一覧を表示し、作成した項目に値が入っているか確認する。

履歴	複製	更新	廃止	No	ホスト名	オペレーション					パラメータ	
						ID	オペレーション名	基準日時	実施予定日時	最終実行日時	ファイル名	ファイル
履歴	複製	更新	廃止	1	targethost	8	getSSL1	2021/07/28 11:25	2021/04/23 17:10	2021/07/28 11:25	test.crt	test.crt



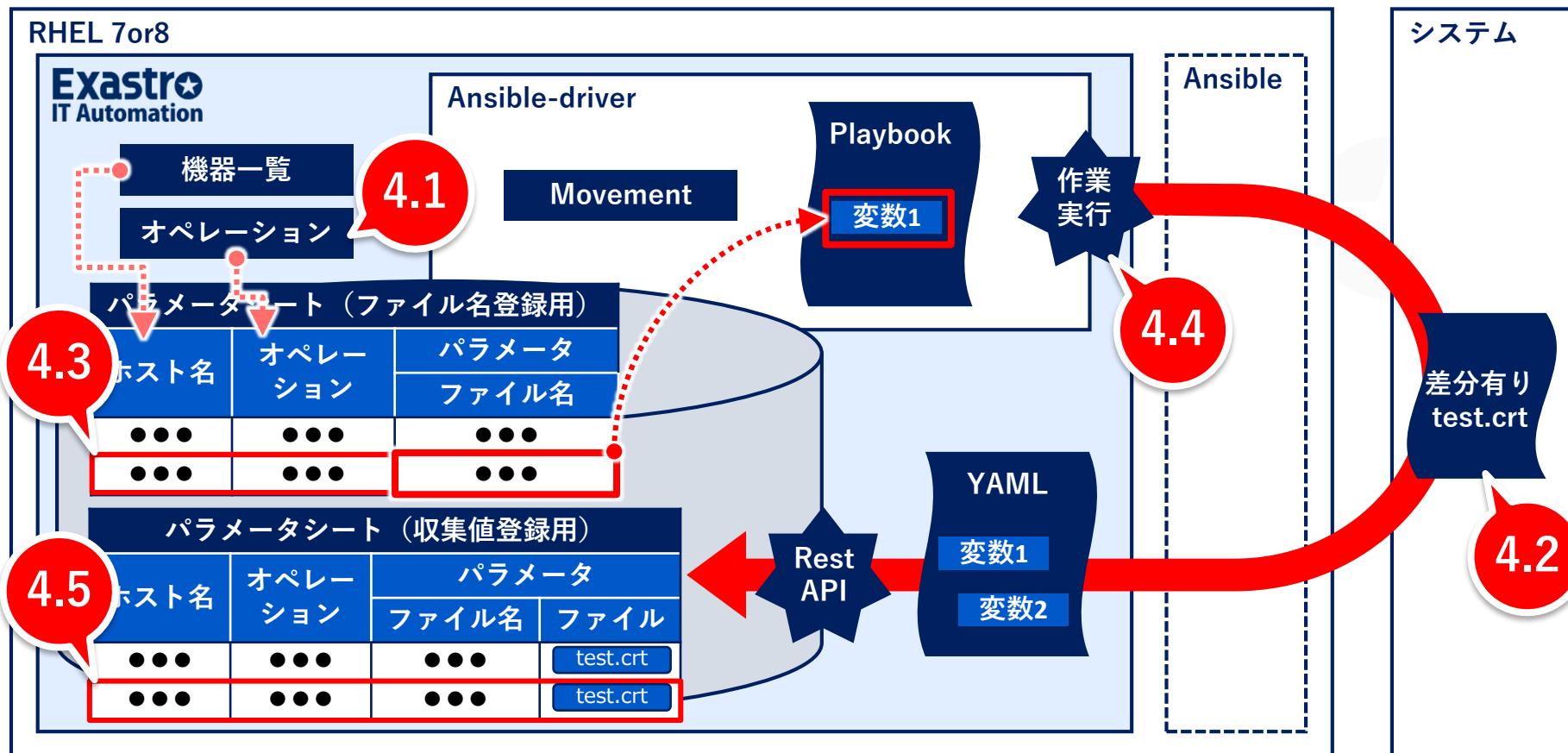
4. 実習4【比較機能】実習3で収集したSSL証明書
ファイルを、異なる日時に収集したファイルと比較する

実習 4 全体図 (1/2)

実習 4 の作業の流れ

- 実習 3 で収集したSSL証明書ファイルとは内容と収集日時が異なるSSL証明書ファイルを収集し、両ファイルを比較します。

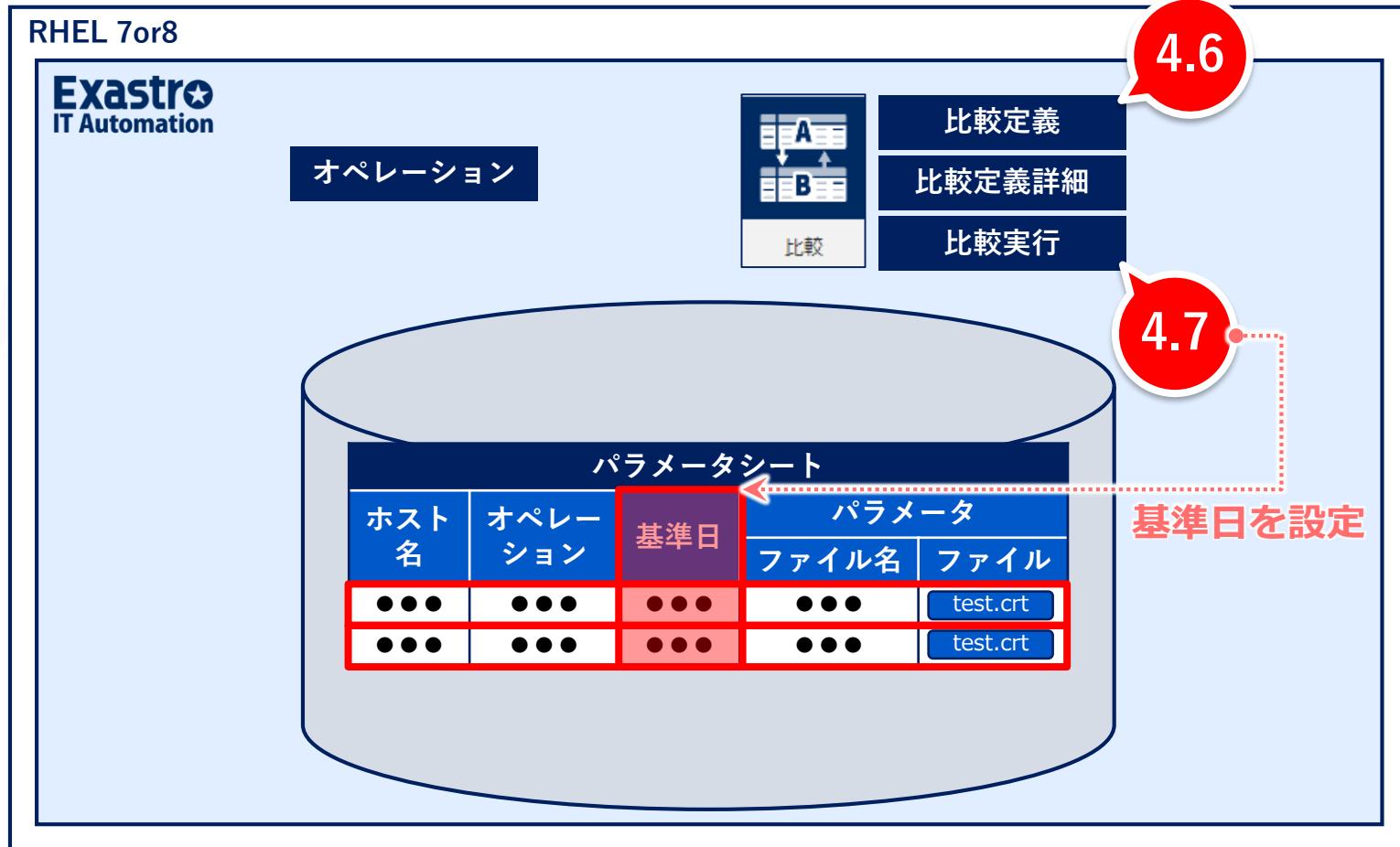
全体図



実習 4 全体図 (2/2)

- 「同一メニューだが収集した日時の異なる値」を比較する際は、比較実行の際に「基準日」を設定します。

全体図（比較）



4.1 オペレーションの登録

| オペレーションを登録する

比較用のオペレーションを登録します。

メニュー：基本コンソール > オペレーション一覧

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

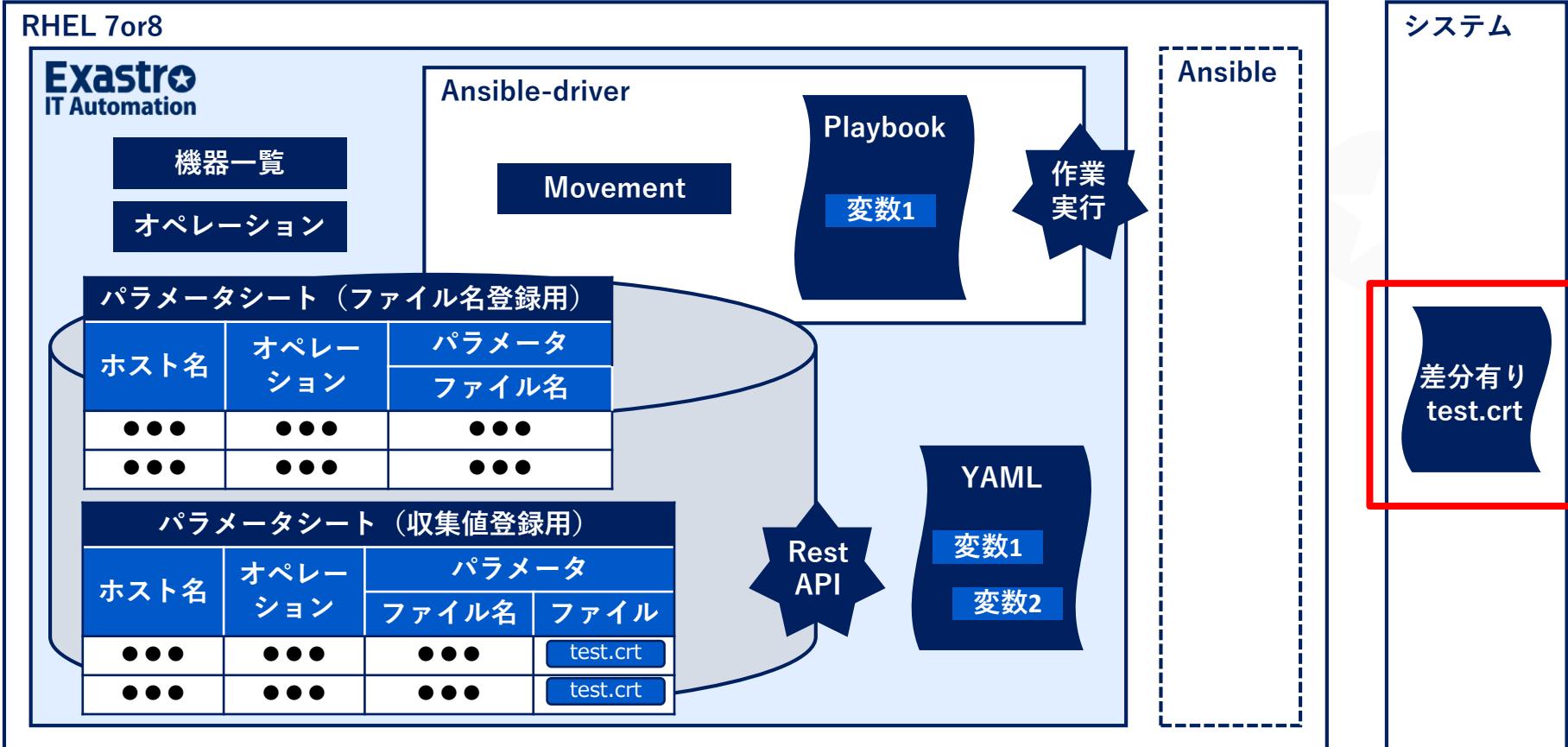
No.	オペレーションID	オペレーション名*	実施予定日時*	アクセス権
自動入力	自動入力	getSSL2	2021/04/28 12:19	<button>設定</button> アクセス許可ロール

オペレーション名 (任意の名称)	実施予定時間 (任意の時間)
getSSL2	2021/04/28 12:19

4.2 差分有りSSL証明書の用意

差分有りSSL証明書を用意しておく

- この実習では「差分あり」の表示を確認したいので、実習3で収集したファイルと差分のあるファイルを収集します。
- ターゲットサーバの/etc/pki/tls/certs/に入ってSSL証明書(test.crt)の中身を一部書き換えておきます。



4.3 ファイル名の登録

■ ファイル名を登録する

- 実習 3 で作成した [SSL証明書名] メニューに比較用のレコードを登録します。
- オペレーションのみ比較用にし、そのほかは実習 3 と同じ内容です。

メニュー：入力用 > SSL証明書名

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

No	ホスト名*	オペレーション	パラメータ ファイル名
2	targethost	2021/04/28 12:19_9:getSSL2	test.crt

ホスト名	オペレーション	パラメータ ファイル名
targethost	getSSL2	test.crt

4.4 作業実行

作業実行する

- 差分有りのSSL証明書を収集します。
- Movementは実習3と同じくgetSSLで、オペレーションだけ比較用のgetSSL2にします。

メニュー : Ansible-Legacy > 作業実行

- ① Movement[一覧] から登録したMovementを選択する。
- ② オペレーション[一覧]から登録したオペレーションを選択する。
- ③ [実行] ボタンを押下する。

The screenshot shows the Exastro interface with the following components:

- Movement[一覧]**: A list of Movements. One entry is selected: "2 getSSL" (Ansible Legacy). This row is highlighted with a red box.
- オペレーション[フィルタ]**: A list of Operations. One entry is selected: "4 getSSL2" (Ansible Legacy). This row is highlighted with a red box.
- Movement[一覧]**: A detailed view of the selected Movement. It shows the Movement ID (2), Movement Name (Ansible Legacy), hosts (IP), and Ansible利用情報 (gather_facts: yes, become: yes). The "最終更新日時" (Last updated) field shows "2021/04/08 14:23:18".
- オペレーション[一覧]**: A detailed view of the selected Operation. It shows the Operation ID (4), Operation Name (getSSL2), hosts (IP), and Ansible利用情報 (gather_facts: yes, become: yes). The "最終更新日時" (Last updated) field shows "2021/04/13 12:42:21".
- 実行**: An orange button at the bottom left of the dialog, which is highlighted with a red box.
- Movement[一覧]**: A summary section on the right showing the Movement ID (2) and Movement Name (Ansible Legacy).
- オペレーション[一覧]**: A summary section on the right showing the Operation ID (4) and Operation Name (getSSL2).

4.5 収集結果の確認

パラメータシートを確認する

- getSSL2がパラメータシートに収集されているか確認します。
- [4.7 比較実行](#)で必要になるので、基準日時を確認しておきます。

メニュー： 入力用（もしくは参照用） > SSL証明書

- [filtrat] ボタンを押下する。
- 一覧を表示し、作成した項目に値が入っているか確認する。

一覧/更新												
履歴	複製	更新	廃止	No	ホスト名	オペレーション					パラメタ	
						ID	オペレーション名	基準日時	実施予定日時	最終実行日時	ファイル名	ファイル
履歴	複製	更新	廃止	1	targethost	8	getSSL1	2021/07/28 11:25	2021/04/23 17:10	2021/07/28 11:25	test.crt	test.crt
履歴	複製	更新	廃止	2	targethost	9	getSSL2	2021/07/28 12:08	2021/04/28 12:19	2021/07/28 12:08	test.crt	test.crt

フィルタ結果件数: 2

4.6 比較定義の登録

比較対象となる2つのメニューを選択する

「同一メニューだが収集した日時の異なる値」を比較するので、比較対象メニューは同じものを選択します。

メニュー：比較 > 比較定義

- ① 登録 > 登録開始を押下する。
- ② 各項目で下表のように選択または入力し、[登録] ボタンを押下する。

No	比較定義名*	比較対象メニュー1*	比較対象メニュー2*	全件一致
自動入力	SSL証明書	2100011611:代入値自動登録用:5:SSL証明書	2100011611:代入値自動登録用:5:SSL証明書	●

比較定義名 (任意の名称)	比較対象メニュー1	比較対象メニュー2	全件一致
SSL証明書	代入値自動登録用:SSL証明書	代入値自動登録用:SSL証明書	●

4.7 比較実行 (1/3)

定義した比較を実行する

- 登録した比較定義「SSL証明書」を選択し、基準日を入力します。
- 基準日は、SSL1と2それぞれが最新となる日時を設定します。
- 基準日については次ページを参照してください。

メニュー：比較 > 比較実行

- 比較実行で、下表のように選択または入力し、[比較] ボタンを押下する。
- 比較結果が表示される。

比較実行

比較定義 : 3:IPアドレス 【 100:OS情報 - 91:Gathered … 】 基準日1 : 2021/07/28 11:30 基準日2 : 2021/07/28 12:30 対象ホスト : 選択

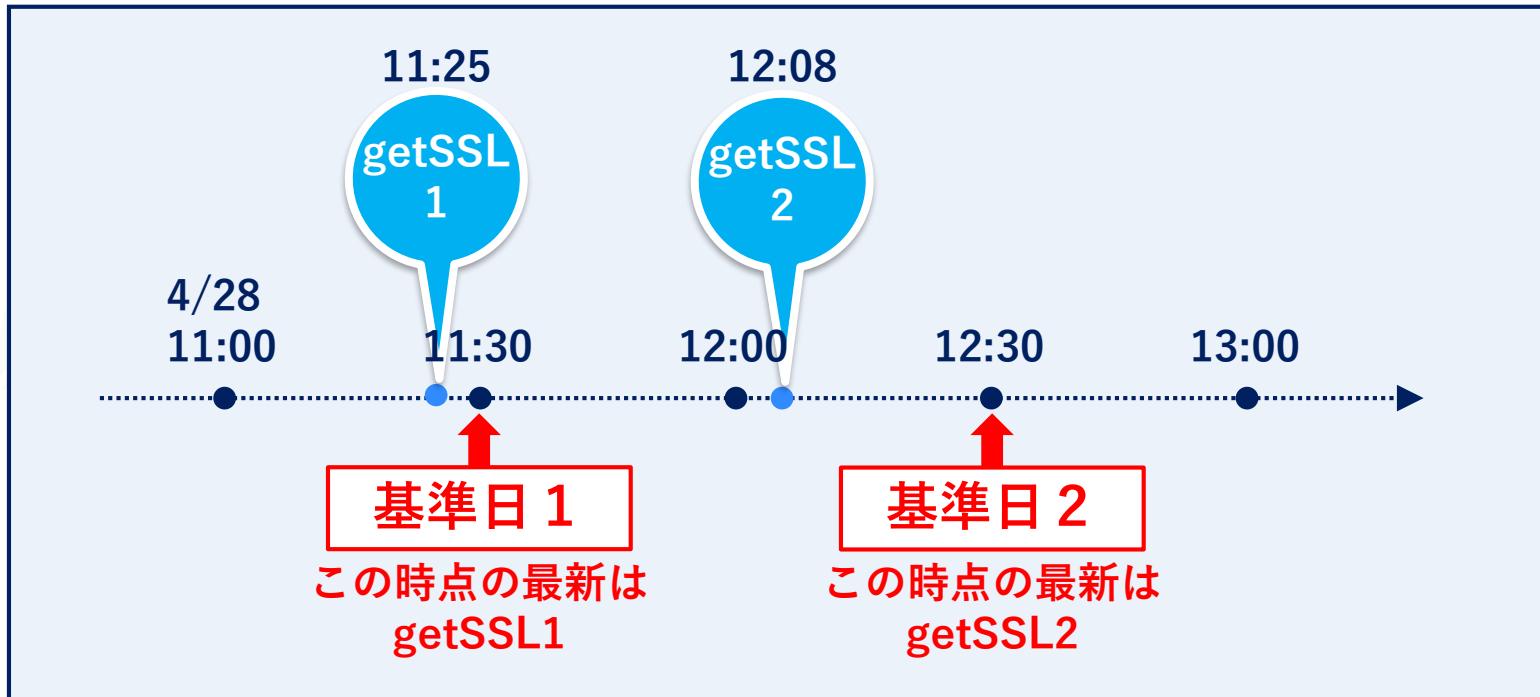
出力内容 : 全件出力 差分のみ

比較

比較定義	基準日 1	基準日 2	出力内容
SSL証明書	2021/7/28 11:30	2021/7/28 12:30	全件出力

4.7 比較実行 (2/3)

- 基準日の設定は以下のようになります。



収集の基準日に応じて比較の基準日を設定する。

4.7 比較実行 (3/3)



比較結果が表示される

比較結果

比較項目	結果	ホスト名	メニュー名称	No	オペレーション名	基準日	パラメータ/ファイル名	パラメータ/ファイル
1	差分あり	targethost	SSL証明書	1	getSSL1	2021/07/28 11:25	test.crt	test.crt
2	差分あり	targethost	SSL証明書	2	getSSL2	2021/07/28 12:08	test.crt	test.crt

Excel出力

CSV出力

差分があれば赤字で出力される。



Exastro