



利用手順マニュアル

astroll_Ansible系ドライバー共通

—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。
- ・ AnsibleTower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	3
1 変数の取り扱い	5
1.1 変数の種類	5
1.2 変数の抽出しおよび具体値登録	7
1.3 代入値登録による変数の扱い	7
2 機能・操作方法説明	8
2.1 Movement 一覧	8
2.2 テンプレート管理	10
2.3 ファイル管理	12
2.4 ロールパッケージ管理	14
2.5 多段変数最大繰返数	15
2.6 代入値自動登録設定	16
2.7 作業対象ホスト	19
2.8 代入値管理	20
3 構築コード記述方法	23
3.1 Playbook の記述	23
3.2 ロールパッケージの記述	24
3.3 astroll readme の記述	27
3.4 読替表の記述	29
3.5 BackYard コンテンツ	31
3.6 Ansible 利用ガイドライン astroll 追加ルール	33
4 トラブルシューティング	34

はじめに

本書では、以下の本書の対象ドライバーにおいて、共通している機能および操作方法について説明します。

本書の対象ドライバー

- ・ astroll Ansible driver (以降、Ansible driver と略)
- ・ astroll Ansible Tower/AWX driver (以降、Ansible Tower/AWX driver と略)

なお、本書では以降の説明中で『対象ドライバー』と記述している箇所については、上記のドライバーを指すものとします。

前提条件

- ・ 前提条件については、対象ドライバーのそれぞれの利用手順マニュアルをご参照ください。

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。

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	【本書】 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	利用手順マニュアル astroll Ansible 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 astroll Ansible 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。

No.	マニュアル名	説明
9	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。
10	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。

1 変数の取り扱い

1.1 変数の種類

対象ドライバーでは、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。

※設定方法の詳細は、対象ドライバーのそれぞれの利用手順マニュアルにおいて「代入値管理」についての説明を記載しておりますので、それらをご参照ください。

Playbook 中の変数で、astroll の変数として扱える変数は以下の 6 種類があります。

種類	内容	ドライバー別 対応状況（「○」ならば対応）			
		Ansible			Tower / AWX
		Legacy	Pioneer	Legacy Role	
通常変数	変数名に対して具体値を 1 つ定義できる変数です。 Playbook 内の変数は {{△VAR_xxx△}} で記述してください。 △:半角スペース xxx: 半角英数字とアンダースコア(_) e.g.) VAR_users: root	○	○	○	○
複数具体値変数	変数名に対して具体値を複数定義できる変数です。 Playbook 内の変数は {{△VAR_xxx△}} で記述してください。 △:半角スペース xxx: 半角英数字とアンダースコア(_) e.g.) VAR_users: - root - mysql	○	○	○	○
多段変数	階層化された変数です。 Playbook 内の変数は {{△VAR_xxx△}} で記述してください。 △:半角スペース xxx: 半角英数字とアンダースコア(_) e.g.) VAR_users: - name: alice authorized:	×	×	○	○
グローバル変数	「グローバル変数」メニューから登録された変数です。 ※「グローバル変数」メニューについては、各ドライバーの利用手順マニュアルをご参照ください。	○	○	○	○

種類	内容	扱えるモード													
		Ansible			Tower / AWX										
		Legacy	Pioneer	Legacy Role											
astroll 独自変数	astroll 独自で定義された変数です。 基本コンソールの機器一覧の下記項目を変数として扱えます。	○	○	○	○										
	<table><tr><th>項目名</th><th>変数名</th></tr><tr><td>ホスト名</td><td>__loginhostname__</td></tr><tr><td>プロトコル</td><td>__loginprotocol__</td></tr><tr><td>ログインユーザ ID</td><td>__loginuser__</td></tr><tr><td>ログインパスワード</td><td>__loginpassword__</td></tr></table>					項目名	変数名	ホスト名	__loginhostname__	プロトコル	__loginprotocol__	ログインユーザ ID	__loginuser__	ログインパスワード	__loginpassword__
	項目名					変数名									
	ホスト名					__loginhostname__									
	プロトコル					__loginprotocol__									
	ログインユーザ ID					__loginuser__									
	ログインパスワード					__loginpassword__									
	機器一覧については、「利用手順マニュアル(astroll システム)」を参照してください。 作業実行時のディレクトリパスを下記の変数として扱えます。														
	<table><tr><th>項目名</th><th>変数名</th></tr><tr><td>作業ディレクトリパス</td><td>__workflowdir__</td></tr></table>					項目名	変数名	作業ディレクトリパス	__workflowdir__						
	項目名					変数名									
作業ディレクトリパス	__workflowdir__														
Playbook 内で作業ディレクトリパス配下にファイルを作成することで、「作業実行」の結果データでファイルをダウンロードすることができます。															
※作業実行については、各ドライバーの利用手順マニュアルをご参照ください。															
Symphony 実行時の各 Movement で共有するディレクトリパスを下記の変数として扱えます。															
<table><tr><th>項目名</th><th>変数名</th></tr><tr><td>Symphony 作業ディレクトリパス</td><td>__symphony_workflowdir__</td></tr></table>	項目名	変数名	Symphony 作業ディレクトリパス	__symphony_workflowdir__											
項目名	変数名														
Symphony 作業ディレクトリパス	__symphony_workflowdir__														
Playbook 内で Symphony 作業ディレクトリパス配下にファイルを作成することで、各 Movement 間でファイルを共有することが出来ます。また、作業実行時は、__workflowdir__と同じパスが設定されます。															
※Symphony 実行については、利用手順マニュアル_astroll_基本コンソールをご参照ください。															
読替変数	Defaults 変数定義ファイルまたは astroll readme に定義されている「VAR_ × × ×」形式以外の変数を astroll で扱う場合の変数「LCA_ × × ×」です。 詳細は「 3.4 読替表の記述」を参照して下さい。	×	×	○	○										

1.2 変数の抽出および具体値登録

astroll にアップロードされた Playbook などの資材から変数を抽出して具体値を登録できます。抽出した変数は、「[2.6 代入値自動登録設定](#)」や「[2.8 代入値管理](#)」で具体値を登録します。登録された変数と具体値は、作業実行時にホスト変数ファイルに出力されます。

※作業実行については、各ドライバーの利用手順マニュアルをご参照ください。

「[2.4 ロールパッケージ管理](#)」でアップロードされたロールパッケージ内の Playbook より、以下の書式の変数定義を抽出します。

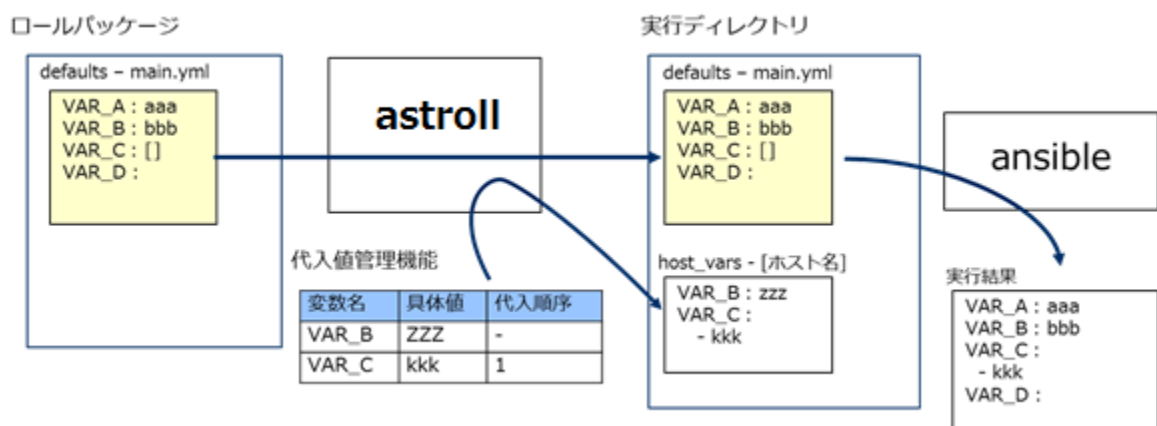
{{△VAR_xxx△}} または {%△VAR_xxx△%}
※ △:半角スペース xxx: 半角英数字とアンダースコア(_)

詳細は、「[3.2 ロールパッケージの記述](#)」をご参照ください。

※ 変数の抽出方法ならびに具体値の登録方法は対象ドライバーで違いがある為、それら詳細については各ドライバーの利用手順マニュアルをご参照ください。

1.3 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。Playbook 中の変数と、代入値管理機能で登録した変数の値の関係を、以下の図に示します。



代入値管理機能で登録した変数の値は、各ホスト用に変数定義ファイル(host_vars)に出力され、Ansible で元の Playbook と変数定義用ファイルを入力として各ホストに実行されます。

この結果、変数の値の優先順位は以下のようになります。

- ① 代入値管理機能で登録した値
 - ② Playbook 中の変数に指定した値
- 詳細は「[2.8 代入値管理](#)」を参照してください。

2 機能・操作方法説明

本章では、対象ドライバーで利用する各コンソールの内、共通している機能・操作方法について説明します。

※ 対象ドライバーで異なる機能・操作については、それぞれの利用手順マニュアルをご参照ください。

※ メニューの画面イメージや Ansible ドライバーにおけるモード別の仕様についても、対象ドライバーのそれぞれの利用手順マニュアルをご参照ください。

■登録画面項目一覧凡例

本章に記載の登録画面項目一覧表については、特別な記載がない限り 以下に示す意味とします。

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
- ・-: 入力任意
- ・/: ユーザー入力不要
- ・×: 非表示

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

2.1 Movement 一覧

Movement の登録／更新／廃止を行います。

(1) 「登録」-「登録開始」ボタンより、Movement 情報の登録を行います。

MovementID	Movement名*	遅延タイム	Ansible利用情報				備考	最終更新日時	最終更新者
			ホスト指定形式*	並列実行数	WinRM接続	gather_facts			
自動入力	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	実施 <input type="text"/>	<input type="text"/>	自動入力	自動入力

図 2.1-1 登録画面(Movement 一覧)

(2) 登録画面の項目一覧は以下のとおりです。

表 2.1-1 登録画面項目一覧(Movement 一覧)

項目	説明	入力 必須	入力形式	制約事項
Movement 名	Movement の名称を入力します。	○	手動入力	最大長 256 バイト
遅延タイマー	Movement が指定期間遅延した場合にステータスを遅延として警告表示したい場合に指定期間(1～)を入力します。(単位:分) 未入力の場合は警告表示しません。	-	手動入力	-
ホスト指定形式	特別に IP アドレスで表現しないホストを指定したい場合に「ホスト名」を選択します。通常は IP が推奨です。	○	リスト選択	-
並列実行数	Ansible が同時実行可能な対象ホスト数を入力します。 この設定値以上の対象ホストがある場合、設定値分の対象ホストに対して実行の後、残りの対象ホストをランダムに実行していきます。 AnsibleTower/AWX ドライバーの場合には、ジョブテンプレートの forks に割り当てます。 ■未入力時の振る舞いについて ・Ansible ドライバーの場合は、サーバーの設定ファイル (/etc/ansible.conf) の内容がデフォルト値となります。 ・AnsibleTower/AWX ドライバーの場合は、AnsibleTower/AWX サーバーのデフォルト値がそのまま適用されます。	-	手動入力	NULL または整数
WinRM 接続 *1	対象ホストが WindowsServer の場合に、「●」を選択します。	-	リスト選択	-
gather_facts	実行時に作業対象ホストの情報(gather_facts)を取得する場合に、「実施」(gather_facts=true)を選択します。 未選択の場合は gather_facts=false で動作します。	-	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 WindowsServer 以外の場合は、必ずリストから空白を選択してください。なお、Ansible-Pioneer は常に gather_facts=false で動作します。

【留意事項】

WinRM 接続で「●」を選択した場合は接続するホストをすべて WindowsServer とみなします。

2.2 テンプレート管理

Playbook 内で定義している 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。

テンプレート管理で Jinja2 テンプレートファイルを登録しておくことで、Playbook 内で定義する Jinja2 テンプレートファイルを template 埋め込み変数で指定することが出来ます。

(1) 「登録」-「登録開始」ボタンより、ファイル管理情報の登録を行います。

図 2.2-1 登録画面(テンプレート管理)

(2) 登録画面の項目は以下のとおりです。

表 2.2-1 登録画面項目一覧(テンプレート管理)

項目	説明	入力必須	入力形式	制約事項
テンプレート埋込変数名	template モジュールや ios_config モジュールなどのパラメータに埋め込む変数名を入力します。 変数名は、「TPF_****」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
テンプレート素材	モジュールのパラメータで使用する Jinja2 テンプレートファイルをアップロードします。	○	ファイル登録	テキスト形式
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、「テンプレート素材」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。

① Playbook の記述

テンプレート管理で登録したテンプレートモジュールを Playbook に記述する場合、該当のパラメータにテンプレート埋込変数名を記述します。テンプレート埋込変数名を使用しない場合、代入値管理で登録した変数や該当ファイルのパスを記述します。

Exp)

Playbook の記述

```
- template: src='{{△TPF_hosts△}}' dest=/etc/hosts
```

△:半角スペース

登録内容

テンプレート埋込変数名	テンプレート素材
TPF_hosts	/etc/hosts

_dest はファイル名も記述してください。ファイル名の指定がない場合、登録したテンプレート素材のファイル名の前に astroll の管理番号が付与された名前のファイル名で処理されます。

たとえば、dest=/etc/ とした場合、ファイル名は/etc/10 桁の数値 _hosts となります。

② 対話ファイルの記述

対話ファイルに記述する場合、テンプレート埋込変数名を記述します。

Exp)

対話ファイルの記述

```
- expect: '{{△_loginuser_△}}@{{△_loginhostname_△}}'
```

```
  exec: 'scp △astroll ユーザ@astroll ホスト名:{{△TPF_hosts△}}△転送先'
```

```
- expect: 'password:'
```

```
  exec: astroll ユーザのパスワード'
```

△:半角スペース

登録内容

テンプレート埋込変数名	テンプレート素材
TPF_hosts	/etc/hosts

転送先はファイル名も記述してください。ファイル名の指定がない場合、登録したファイル素材のファイル名の前に astroll の管理番号が付与された名前のファイル名で処理されます。

例えば、転送先=/etc/ とした場合、ファイル名は/etc/10 桁の数字 _hosts となります。

{{△TPF_hosts△}}は実行時に転送元の絶対パスに置換されます。

内部の処理で template 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「[2.6 代入値自動登録設定](#)」や「[2.8 代入値管理](#)」で具体値が登録可能になります。

なお、取り出しのタイミングはリアルタイムではないので、「[2.6 代入値自動登録設定](#)」や「[2.8 代入値管理](#)」で変数が扱えるまでに時間がかかる場合があります。

※ 反映のタイミングは [自動プロセスの起動周期](#)に依存します。詳細は各ドライバーの利用手順マニュアルで「[起動周期](#)」について記載していますので、そちらをご参照ください

2.3 ファイル管理

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。

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。

(1) 「登録」-「登録開始」ボタンより、ファイル管理情報の登録を行います。

図 2.3-1 登録画面(ファイル管理)

(2) 登録画面の項目は以下のとおりです。

表 2.3-1 登録画面項目一覧(ファイル管理)

項目	説明	入力必須	入力形式	制約事項
ファイル埋込変数名	各モジュールのパラメータに埋め込む変数名を入力します。 変数名は、「CPF_****」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
ファイル素材	各モジュールで使用する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、「ファイル素材」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。

① Playbook の記述

プレイブックの各モジュールに記述する場合、ファイル埋め込み変数を記述します。

e.g)

Playbook の記述

登録内容

```
-copy: src='{{△CPF_hosts△}}' dest=/etc/hosts
```

△:半角スペース

ファイル埋込変数名	ファイル素材
CPF_hosts	hosts

dest はファイル名も記述してください。ファイル名の指定がない場合、登録した**ファイル素材のファイル名の前に astroll の管理番号が付与された名前のファイル名**で処理されます。

たとえば、dest=/etc/ とした場合、ファイル名は/etc/10 桁の数字_hosts となります。

```
-unarchive src={{△CPF_tool_tgz△}} dest=/usr/local/bin
```

△:半角スペース

ファイル埋込変数名	ファイル素材
CPF_tool_tgz	tool.tgz

② 対話ファイルの記述

対話ファイルに記述する場合、ファイル埋め込み変数を記述します。

e.g)

対話ファイルの記述

登録内容

```
- expect: '{{△loginuser_△}}@{{△loginhostname_△}}'
```

```
  exec: 'scp △astroll ユーザ@astroll ホスト名:{{△CPF_hosts△}}△転送先'
```

```
- expect: 'password:'
```

```
  exec: astroll ユーザのパスワード'
```

△:半角スペース

ファイル埋込変数名	ファイル素材
CPF_hosts	hosts

転送先はファイル名も記述してください。ファイル名の指定がない場合、登録した**ファイル素材のファイル名の前に astroll の管理番号が付与された名前のファイル名**で処理されます。

たとえば、転送先=/etc/ とした場合、ファイル名は/etc/10 桁の数字_hosts となります。

{{△CPF_hosts△}}は実行時に転送元の絶対パスに置換されます。

2.4 ロールパッケージ管理

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。

(1) 「登録」-「登録開始」ボタンより、ロールパッケージ情報の登録を行います。

図 2.4-1 登録画面(ロールパッケージ管理)

(2) 登録画面の項目一覧は以下のとおりです。

表 2.4-1 登録画面項目一覧(ロールパッケージ管理)

項目	説明	入力 必須	入力形式	制約事項
ロールパッケージ名	astroll で管理するロールパッケージ名を入力します。	○	手動入力	最大長 128 バイト
ロールパッケージファイル	作成したロールパッケージファイル(zip 形式)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、「ロールパッケージファイル」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)」にロールパッケージ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。

内部の処理でロールパッケージ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、「2.6 代入値自動登録設定」や「2.8 代入値管理」で具体値の登録が可能になります。

※ 反映のタイミングは 自動プロセスの起動周期に依存します。詳細は各ドライバーの利用手順マニュアルで「起動周期」について記載していますので、そちらをご参照ください。

2.5 多段変数最大繰返数

「[2.4 ロールパッケージ管理](#)」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、繰返配列定義を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。

- (1) 「一覧」-「更新」ボタンより最大繰返数の更新を行います。(※登録ボタンではありません)

項目	変数名	メンバー変数名 (繰返し有)	最大繰返数*	備考	最終更新日時	最終更新者
1	VAR_users	-	2		自動入力	自動入力

図 2.5-1 登録画面(多段変数最大繰返数管理)

- (2) 登録画面の項目一覧は以下のとおりです。

表 2.5-1 登録画面項目一覧(多段変数最大繰返数管理)

項目	説明	入力必須	入力形式	制約事項
最大繰返数	配列の最大繰返数を 1～999 の範囲で入力します。	○	手動入力	入力値 1～999
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

メンバー変数名の表示は各階層の変数を「.」でスコープします。また、1 階層目が繰返配列の場合はメンバー変数名が「-」と表示されます。

e.g.)

変数定義	メンバー変数の表示	最大繰返数の初期値
VAR_users:		
- name: alice	-	1
authorized:		
- /tmp/alice/onekey.pub		
- craete_users:	craete_users	2
Name: root		
password: xxxxxxxx		
- craete_users:		
Name: mysql		
password: xxxxxxxx		

内部の処理でロールパッケージ内に定義している多段変数繰返配列で定義されているメンバー変数の繰返数を初期登録します。初期登録後、多段変数最大繰返数管理で繰返数を更新することが出来ます。

※ 初期登録および繰返数の更新は **astroll プロセス周期** に依存します。

詳細は各ドライバーの利用手順マニュアルをご参照ください。

2.6 代入値自動登録設定

連携対象としたパラメータリスト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。

※パラメータリストのカスタマイズ時に連携できるオプション機能です。デフォルトでは利用しません。

(1) 「登録」-「登録開始」ボタンより代入値自動登録設定を行います。

図 2.6-1 登録画面(代入値自動登録設定)

表 2.6-1 ドライバー別 対応カラム一覧(代入値自動登録設定)

カラム		ドライバー			
		Ansible			Tower / AWX
		Legacy	Legacy Role	Pioneer	
メニューグループ:メニュー		○	○	○	○
項目		○	○	○	○
登録方式		○	○	○	○
Movement		○	○	○	○
Key 情報	変数名	○	○	○	○
	メンバー変数名	—	▲	—	▲
	代入順序	△	△	△	△
Value 情報	変数名	○	○	○	○
	メンバー変数名	—	▲	—	▲
	代入順序	△	△	△	△
NULL 連携		●	●	●	●

○:必須

●:任意

△:選択した変数が複数具体値設定可能な変数の場合のみ必須

▲:選択した変数が多段変数の場合のみ必須

—:非表示

(2) 登録画面の項目一覧は以下のとおりです。

表 2.6-2 登録画面項目一覧(代入値自動登録設定)

カラム		説明	入力 必須	入力形式	制約事項
メニューグループ: メニュー		パラメータリストのメニューが表示されます。 紐付対象のメニューを選択します。	○	リスト選択	-
項目		選択したパラメータリストのメニューの項目が表示 されます。紐付対象の項目を選択します。	○	リスト選択	-
登録方式		Value 型: 項目の設定値を紐付けた変数の具体値と する場合に選択します。 Key 型: 項目の名称を紐付けた変数の具体値とする 場合に選択します。 項目の設定値が空白の場合は紐付け対象外となリ ます。 Key-Value 型: 項目の名称(Key)と設定値(Value)を 紐付けた変数の具体値とする場合に選択します。	○	リスト選択	-
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	-
Key 情報	変数名	Movement 詳細で登録した資材で使用している変数 が表示されます。 Key 型で具体値を紐付けたい変数を選択します。	○ または ／	リスト選択	登録方式で Key 型 または Key-Value 型を 選択した場合は必須
	メンバー変数名	変数名で多段変数を選択した場合に多段変数の メンバー変数が表示されます。 メンバー変数を選択します。	○ または ／	リスト選択	-
	代入順序	複数具体値が設定できる変数の場合のみ必須入力 になります。 具体値の代入順序(1～)を入力します。入力値に 従い昇順で代入されます。具体値が複数ない場合 でも代入順序(1～)を入力します。	○ または ／	手動入力	ブランク または、 正の整数
Value 情報	変数名	Movement 詳細で登録した資材で使用している変数 が表示されます。 Value 型で具体値を紐付けたい変数を選択します。	○ または ／	リスト選択	登録方式で Value 型 または Key-Value 型を 選択した場合は必須
	メンバー変数名	変数名で多段変数を選択した場合に多段変数の メンバー変数が表示されます。 メンバー変数を選択します。	○ または ／	リスト 選択	-
	代入順序	複数具体値が設定できる変数の場合のみ必須入力 になります。 具体値の代入順序(1～)を入力します。入力値に 従い昇順で代入されます。具体値が複数ない場合 でも代入順序(1～)を入力します。	○ または ／	手動入力	ブランク または、 正の整数

カラム	説明	入力 必須	入力形式	制約事項
NULL 連携	<p>パラメータシート of 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。</p> <ul style="list-style-type: none"> ・「有効」の場合、パラメータシート of 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。 ・空白の場合、Ansible インターフェース情報の「NULL 連携」の値が適用されます。 	-	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

※ メンバー変数名の表記内容については「[2.8 代入値管理](#)」をご参照ください。

※ 代入順序の入力については各ドライバーの利用手順マニュアルをご参照ください。

2.7 作業対象ホスト

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。

- (1) 「登録」-「登録開始」ボタンより、作業対象ホスト登録を行います。

※*は必須項目です。

図 2.7-1 登録画面(作業対象ホスト)

- (2) 登録画面の項目一覧は以下のとおりです。

表 2.7-1 登録画面項目一覧(作業対象ホスト)

項目	説明	入力 必須	入力形式	制約事項
オペレーション	投入オペレーション一覧に登録されているオペレーションが表示されます。 オペレーションを選択します。	○	リスト選択	-
Movement	Movement 一覧に登録されている Movement が表示されます。 オペレーションに紐付ける Movement を選択します。	○	リスト選択	-
ホスト	機器一覧に登録されているホスト名が表示されます。 オペレーションに紐付けるホストを選択します。	○	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

2.8 代入値管理

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も本機能で設定することができます。詳しくは「[3.4 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。

(1) 「登録」-「登録開始」ボタンより代入値管理を行います。

項目	オペレーション*	Movement	ホスト	変数名	メンバー変数名	具伴値	代入順序	デフォルト値
自動入力	<input type="text"/>	オペレーションを選択して下さい	オペレーションを選択して下さい	Movementを選択して下さい	変数名を選択して下さい	<input type="text"/>		

図 2.8-1 登録画面(代入値管理)

代入値管理の変数名は、アップロードされた Playbook や代入値自動登録設定で登録された情報から反映されます。

※ 反映のタイミングは 自動プロセスの起動周期 に依存します。
詳細は各ドライバーの利用手順マニュアルをご参照ください。

(2) 登録画面の項目一覧は以下のとおりです。

表 2.8-1 ドライバー別 対応カラム一覧(代入値管理)

カラム	ドライバー別 対応状況			
	Ansible			Tower / AWX
	Legacy	Legacy Role	Pioneer	
オペレーション	○	○	○	○
Movement	○	○	○	○
ホスト	○	○	○	○
変数名	○	○	○	○
メンバー変数名	—	▲	—	▲
代入順序	△	△	△	△
デフォルト値(表示のみ)	—	○	—	○

○:必須

△:選択した変数が複数具体値設定可能な変数の場合のみ必須

▲:選択した変数が多段変数の場合のみ必須

—:非表示

表 2.8-2 登録画面項目一覧(代入値管理)

項目	説明	入力 必須	入力形式	制約事項
オペレーション	作業対象ホストに登録されているオペレーションが表示されます。 オペレーションを選択します。	○	リスト選択	-
Movement	作業対象ホストに登録されているデータの中から、選択されたオペレーションに紐づく Movement が表示されます。 Movement を選択します。	○	リスト選択	-
ホスト	作業対象ホストに登録されているデータで選択されたオペレーションと Movement に紐づくホストが表示されます。 ホストを選択します。	○	リスト選択	-
変数名	Movement 詳細にて登録されている資材の中から、選択された Movement にアタッチしている変数名が表示されます。 変数を選択します。	○	リスト選択	-
メンバー変数名	変数名で多段変数を選択した場合に多段変数のメンバー変数が表示されます。 メンバー変数を選択します。	○ または ／	リスト選択	-
具体値	オペレーション/Movement/ホストで使用する変数の具体値を入力します。 ※ 変数の具体値として別途定義されている変数を設定した場合の挙動については、「 表 2.8-3 ドライバー別具体値に設定された変数の扱い一覧 」をご参照ください。	○	手動入力	最大長 1024 バイト
代入順序	複数具体値が設定できる変数の場合のみ必須入力になります。 具体値の代入順序(1～)を入力します。入力値に従い昇順で代入されます。具体値が複数ない場合でも代入順序(1～)を入力します。	○ または ／	手動入力	ブランク または、 正の整数
デフォルト値	変数名およびメンバー変数名で選択されている変数のデフォルト変数定義ファイル(defaults->main.yml)に設定されている具体値を表示します。 詳しくは「 _3.3astroll readme の記述 」を参照してください。 具体値が「Yes」「Y」「y」の場合は True が表示されます。具体値が「No」「N」「n」の場合は False が表示されます。	-	表示のみ	-
備考	自由記述欄です。	-	手動入力	最大長 4000 バイト

※ 代入順序の入力については各ドライバーの利用手順マニュアルを ご参照ください。

表 2.8-3 ドライバー別 具体値に設定された変数の扱い一覧

カラム	ドライバー別 変数の扱い			
	Ansible			Tower / AWX
	Legacy	Legacy Role	Pioneer	
ファイル埋込変数	○	○	×	○
テンプレート埋込変数	○	×	×	×

○:扱われる

×:扱わない(単なる値として見なされる) *1

【メンバー変数名の表示内容】

多段変数の場合にのみメンバー変数の選択が必要になります。メンバー変数に表示される変数は具体値を必要とする変数のみです。

メンバー変数名の表示は各階層の変数を「.」でスコープします。繰返配列の場合は「[]」で繰返位置(0～)をスコープします。繰返し配列の数は「[2.5 多段変数最大繰返数](#)」で設定を行います。

e.g.)

変数定義	メンバー変数の表示
VAR_users:	
- name: alice	[0].name
authorized:	[0].authorized
- /tmp/alice/onekey.pub	
mysql:	
password: mysql-password	[0].mysql.password
hosts:	[0].mysql.hosts
- "127.0.0.1"	
- "localhost"	
- name: bob	[1].name
authorized:	[1].authorized
- /tmp/alice/onekey.pub	
mysql:	[1].mysql.password
password: mysql-password	[1].mysql.hosts
hosts:	※mysql は階層を示す変数なのでメンバー変数には表示されません。
- "127.0.0.1"	
- "localhost"	

代入値自動登録設定で登録した情報は、内部の処理で代入値管理と作業対象ホストへ反映されます。

※ 反映のタイミングは 自動プロセスの起動周期 に依存します。
詳細は各ドライバーの利用手順マニュアルをご参照ください。

*1 ドライバーに該当の変数を定義する機能が無い場合も含まれます

3 構築コード記述方法

3.1 Playbook の記述

基本書式については Ansible の公式内容を参照してください。

文字コードは、UTF-8 で作成してください。

Playbook 内のインデントは 2 倍数で調整してください。

```
e.g.)
-△name: サービススクリプト配置
△△template:
△△△△src:  "{{ item.src }}"
△△△△dest: "{{ item.dest }}"
△△△△owner: "{{ item.owner is none |ternary('root', item.owner) }}"
△△△△group: "{{ item.group is none |ternary('bacula', item.group) }}"
△△△△mode:  "{{ item.mode is none |ternary('0654', item.mode) }}"
           ～略～
△:半角スペース
```


3.2 ロールパッケージの記述

基本書式については Ansible ベストプラクティスの公式内容を参照してください。

文字コードは、UTF-8 で作成してください。

「[2.4 ロールパッケージ管理](#)」でアップロードするロールパッケージファイルの Zip に含めるべきディレクトリと、astroll での扱いについて記述します。

(上位ディレクトリ)

site.yml	site.yml(マスターPlaybook)は astroll で作成します。存在する場合は上書きします。
hosts	hosts ファイルは astroll で作成します。存在する場合は上書きします。
—group_vars	ホストグループ変数は扱えません。group_vars ディレクトリが存在する場合は削除します。
—host_vars	ホスト変数は astroll で作成します。host_vars ディレクトリが存在する場合は上書きします。
—astroll readme	astroll readme はロールごとに、代入値管理機能に各変数の型を指定するファイルです。 無くてもエラーにはなりません。
	astroll readme は該当ロールのデフォルト変数定義ファイルと同様の扱いをします。
	astroll readme のファイル名 astroll_readme_[ロール名].yml e.g.)
	ロール名: mysql ファイル名: astroll_readme_mysql.yml
—読替表	読替表はロール毎に定義します。defaults 変数定義ファイルまたは astroll readme に定義されている「VAR_xxx」以外の変数を astroll で扱えるようにするための定義ファイルです。 無くてもエラーにはなりません。
	読替表のファイル名 astroll_translation-table_[ロール名].txt e.g.)
	ロール名: mysql ファイル名: astroll_translation-table_mysql.txt
	上記以外のディレクトリやファイルが存在する場合は astroll は関知しません。
—roles	roles ディレクトリが存在しない場合はアップロードでエラーになります。
—[role 名①]	role 名ディレクトリが存在しない場合はアップロードでエラーになります。
—readme.mc	astroll は関知しません。
—tasks	tasks ディレクトリが存在しない場合はアップロードでエラーになります。 サブディレクトリが存在する場合はアップロードでエラーになります。
	tasks 配下のファイルは 変数抜出 の対象となります。
main.yml	main.yml がいない場合はアップロードでエラーになります。
—handlers	handlers ディレクトリの有無は関知しません。 サブディレクトリが存在した場合はアップロードでエラーになります。
	handlers 配下のファイルは 変数抜出 の対象となります。
main.yml	main.yml の有無は関知しません。
iptables.yml	main.yml が存在しないで他ファイルがある場合はアップロードでエラーになります。
—templates	templates ディレクトリの有無は関知しません。
hosts.j2	templates 配下のファイルは 変数抜出 の対象となります。
user_files	サブディレクトリが存在した場合は、サブディレクトリ配下のファイルも 変数抜出 の対象 となります。
user.j2	
—files	files ディレクトリの有無は関知しません。 サブディレクトリの有無も関知しません。

		sudoers	ファイルの有無、内容は関知しません。
		vars	vars ディレクトリの有無は関知しません。
			サブディレクトリが存在した場合はアップロードでエラーになります。
		main.yml	ファイルの有無、内容は関知しません。
			main.yml 以外のファイルも同様です。
		defaults	defaults ディレクトリがない場合はアップロードでエラーになります。
		main.yml	main.yml がない場合はアップロードでエラーになります。
			role 内の Playbook で使用している変数など astroll より具体値を設定する必要がある
			変数は main.yml に 変数の構造 を登録します。
			main.yml に登録されていない変数を role 内の Playbook で使用している場合は
			アップロードでエラーになります。
			main.yml にのみ登録されている変数が存在してもエラーとなりません。
			サブディレクトリが存在した場合はアップロードでエラーになります。
			main.yml 以外のファイルが存在する場合はアップロードでエラーになります。
		meta	meta ディレクトリの有無は関知しません。
			meta 配下のファイルは 変数抽出 の対象となります。
			サブディレクトリが存在した場合はアップロードでエラーになります。
		main.yml	ファイルの有無、内容は関知しません。
			main.yml 以外のファイルも同様です。
		meta.yml	main.yml が存在しないで他ファイルがある場合はアップロードでエラーになります。
		上記以外のディレクトリやファイルが存在する場合、astroll は関知しません。	
		[role 名②]	ロールの数に特に制限はありません。

(1) 変数抽出と変数構造

各ファイルから抽出する変数は、以下の書式の変数のみになります。

```
{{ VAR_xxx }}
```

xxx: 半角英数字とアンダースコア(_)

抽出される変数はデフォルト変数定義ファイル(defaults->main.yml)に変数の構造を定義する必要があります。

【留意事項】

デフォルト変数定義ファイルに Ansible の文法エラーがあると、変数の構造を誤解析する可能性があります。Ansible の文法エラーがないデフォルト定義ファイルを使用してください。

(2) 変数名の一意管理

astroll の代入値管理で登録された変数の情報はホスト変数として扱います。変数名はドライバー毎の全ロールパッケージで一意管理します。ロールを跨って同じ変数名を使用しているが変数構造が違う場合は、アップロード時にエラーとなります。

e.g.) 通常変数と多段変数や多段変数同士で多段構造が違う場合など。

(3) デフォルト変数定義ファイル(defaults->main.yml)の astroll 独自仕様

デフォルト変数定義ファイルの記述(変数定義)には astroll 独自仕様があります。

詳しくは、別資料「Ansible 利用ガイドライン(astroll 追加ルール)」を参照してください。

(4) Ansible Role Directory Structure における所定ディレクトリのサブディレクトリについて注意事項

Ansible Role Directory Structure における所定ディレクトリの下位に当該の所定ディレクトリ名と同名のサブディレクトリ(例えば、files ディレクトリの下位にそれと同名の files ディレクトリなど)を作成されていると、作業実行時にエラーになります。^{*2}

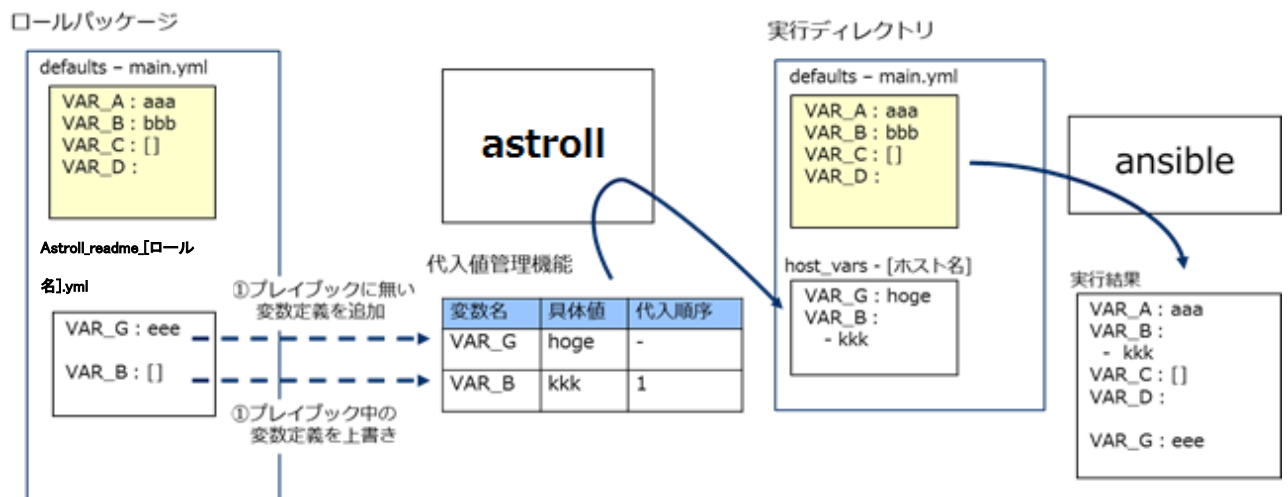
^{*2} ディレクトリが存在していても main.yml で当該に対する path の指定が無ければエラーになりません。

3.3 astroll readme の記述

代入値管理機能は、defaults 変数定義ファイルに定義した変数の型を解釈して、各変数およびそのメンバー変数などに変数の値を設定します。

Playbook 中に直接変数を定義したくない場合など、defaults 変数定義ファイルに変数が定義されていない場合、astroll readme ファイルに変数の定義を設定することで、代入値管理機能で変数の値を指定することができます。

astroll readme ファイルと、代入値管理機能の関係を以下の図に示します。



Playbook 中に無い変数を astroll readme ファイルで定義した場合、定義した変数を代入値管理機能で値を設定することができます。

また、Playbook 中の変数と異なる型を astroll readme ファイルで定義した場合、代入値管理機能には、上書きした変数の型で値を登録することができます。

代入値管理機能で設定した値は、各ホスト用に変数定義ファイル(host_vars)に出力され、Ansible で元の Playbook と変数定義用ファイルを入力として各ホストに実行されます。

astroll readme ファイルは代入値管理機能に変数情報を与えるためだけに使用され、astroll readme に定義した変数および、変数の値は Ansible 実行には影響を与えません。

astroll readme の作成は任意です。astroll readme と defaults 変数定義ファイルで変数定義が重なった場合など、以下のルールで処理されます。

表 3.3-1 変数採用ルール

defaults 変数定義ファイル	astroll readme	変数定義の採用先
定義あり	定義なし	デフォルト変数定義ファイル
定義なし	定義あり	astroll readme
定義あり	定義あり	astroll readme

また、「[2.8 代入値管理](#)」に表示するデフォルト値は以下のルールで処理されます。

表 3.3-2 デフォルト値表示ルール

defaults 変数 定義ファイル	astroll readme	デフォルト値の扱い
あり	なし	デフォルト変数定義ファイルを採用。
なし	あり	デフォルト値なしの扱いとなる。
あり	あり	デフォルト変数定義ファイルを採用。 ただし、変数定義が一致している場合のみ。変数定義が一致していない場合はデフォルト値なしの扱いとなる。

astroll readme は、作業実行時はロールパッケージから切り離されます。

astroll readme 記載した変数と具体値は適用されません。

3.4 読替表の記述

defaults 変数定義ファイルまたは astroll readme に定義されている「VAR_xxx」以外の変数に対して、「2.8 代入値管理」機能で変数の具体値を設定出来るようにするための設定を行うファイルです。
defaults 変数定義ファイルまたは astroll readme に定義されている「VAR_xxx」以外の変数「任意変数」に対して代入値管理機能で扱う変数「読替変数」の紐付を定義します。

(1) 読替表のフォーマット

テキスト形式で下記フォーマットとなります。

ロール内で読替変数と任意変数の組合せは一意である必要があります。

読替変数(\$s*):(\$s+)任意変数

読替変数:LCA_***

***: 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)

任意変数:(最小値:1 バイト、最大値:128 バイト)

(\$s*): 半角スペース 0 個以上

(\$s+): 半角スペース 1 個以上

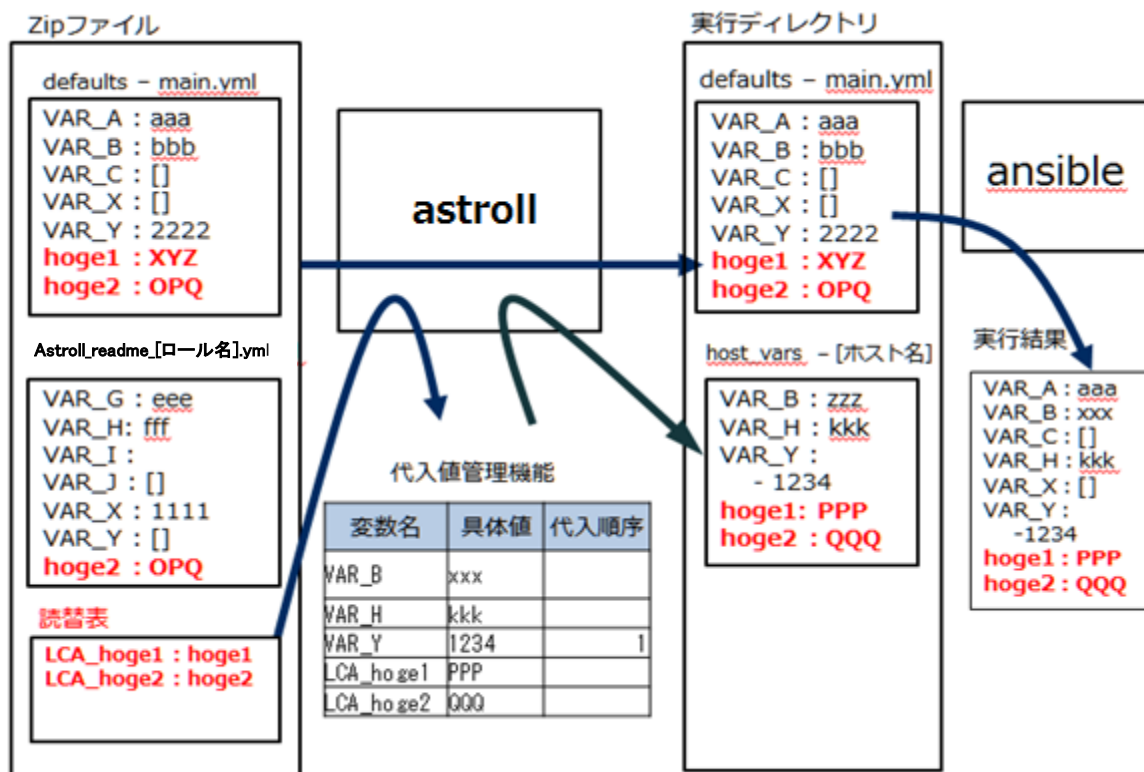
e.g.)

LCA_var1: var1

#から始まる行はコメント行

LCA_var2: var2

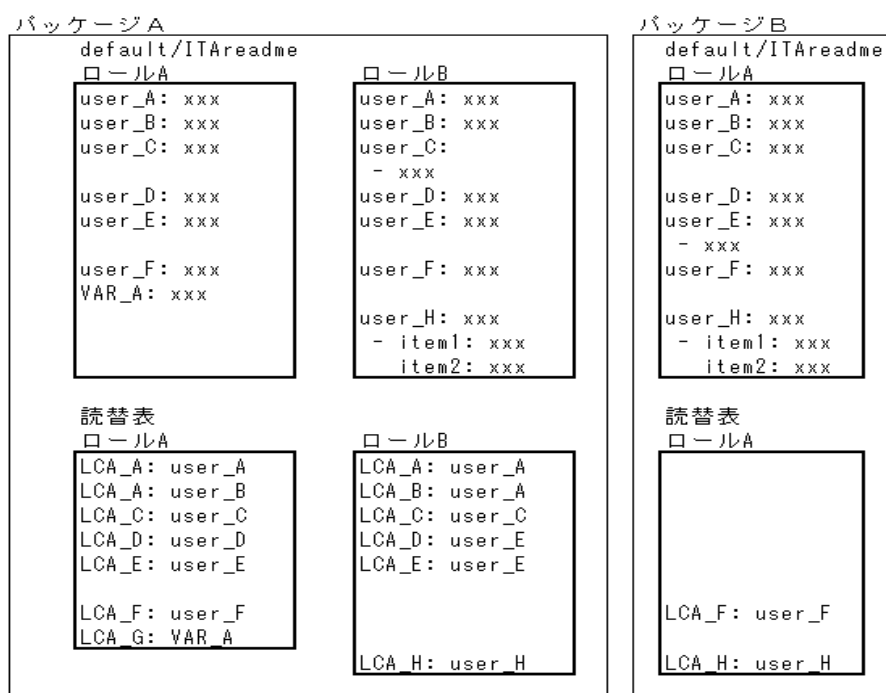
代入値管理機能の関係を以下の図に示します。



(2) 注意事項

読替表を作成する際の注意事項を列挙します。

ケース	astroll の動作	備考
読替表はあるが、defaults 変数定義ファイルと astroll readme がない。(ロール毎)	読替表を読込まない。	
任意変数に VAR_で始まる変数が定義されている。	ロールパッケージアップロード時にエラーになります。	
defaults 変数定義ファイルと astroll readme に定義されていない任意変数を使用している。	ロールパッケージアップロード時にエラーになります。	
ロール内で読替変数が重複定義されている	ロールパッケージアップロード時にエラーになります。	パッケージ A->ロール A LCA_A: user_A/LCA_A: user_B
ロール内で任意変数が重複定義されている	ロールパッケージアップロード時にエラーになります。	パッケージ A->ロール B LCA_A: user_A/LCA_B: user_A
ロール間で任意変数の構造に差異がある。	ロールパッケージアップロード時にエラーになります。	パッケージ A->ロール A/B LCA_C: user_C
ロールパッケージ内で置換変数と任意変数の組合せが一意でない。	ロールパッケージアップロード時にエラーになります。	パッケージ A ロール A LCA_D: user_D ロール B LCA_D: user_E
ロールパッケージ間で任意変数の構造に差異がある。	ロールパッケージアップロード時にエラーになりませんが、読替変数が代入値管理に表示されません。	パッケージ A->ロール A LCA_F: user_F パッケージ B->ロール A LCA_F: user_F
ロールパッケージ間で多段の任意変数のを定義している	多段の構造が一致しているのでエラーにはなりませんが、多段繰返回数の設定は各パッケージで共通の設定となります。	パッケージ A->ロール B LCA_H: user_H パッケージ B->ロール A LCA_H: user_H



3.5 BackYard コンテンツ

BackYard は、Web コンソールと連携し、サーバー上で独立して動作する常駐プロセスです。常駐プロセスでは以下の処理を行います。

(1) 変数自動登録

変数解析対象の資材をアップロードした場合、アップロードされた資材から変数を取り出します。

表 3.5-1 ドライバー別 具体値に設定された変数の扱い一覧

メニュー	ドライバー別 変数解析対象			
	Ansible			Tower / AWX
	Legacy	Legacy Role	Pioneer	
プレイブック素材集	○	×	×	×
ロールパッケージ管理	×	○	×	○
対話ファイル素材集	×	×	○	×

なお、取出すタイミングは 自動プロセスの起動周期 に依存します。

※ 自動プロセスの起動周期については、各ドライバーの利用手順マニュアルをご参照ください。

※変数名の一意管理

取出した変数名は、ドライバー毎(モードを持つ場合はモード毎)に全資材で一意管理されます。

デフォルト変数定義ファイルで変数構造を定義しますので、各資材で変数構造が違ふ場合の留意事項を以下に明記します。

●単一ロールパッケージ

ロールを跨って同じ変数名を使用しているが変数構造が違ふ場合。

※通常変数と多段変数や多段変数同士で多段構造が違う場合など

⇒ アップロード時にエラーとなります。

●全ロールパッケージ

ロールパッケージを跨って同じ変数名を使用しているが変数構造が違う場合。

⇒ アップロード時にエラーにはなりませんが、「2.6 代入値自動登録設定」と「2.8 代入値管理」に該当の変数が表示されません。

(2) 代入自動値登録設定

連携対象としたパラメータリストのオペレーションとホスト毎の項目の設定値を紐付けた Movement と変数の情報を代入値管理と作業対象ホストに反映されます。

なお、反映のタイミングは前述と同様に 自動プロセスの起動周期 に依存します。

※ 自動プロセスの起動周期については、各ドライバーの利用手順マニュアルをご参照ください。

作業対象ホストと代入値管理は複数の操作者が更新を行います。最終更新者が他操作者の場合は反映処理をしません。

代入値自動登録設定のデータを反映したい場合は、代入値管理で該当レコードを廃止にする。他 BackYard 処理で該当レコードの更新を無効にする。などの操作を行ってください。

作業対象ホストと代入値管理への反映ルールを以下に明記します。

① 代入値自動登録に登録されている情報を代入値管理へ反映時

代入値管理 の状態	該当レコード なし	該当レコードあり			該当レコード 廃止中
		=具体値	≠具体値		
			最終更新者		
			BackYard 処理	他操作者	
代入値管理 への反映	新規レコード 追加	-	該当レコードの 具体値更新	-	廃止レコード 復活

※ 該当レコード:オペレーション+ホスト+Movement+変数名+(メンバー変数)+(代入順序)が同一のレコードの意

② 代入値自動登録に登録されていない情報(代入値管理のみに登録)を代入値管理へ反映時

代入値管理 の状態	該当レコードあり	
	最終更新者	
	BackYard 処理	他操作者
代入値管理への反映	該当レコード廃止	-

③ 代入値自動登録に登録されている情報を作業対象ホストへの反映時

作業対象ホストの状態	該当レコードなし	該当レコードあり	該当レコード廃止中
作業対象ホストへの反映	新規レコード追加	-	廃止レコード復活

該当レコード:オペレーション+ホスト+Movement が同一のレコードの意

④ 代入値自動登録に登録されていない情報(作業対象ホストのみに登録)を作業対象ホストへ反映時

代入値管理の状態	該当レコードあり	
	最終更新者	
	BackYard 処理	他操作者
代入値管理への反映	該当レコード廃止	-

- ⑤ 同一変数に複数の項目の紐付が登録されている場合
代入値管理および作業対象ホストへの反映対象外となります。

3.6 Ansible 利用ガイドライン astroll 追加ルール

astroll を使用して、Ansible で実行する為の Playbook 作成ガイドラインを記述します。
詳しくは、別資料「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」を参照してください。

4 トラブルシューティング

No	内容
Q-1	機器一覧の認証方式を公開鍵認証に設定したホストに対してオーケストレーターの作業実行をしたが、ターゲットホストへの接続でエラーとなる。
A-1	Ansible サーバーおよび AnsibleTower サーバーまたは AWX サーバーの .ssh/known_hosts にターゲットホスト側の公開鍵が登録されているかを確認してください。 上記サーバーからターゲットホストに対して ssh コマンドで該当ホスト名または IP アドレスでログインすることで公開鍵が登録されます。
Q-2	機器一覧の認証方式をパスワード認証に設定したホストに対してオーケストレーターの作業実行をしたが、ターゲットホストへの接続でエラーとなる。
A-2	ターゲットホスト側の ssh の設定で password 認証を許可しているか確認してください。
Q-3	Movement 一覧でホスト指定形式を指定して各オーケストレーターの作業実行をしたがターゲットホストへの接続でエラーとなる。
A-3	/etc/hosts の登録など該当ホスト名のアドレス解決が出来ているか確認してください。 ssh 接続の場合、Ansible サーバーおよび AnsibleTower サーバーまたは AWX サーバーの known_hosts に該当ホスト名でターゲットホスト側の公開鍵が登録されているかを確認して下さい。
Q-4	プレイブックで/tmp にファイルを作成しているが、/tmp にファイルが作成されていない。
A-4	RHEL7 より PrivateTmp 機能が追加されています。astroll でインストールした Apache は PrivateTmp 機能を有効にしています。astroll からのプレイブックの実行は Apache を経由していますので、/tmp が別ディレクトリにバインドされています。astroll ではプレイブックからの /tmp へのファイルアクセスは控えて頂くことを推奨しています。
Q-5	Ansible から Windows サーバへの接続ができない。
A-5	Windows サーバで Powershell を管理者権限で起動し、下記の各コマンドを実施して下さい。 C:¥User¥User> Enable-PSRemoting -Force ↓ C:¥User¥User> Set-NetConnectionProfile -InterfaceAlias (Get-NetConnectionProfile -IPv4Connectivity Internet).InterfaceAlias -NetworkCategory Private ↓ C:¥User¥User> winrm set winrm/config/service/auth '@{Basic="true"}' ↓ C:¥User¥User> winrm set winrm/config/service '@{AllowUnencrypted="true"}'