

拡張カスタムジョブ部品利用
の手引き

JobCenter

R15.4

-
- Windows, Windows Server, Microsoft Azure, Microsoft Excel および Internet Explorer は、米国 Microsoft Corporation 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。
 - UNIX は、The Open Group が独占的にライセンスしている米国ならびにほかの国における登録商標です。
 - HP-UX は、米国 HP Hewlett Packard Group LLC の商標です。
 - AIX は、米国 IBM Corporation の商標です。
 - 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。
 - Oracle Linux, Oracle Clusterware および Java は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
 - 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。
 - SUSE は、SUSE LLC 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。
 - NQS は、NASA Ames Research Center のために Sterling Software 社が開発した Network Queuing System です。
 - SAP ERP, SAP NetWeaver BW および ABAP は、SAP AG の登録商標または商標です。
 - Amazon Web Services は、Amazon.com, Inc. またはその関連会社の米国およびその他の国における商標です。
 - iPad および Safari は、米国およびその他の国で登録された Apple Inc. の商標です。
 - Docker は、米国およびその他の国で登録された Docker, Inc. の登録商標または商標です。
 - その他、本書に記載されているソフトウェア製品およびハードウェア製品の名称は、関係各社の登録商標または商標です。

なお、本書内では、R、TM、cの記号は省略しています。

輸出する際の注意事項

本製品（ソフトウェア）は、外国為替令に定める提供を規制される技術に該当いたしますので、日本国外へ持ち出す際には日本国政府の役務取引許可申請等必要な手続きをお取りください。許可手続き等にあたり特別な資料等が必要な場合には、お買い上げの販売店またはお近くの当社営業拠点にご相談ください。

はじめに

本書は、拡張カスタムジョブ部品の利用方法について説明しています。なお、本書内に記載されている画面例と実際の画面とは異なることがありますので注意してください。

本書の内容は将来、予告なしに変更する場合があります。あらかじめご了承ください。

1. マニュアルの読み方

- 本バージョンにおける新規機能や変更事項を理解したい場合
→ <リリースメモ>を参照してください。
- JobCenter を新規にインストール、またはバージョンアップされる場合
→ <インストールガイド>を参照してください。
- JobCenter をコンテナ環境で構築、運用をする場合
→ <コンテナガイド>を参照してください。
- JobCenter を初めて利用される場合
→ <クイックスタート編>を参照してください。
- JobCenter の基本的な操作方法を理解したい場合
→ <基本操作ガイド>を参照してください。
- 環境の構築や各種機能の設定を理解したい場合
→ <環境構築ガイド>を参照してください。
- JobCenter の操作をコマンドラインから行う場合
→ <コマンドリファレンス>を参照してください。
- JobCenter の運用方法を理解したい場合
→ <運用・構築ガイド>を参照してください。
- 運用中のJobCenter を新環境に移行する場合
→ <移行ガイド>を参照してください。
- クラスタ環境で運用中のJobCenter をバージョンアップする場合
→ <クラスタ環境でのバージョンアップ・パッチ適用ガイド>を参照してください。
- その他機能についてお知りになりたい場合
→ 関連マニュアルの内容をお読みいただき、目的のマニュアルを参照してください。

2. 凡例

本書内での凡例を紹介します。

	気をつけて読んでいただきたい内容です。
	本文中の補足説明
	本文中のヒントとなる説明
注	本文中につけた注の説明
—	UNIX版のインストール画面の説明では、__部分(下線部分)はキーボードからの入力を示します。

3. 関連マニュアル

JobCenter に関するマニュアルです。JobCenter メディア内に格納されています。

最新のマニュアルは、JobCenter 製品サイトのダウンロードのページを参照してください。

<https://jpn.nec.com/websam/jobcenter/download.html>

資料名	概要
JobCenter インストールガイド	JobCenterを新規にインストール、またはバージョンアップする場合の方法について説明しています。
JobCenter クイックスタート編	初めてJobCenterをお使いになる方を対象に、JobCenterの基本的な機能と一通りの操作を説明しています。
JobCenter 基本操作ガイド	JobCenterの基本機能、操作方法について説明しています。
JobCenter 環境構築ガイド	JobCenterを利用するために必要な環境の構築、環境の移行や他製品との連携などの各種設定方法について説明しています。
JobCenter NQS機能利用の手引き	JobCenterの基盤であるNQSの機能をJobCenterから利用する方法について説明しています。
JobCenter 操作・実行ログ機能利用の手引き	JobCenter CL/Winからの操作ログ、ジョブネットワーク実行ログ取得機能および設定方法について説明しています。
JobCenter コマンドリファレンス	GUIと同様にジョブネットワークの投入、実行状況の参照などをコマンドラインから行うために、JobCenterで用意されているコマンドについて説明しています。
JobCenter クラスタ機能利用の手引き	クラスタシステムでJobCenterを操作するための連携方法について説明しています。
JobCenter Helper機能利用の手引き	Excelを用いたJobCenterの効率的な運用をサポートするJobCenter Definition Helper (定義情報のメンテナンス)、JobCenter Report Helper (帳票作成)、JobCenter Analysis Helper (性能分析)の3つの機能について説明しています。
JobCenter SAP機能利用の手引き	JobCenterをSAPと連携させるための方法について説明しています。
JobCenter WebOTX Batch Server連携機能利用の手引き	JobCenterをWebOTX Batch Serverと連携させるための方法について説明しています。
JobCenter Web機能利用の手引き	Webブラウザ上でジョブ監視を行うことができるJobCenter CL/Webについて説明しています。
JobCenter テキスト定義機能の利用手引き	JobCenterの定義情報をテキストファイルで定義する方法について説明しています。
JobCenter クラスタ環境でのバージョンアップ・パッチ適用ガイド	クラスタ環境で運用しているJobCenterのアップデート、パッチ適用手順を説明しています。
JobCenter 拡張カスタムジョブ部品利用の手引き	拡張カスタムジョブとして提供される各部品の利用方法について説明しています。
JobCenter 運用・構築ガイド	JobCenterの設計、構築、開発、運用について横断的に説明しています。
JobCenter 移行ガイド	運用中のJobCenterを別の新環境に移行する手順について横断的に説明しています。
JobCenter コンテナガイド	JobCenterをコンテナ環境で構築・運用する方法について説明しています。
JobCenter R15.4 リリースメモ	バージョン固有の情報を記載しています。

4. 改版履歴

版数	変更日付	項目	形式	変更内容
1	2020/06/25	新規作成	－	第1版

目次

はじめに	iii
1. マニュアルの読み方	iv
2. 凡例	v
3. 関連マニュアル	vi
4. 改版履歴	vii
1. 概要	1
1.1. 適用方法	2
1.2. 共通仕様	3
1.2.1. 投入キュー	3
1.2.2. カスタムジョブテンプレートのカスタマイズ	4
1.2.3. 暗号化パラメータの利用	4
1.2.4. ジョブネットワークの環境変数の利用	5
1.2.5. 表示アイコン	6
1.2.6. 注意事項	6
2. メール送信部品	8
2.1. メール送信部品	9
2.1.1. 部品情報	9
2.1.2. スクリプト	9
2.1.3. カスタムパラメータ設定	9
2.1.4. 実行結果	10
2.1.5. 主要メッセージ	10
2.1.6. 注意事項	10
3. HTTP接続部品	12
3.1. HTTP接続部品共通仕様	13
3.1.1. リクエスト種別(HTTPメソッド)	13
3.1.2. 制御パラメータ(カスタムパラメータ)一覧	13
3.1.3. HTTPリクエストパラメータ	16
3.1.4. 実行結果	18
3.1.5. 主要メッセージ	19
3.1.6. 注意事項	19
3.2. HTTP-GET部品	20
3.2.1. 部品情報	20
3.3. HTTP-PUT部品	21
3.3.1. 部品情報	21
3.4. HTTP-POST部品	22
3.4.1. 部品情報	22
3.5. HTTP-DELETE部品	23
3.5.1. 部品情報	23
3.6. HTTP-HEAD部品	24
3.6.1. 部品情報	24
3.7. HTTP-PATCH部品	25
3.7.1. 部品情報	25
4. Amazon Web Services 連携部品	26
4.1. EC2	27
4.1.1. EC2部品共通のカスタムパラメータ	27
4.1.2. インスタンス起動	28
4.1.3. インスタンス停止	30
4.1.4. インスタンス情報取得	32
4.1.5. インスタンスプロパティ変更	34
4.1.6. AMI作成	36
4.1.7. 世代管理付きAMI作成	38
4.1.8. タグ作成	42
4.1.9. スナップショット作成	44
4.1.10. スナップショット削除	49

4.1.11. スナップショット情報取得	51
4.1.12. スナップショットコピー	53
4.1.13. EBSボリュームサイズ拡張	55
4.1.14. EBSボリュームタイプ変更	57
4.2. S3	59
4.2.1. S3部品共通のカスタムパラメータ	59
4.2.2. S3部品共通仕様	59
4.2.3. ファイルダウンロード	60
4.2.4. ファイルアップロード	64
4.2.5. ファイル削除	68
4.2.6. S3部品共通の注意事項	69
4.3. ELB	70
4.3.1. ELB部品共通のカスタムパラメータ	70
4.3.2. クラシックロードバランサーへのインスタンス登録	71
4.3.3. クラシックロードバランサーからのインスタンス登録解除	73
4.4. CF	75
4.4.1. CF部品共通のカスタムパラメータ	75
4.4.2. スタック作成	76
4.4.3. スタック削除	78
4.4.4. スタック情報取得	80
4.5. CW	82
4.5.1. CW部品共通のカスタムパラメータ	82
4.5.2. アラーム情報取得	83
4.5.3. アラーム状態待ち合わせ	85
4.6. RDS	88
4.6.1. RDS部品共通のカスタムパラメータ	88
4.6.2. インスタンス起動	89
4.6.3. インスタンス停止	92
5. Microsoft Azure(V2) 連携部品	96
5.1. 部品共通のカスタムパラメータ	97
5.1.1. カスタムパラメータ設定	97
5.2. 仮想マシン起動部品	98
5.2.1. 部品情報	98
5.2.2. 部品説明	98
5.2.3. スクリプト	98
5.2.4. カスタムパラメータ設定	98
5.2.5. 実行結果	99
5.2.6. 主要メッセージ	99
5.2.7. 注意事項	99
5.3. 仮想マシン停止およびリソース解放部品	100
5.3.1. 部品情報	100
5.3.2. 部品説明	100
5.3.3. スクリプト	100
5.3.4. カスタムパラメータ設定	100
5.3.5. 実行結果	101
5.3.6. 主要メッセージ	101
5.3.7. 注意事項	101
5.4. 仮想マシン停止部品	102
5.4.1. 部品情報	102
5.4.2. 部品説明	102
5.4.3. スクリプト	102
5.4.4. カスタムパラメータ設定	102
5.4.5. 実行結果	103
5.4.6. 主要メッセージ	103
5.4.7. 注意事項	103
5.5. 仮想マシン情報(モデルビュー)取得	104

5.5.1. 部品情報	104
5.5.2. 部品説明	104
5.5.3. スクリプト	104
5.5.4. カスタムパラメータ設定	104
5.5.5. 実行結果	104
5.5.6. 主要メッセージ	105
5.5.7. 注意事項	105
5.6. 仮想マシン情報(インスタンスビュー)取得	106
5.6.1. 部品情報	106
5.6.2. 部品説明	106
5.6.3. スクリプト	106
5.6.4. カスタムパラメータ設定	106
5.6.5. 実行結果	106
5.6.6. 主要メッセージ	107
5.6.7. 注意事項	107
6. UiPath 連携部品	108
6.1. ジョブ開始	109
6.1.1. 部品情報	109
6.1.2. 部品説明	109
6.1.3. スクリプト	109
6.1.4. カスタムパラメータ設定	109
6.1.5. 部品詳細	110
6.1.6. 実行結果	112
6.1.7. 主要メッセージ	112
6.1.8. 使用API	113
6.1.9. 注意事項	114
7. Box連携部品	115
7.1. Box連携部品概要	116
7.1.1. 事前準備	116
7.2. 部品共通のカスタムパラメータ	122
7.2.1. カスタムパラメータ設定	122
7.3. BOX_APIトークン取得部品	123
7.3.1. 部品情報	123
7.3.2. 部品説明	123
7.3.3. スクリプト	123
7.3.4. カスタムパラメータ設定	123
7.3.5. 実行結果	123
7.3.6. 主要メッセージ	123
7.3.7. 注意事項	124
7.4. BOXアップロード部品	125
7.4.1. 部品情報	125
7.4.2. 部品説明	125
7.4.3. スクリプト	125
7.4.4. カスタムパラメータ設定	125
7.4.5. 実行結果	126
7.4.6. 主要メッセージ	126
7.4.7. 注意事項	127
7.5. BOXダウンロード部品	128
7.5.1. 部品情報	128
7.5.2. 部品説明	128
7.5.3. スクリプト	128
7.5.4. カスタムパラメータ設定	128
7.5.5. 実行結果	129
7.5.6. 主要メッセージ	129
7.5.7. 注意事項	130
7.6. コンテンツID取得部品	131

7.6.1. 部品情報	131
7.6.2. 部品説明	131
7.6.3. スクリプト	131
7.6.4. カスタムパラメータ設定	131
7.6.5. 実行結果	132
7.6.6. 主要メッセージ	132
7.6.7. 注意事項	132

1. 概要

本章では拡張カスタムジョブ部品について説明します。

拡張カスタムジョブは、部品の種別毎にJPFファイルとして提供され、ファイルをアップロードすることで利用することができます。

アップロードした部品はマネージャフレーム中のカスタムジョブ定義一覧で参照することができます。種別毎にグループ化され、Systemグループ配下に格納されます。

JobCenter環境への適用方法は、「[1.1 適用方法](#)」および<基本操作ガイド>の12章「アップロード・ダウンロード機能」を参照してください。また、カスタムジョブ定義の利用方法については<基本操作ガイド>の5章「カスタムジョブの利用方法」を参照してください。

本手引きでは各部品の利用方法について説明します。

1.1. 適用方法

1. 拡張カスタムジョブのJPFファイルを、CL/Winがインストールされたマシンの任意の場所にコピーします。



拡張カスタムジョブのJPFファイルはJobCenterメディアに同梱されています。また最新JPFファイルはJobCenter 製品サイトのダウンロードページから入手する事ができます。

JobCenterメディア同梱のJPFファイルの格納場所は、JobCenterメディアのリリースメモ (RELMEMO)を参照してください。

2. CL/WinからJobCenter管理者ユーザで対象サーバへログインします。 [ファイル]-[Helper機能]-[アップロード]を選択します。
3. 「アップロード対象のユーザの指定」画面でJobCenter管理者ユーザを選択します。
4. (1)でコピーしたJPFファイルを選択して「アップロード」を行います。
5. CL/Winのマネージャフレームで、カスタムジョブ定義一覧を表示して、Systemグループ配下に拡張カスタムジョブの定義が登録されていることを確認して下さい。

1.2. 共通仕様

1.2.1. 投入キュー

拡張カスタムジョブ部品はパイプキューに投入することもできますが、カスタムキュー属性を持つバッチキューで実行する必要があります。カスタムキュー属性を持たないバッチキューに投入した場合には、正常に動作しません。

1.2.1.1. カスタムキュー属性の付与手順

バッチキューにカスタムキュー属性を付与する手順を説明します。

1. CL/WinからJobCenter管理者ユーザで対象サーバへログインし、NQSフレームを開いて [マシン一覧] を表示します。
2. カスタムキュー属性を付与したいバッチキューを選択し、右クリックしたときのポップアップメニューから [キューパラメータ] を選択します。

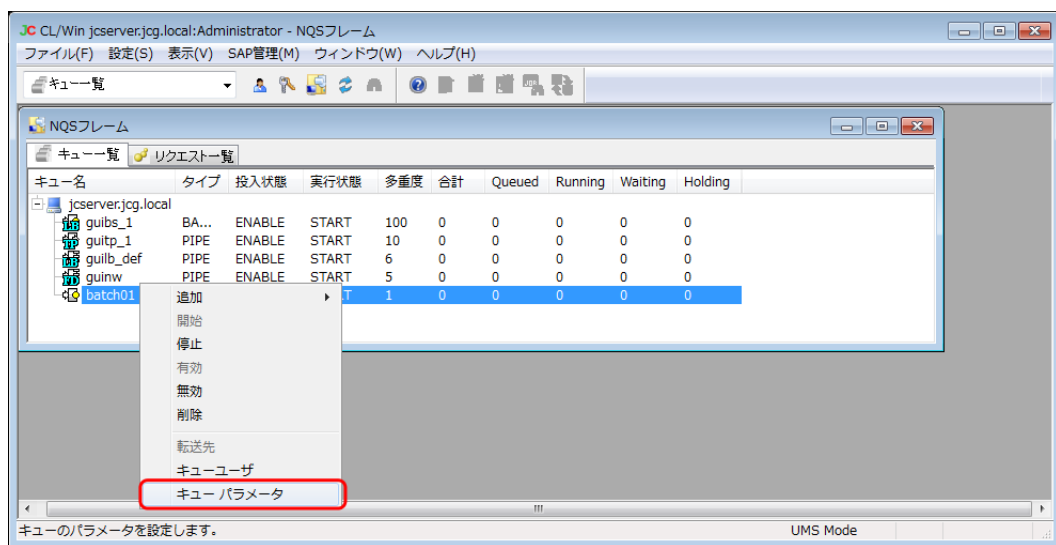


図1.1 バッチキューの [キューパラメータ] 選択画面例

3. [バッチキューパラメータ] ダイアログが表示されるので、[キュー] タブ [キュー属性] - [Custom] を [ON] に設定します。

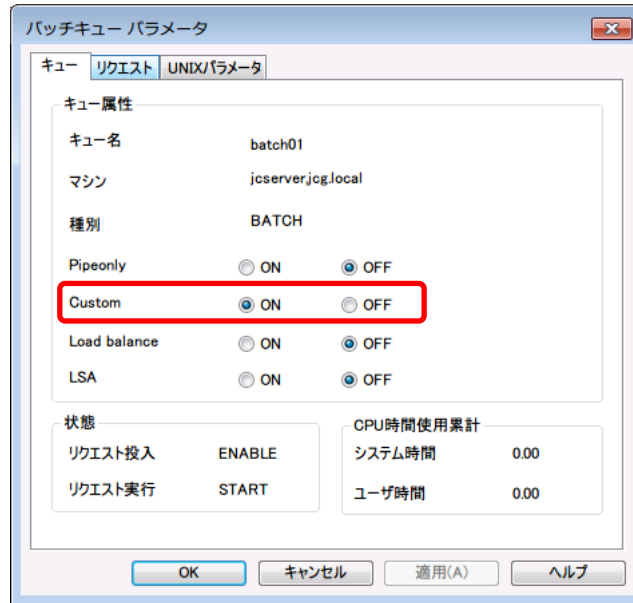


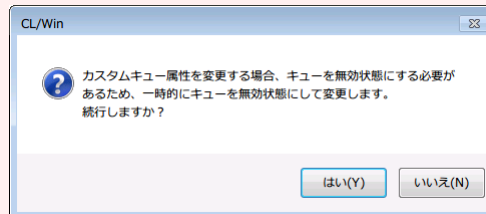
図1.2 バッチキューパラメータの設定画面例

4. 設定後、[OK] ボタンをクリックします。



カスタムキュー属性を変更する場合、キューを無効状態にする必要があります。キューを有効状態のまま変更する場合は、次のような警告ダイアログが表示されます。続行すると、内部動作として一時的にキューを無効化したうえでカスタムキュー属性を変更し、キューを有効状態に戻します。

本ダイアログが表示された場合は、設定変更するキューにジョブが投入されないことを事前に確認してから続行してください。



1.2.2. カスタムジョブテンプレートのカスタマイズ

Systemグループ配下の部品のテンプレートについては製造元から配布されたものであり、スクリプト、パラメータの変更ができません。変更する場合にはUserグループ配下に定義をコピーしてからご利用ください。

1.2.3. 暗号化パラメータの利用

拡張カスタムジョブ部品ではカスタムパラメータの設定時に、あらかじめJobCenterに登録したパラメータへの展開を指定する事ができます。

JobCenterへ登録したパラメータは、JobCenter内部では暗号化して保存し、カスタムパラメータの展開も実行時に行うため、パスワードなどの秘匿情報の設定はこの暗号化パラメータの利用を推奨いたします。暗号化パラメータの登録方法については、<コマンドリファレンス>の「3.29 paramctrl 暗号化DBへのパラメータ登録」を参照してください。

なお、暗号化パラメータの登録は、拡張カスタムジョブ部品を配置したジョブネットワークフローを持つサイトで行ってください。

1.2.3.1. 暗号化パラメータの展開指定方法

暗号化パラメータを拡張カスタムジョブ部品の実行時に展開するには、カスタムパラメータに「\$key\$」(keyは展開したい暗号化パラメータのキー名)という形式で暗号化パラメータのキーを指定します。

変数	値	説明
file_include		【任意】標準出力、標準エラー出力のファイル名
mail_bcc		【任意】メールのBCCの送信先アドレス
mail_cc		【任意】メールのCCの送信先アドレス
mail_from		【必須】メールの送信元アドレス
mail_password	\$jcmgr_password\$	【任意】SMTP認証のパスワード
mail_port		【任意】メールサーバーのSMTPポート番号
mail_server		【必須】メールサーバーを指定してください
mail_subject		【必須】メールの件名を入力してください
mail_to		【必須】メールの送信先アドレス
mail_user	jcmgr	【任意】SMTP認証のユーザー名
ssl_enable		【任意】SSLを使って送信するかどうか

説明
【任意】SMTP認証のパスワードを入力(例:password)

新規 編集 削除

アイコン設定
アイコン種別 Wave-MAIL-send

バージョン情報
更新日時 2017/05/24 08:27:49

OK キャンセル 適用(A) ヘルプ

図1.3 暗号化パラメータの展開指定例



暗号化パラメータを展開する場合は、そのカスタムパラメータの値には「\$key\$」のみを指定してください。「\$key\$」の前後に文字を含めた場合、暗号化パラメータの展開の指定とみなされません。

1.2.4. ジョブネットワークの環境変数の利用

拡張カスタムジョブ部品ではカスタムパラメータの設定時に、ジョブネットワークの環境変数の値を参照することができます。

変数継承機能と組み合わせて、カスタムパラメータの値を拡張カスタムジョブ部品の実行時に動的に決定することも可能です。変数継承機能については、<基本操作ガイド>の「4.2.3 単位ジョブ間で情報を引き継ぐ」を参照してください。

また、参照した環境変数の値が「\$key\$」(keyは展開したい暗号化パラメータのキー名)となっている場合、それを暗号化パラメータの展開の指定であるとみなして暗号化パラメータの値に展開されます。このように環境変数を経由することで、展開する暗号化パラメータのキー名を拡張カスタムジョブ部品の実行時に動的に決定することが可能です。

1.2.4.1. ジョブネットワークの環境変数の参照方法

ジョブネットワークの環境変数の値を拡張カスタムジョブ部品の実行時に参照するには、カスタムパラメータに「%name%」(nameは参照したい環境変数の名前)という形式で環境変数名を指定します。

カスタムジョブパラメータ設定/メール送信

実行設定 UNDIパラメータ 結果 クリティカルポイント警告 その他 カスタムパラメータ設定

変数一覧

変数	値	説明
file_include		[任意]標準出力、標準エラー...
mail_bcc		[任意]メールのBCCの送信...
mail_cc		[任意]メールのCCの送信...
mail_from		[必須]メールの送信元アドレス...
mail_passwd	\$jcmer_password\$	[任意]SMTP認証のパスワード...
mail_port		[任意]メールサーバのSMTPポート...
mail_server		[必須]メールサーバを指定...
mail_subject	%mail_subject%	[必須]メールの件名を入力...
mail_to		[必須]メールの送信先アドレス...
mail_user	jcmer	[任意]SMTP認証のユーザー名...
ssl_enable		[任意]SSLを使って送信する...

説明

[必須]メールの件名を入力(例:完了通知)

新規 編集 削除

アイコン設定

アイコン種別 Wave-MAIL-send

バージョン情報

更新日時 2017/05/24 08:27:49

OK キャンセル 適用(A) ヘルプ

図1.4 ジョブネットワークの環境変数の参照例



■ジョブネットワークの環境変数を参照する場合は、そのカスタムパラメータの値には「%name%」のみを指定してください。「%name%」の前後に文字を含めた場合、環境変数の参照の指定とみなされません。

■環境変数を経由して暗号化パラメータを展開する場合は、参照する環境変数の値が「\$key\$」のみとなるようにしてください。「\$key\$」の前後に文字を含めた場合、暗号化パラメータの展開の指定とみなされません。

1.2.5. 表示アイコン

拡張カスタムジョブの表示アイコンは、「1.1 適用方法」の「アップロード」操作により、対象サーバに配置されます。

追加された表示アイコンは、CL/WinおよびCL/Webによる対象サーバへの次のログイン時から反映されます。

1.2.6. 注意事項

■カスタムパラメータの設定時に入力できない文字があります。

入力できない文字を設定する必要がある場合には、暗号化パラメータ設定を行って変数として使用してください。



■カスタムパラメータ設定では以下の3種類の文字は使用できません。

" # '

■拡張カスタムジョブを実行したとき、以下のようなカスタムパラメータの値は未指定として扱われます。

- カスタムパラメータの設定時に暗号化パラメータを展開するように設定しているが、拡張カスタムジョブの実行時に指定した暗号化パラメータが存在しなかった場合
- カスタムパラメータの設定時に環境変数の値を参照するように設定しているが、拡張カスタムジョブの実行時に指定した環境変数が存在しなかった場合

■R15.3以降では、カスタムパラメータの設定時に環境変数の値を参照するように設定したとき、参照した環境変数の値が「\$key\$」のように「\$」で始まり「\$」で終わる値となっている場合、自動的に暗号化パラメータの値に展開されます。

参照する環境変数の値としてこのような値が想定される場合、先に環境変数の値を暗号化パラメータとして登録し、環境変数を経由して暗号化パラメータの展開を行うようにしてください。

2. メール送信部品

メール送信部品について説明します。

2.1. メール送信部品

2.1.1. 部品情報

■格納パス
/Root/System/Tools/Mail

■定義名
send_mail

■表示アイコン



2.1.2. スクリプト

メール本文として記載したい内容を記述します。

カスタムジョブパラメータでfile_includeを設定している場合、{{{file_include}}}を記述することで内容が展開されます。

また、環境変数についてもメール本文中に展開することができます。利用する場合は{{#getenv}}環境変数名{/getenv}}と記述してください。



{{{file_include}}} で展開する標準出力・標準エラー出力が巨大なサイズになっている場合、SMTPサーバに設定されているメッセージサイズの上限を超える場合があります。

巨大な標準出力・標準エラー出力データの出力は、CL/Winでのトラック表示時に時間がかかる等の影響もあるため、単位ジョブスクリプト内で他のファイルに(アプリケーションやコマンドのログとして)リダイレクトするなどして巨大な出力とならないようご注意ください。

2.1.3. カスタムパラメータ設定

必須	変数	説明	設定例
	file_include	JNWの同一階層にある他の部品の標準出力、標準エラー出力をメールに取り込んで送信する場合に使用します。スクリプトに{{{file_include}}}と記載することで内容が展開されます。ジョブ名.oで標準出力、ジョブ名.eで標準エラー出力を読み込みます。	JOB1.o 処理1.e
	mail_bcc	BCCとして送信アドレスを指定します。複数指定する場合は半角スペース、;(セミコロン)または,(カンマ)区切りで指定します。	admin@mail.local user1@mail.local, user2@mail.local
	mail_cc	CCとして送信アドレスを指定します。複数指定する場合は半角スペース、;(セミコロン)または,(カンマ)区切りで指定します。	admin@mail.local user1@mail.local, user2@mail.local
○	mail_from	Fromアドレスを指定します。	admin@mail.local
	mail_passwd	SMTPサーバに認証が必要な場合のパスワードを指定します。暗号化パラメータを参照する場合は、\$変数名\$を指定します。	PASS \$USER_PASSWD\$
	mail_port	SMTPサーバのポート番号を指定します。空欄の場合は25を使用します。	25 465

必須	変数	説明	設定例
○	mail_server	SMTPサーバ名を指定します。	smtp.server.local 127.0.0.1
○	mail_subject	メールの件名を指定します。	テスト 処理結果通知
○	mail_to	TOとして送信アドレスを指定します。複数指定する場合は半角スペース、;(セミコロン)または,(カンマ)区切りで指定します。	admin@mail.local user1@mail.local, user2@mail.local
	mail_user	SMTPサーバに認証が必要な場合のユーザ名を指定します。暗号化パラメータを参照する場合は、\$変数名\$を指定します。	USER \$USER_NAME\$
	ssl_enable	SMTPサーバとの通信にSSLによる暗号化が必要な場合はtrueを指定します。空欄またはtrue以外を指定した場合はfalseとして扱います。なお対応しているプロトコルはSMTPsになります。STARTTLSには対応しておりません。	true false
	transfer_encoding	メール本文のエンコード方式を '8bit' または 'base64' で指定します。空欄または'8bit'を指定した場合、メール本文はUTF-8でエンコードしたメッセージを送信します。'base64'を指定した場合、メール本文はUTF-8でエンコードした文字列をbase64変換したメッセージを送信します。	8bit base64

2.1.4. 実行結果

メール本文として送信した内容が出力結果に表示されます。異常が発生した場合もエラーメッセージが出力結果に表示されます。

2.1.5. 主要メッセージ

メッセージ	内容
parameter [mail_XXX] is required.	必須パラメータが指定されていない場合に出力されます。
XXX: Not number	ポート番号の指定で数字以外を指定した場合に出力されます。
XXX: No such file or directory	file_includeの指定でファイルを読み込むことができなかった場合に出力されます。
host or service not provided, or not known	メールサーバの指定が間違っている場合に出力されます。
connection refused	メールサーバの指定が間違っている、もしくはポート番号の指定が間違っている場合に出力されます。
timeout	メールサーバの指定が間違っている、もしくはメールサーバが応答しない場合に出力されます。
wrong version number	ssl_enableの指定で、設定が不要なメールサーバに対して設定した場合に出力されることがあります。

2.1.6. 注意事項

■アップロードした部品(Systemグループ配下にある部品)は、スクリプト(メール本文)を編集することができません。Userグループ配下に必要となる定型文の数だけコピーした後、スクリプト編集してご利用ください。

■SMTP認証ではplainとloginのみをサポートしています。

3. HTTP接続部品

HTTP接続部品について説明します。

HTTP接続部品は、Get、Put、Post、Delete、Head および Patch の各HTTPメソッドに対応するテンプレート群で構成される機能です。各HTTP接続部品は、ユーザ認証やProxy設定等の機能を持ち、任意のサービスへのRESTリクエスト発行に利用することができます。

3.1. HTTP接続部品共通仕様

各HTTP接続部品で共通の設定方法、およびパラメータについて説明します。

各HTTP接続部品の設定は、部品そのものに対する制御パラメータと、HTTPリクエストに含めるパラメータによって設定方法が異なります。 部品の制御パラメータはカスタムパラメータ設定画面から、HTTPリクエストパラメータは部品のスクリプト画面から設定します。

3.1.1. リクエスト種別(HTTPメソッド)

HTTP接続部品のリクエスト種別(HTTPメソッド)は、対応する部品ごとに固有です。 HTTPサーバに送信したいリクエスト(HTTPメソッド)に合わせて、Get、Put、Post、Delete、Head および Patch の各HTTPメソッドに対応するテンプレートを選択し、 制御パラメータおよびHTTPリクエストパラメータの設定を行って下さい。

3.1.2. 制御パラメータ(カスタムパラメータ)一覧

各HTTP接続部品で設定可能なカスタムパラメータは以下のとおりです。 各パラメータの詳細は、「[3.1.2.1 基本パラメータ](#)」～「[3.1.3 HTTPリクエストパラメータ](#)」で説明します。

必須	変数	説明	設定例
○	url	接続先のURL	http://www.nec.com/
	expected_status	正常終了とみなすHTTPステータスコード	2xx 302
	retry_count	サーバ接続エラー時のリトライ回数	5
	retry_interval	サーバ接続エラー時リトライの実行間隔 (秒)	3
	tls_version	HTTPS接続時のTLSバージョン	1.1
	request_body_file	リクエストボディ格納ファイルパス	/var/app/request.json
	response_header_file	レスポンスヘッダ情報格納ファイルパス	/var/app/response_header.txt
	response_body_file	レスポンスボディ格納ファイルパス	/var/app/output.json
	response_body_stdout	レスポンスボディの標準出力への出力制限 (bytes)	1024
	proxy_server	プロキシサーバ	http:// proxy.example.com:8080
	proxy_username	プロキシサーバの認証ユーザ名	johndoe
	proxy_password	プロキシサーバの認証パスワード	some-secret-password
	authenticate	HTTP接続先の認証方式	oauth2
	auth_username	HTTP接続先の認証ユーザ名	johndoe
	auth_password	HTTP接続先の認証パスワード	some-secret-password
	auth_token	認証トークン	some-authenticate- token
	oauth_realm	OAuth認証のrealm	some-oAuth-realm
	oauth_consumer_key	OAuth認証のコンシューマキー	some-oauth-consumer- key
	oauth_consumer_secret	OAuth認証のコンシューマキーシークレット	some-oAuth-consumer- key-sercret
	oauth_grant_type	OAuth2認証フロー種別	client_credentials
	oauth_token_endpoint	OAuth2認証エンドポイント	https:// api.example.com/ oauth2/endpoints/ authorize

3.1.2.1. 基本パラメータ

HTTP接続部品の基本的動作を制御するパラメータについて説明します。

■url

接続先のURLを指定します。本パラメータの指定は必須であり、以下の形式で指定する必要があります。

`http(s)://<host>[:<port>]/<path>`



クエリパラメータはURLではなく、スクリプトで指定します。詳細は「[3.1.3.1 URLクエリパラメータ](#)」を参照してください。

■retry_count

サーバ接続エラー時のリトライ回数を指定します。retry_interval と同時に指定し、いずれかのパラメータが省略されている場合はリトライ動作を行いません。

ネットワーク到達不可の場合、コネクション確立エラーの場合、および HTTPサーバからのレスポンスコードが 503 の場合、本パラメータで指定した回数までHTTPリクエストの発行を再実行します。

■retry_interval

サーバ接続エラー時のリトライ間隔を秒単位で指定します。retry_count と同時に指定し、いずれかのパラメータが省略されている場合はリトライ動作を行いません。

retry_count によるHTTPリクエスト再発行を、本パラメータで指定した秒数の間隔を空けてから行います。ただし、HTTPサーバからのレスポンスコードが 503 でリトライを行う場合で、レスポンスヘッダに Retry-After が設定されている場合は Retry-After の指定が優先されます。

■expected_status

正常終了とみなすHTTPステータスを指定します。

HTTPステータスは ' '(半角スペース)、','(カンマ)、';'(セミコロン) で区切って複数のステータスを指定可能です。また、任意の10進数1文字のワイルドカードとして 'x'(小文字エックス) を指定することが可能です。



expected_status の設定例は次のようになります。

200 および 201 を正常終了とみなす場合: '200 201'

200番台を正常終了とみなす場合: '2xx'

■tls_version

HTTPS接続時のTLSバージョンを指定します。パラメータ省略時は TLS 1.2 での通信を行います。

指定可能な値は 1.0 / 1.1 / 1.2 です。

3.1.2.2. HTTP Proxyパラメータ

HTTP Proxy 経由で接続する場合に指定するパラメータについて説明します。

■proxy_server

ProxyサーバのURLを指定します。パラメータ省略時は、Proxyサーバ経由の接続を行いません。以下の形式で指定する必要があります。

`http(s)://<host>[:<port>]`

■proxy_username

Proxyサーバの認証ユーザ名を指定します。認証なしProxyサーバを利用する場合は、本パラメータの設定は必要ありません。

■proxy_password

Proxyサーバの認証パスワードを指定します。認証なしProxyサーバを利用する場合は、本パラメータの指定は必要ありません。

3.1.2.3. HTTP認証パラメータ

HTTP 接続先との認証を指定するパラメータについて説明します。

■authenticate

HTTP認証の方式を指定します。パラメータ省略時はHTTP認証処理は行いません。 指定可能な認証方式は、以下のとおりです。

- basic
- digest
- bearer
- oauth
- oauth2

■auth_username

認証ユーザ名を指定します。認証方式が basic, digest の場合は指定が必須です。また、認証方式が oauth2 かつ 認証フロー種別が password の場合は指定が必須です。

■auth_password

認証パスワードを指定します。認証方式が basic, digest 場合は指定が必須です。また、認証方式が oauth2 かつ 認証フロー種別が password の場合は指定が必須です。

■auth_token

認証トークンを指定します。認証方式が bearer の場合は指定が必須です。

■oauth_realm

OAuth認証のrealmを指定します。パラメータ省略時はrealmを空文字列として認証処理を行います。

■oauth_consumer_key

OAuth認証のコンシューマキーを指定します。認証方式が oauth, oauth2 の場合は指定が必須です。

■oauth_consumer_secret

OAuth認証のコンシューマキーシークレットを指定します。認証方式が oauth, oauth2 の場合は指定が必須です。

■oauth_grant_type

OAuth2認証フロー種別を指定します。認証方式が oauth2 の場合は指定が必須です。 指定可能なフロー種別は、以下のとおりです。

- password

- client_credentials

- oauth_token_endpoint

OAuth2トークンエンドポイントのURLを指定します。認証方式が oauth2 の場合は指定が必須です。



HTTP接続部品は、トークンエンドポイントから取得するトークン情報がJSON形式であり、アクセストークンは access_token というメンバー名で参照できることを前提として動作します。

3.1.2.4. 入出力パラメータ

HTTP接続部品の入力または出力に関わるパラメータについて説明します。

- request_body_file

HTTPリクエストのメッセージボディとして送信するファイルを絶対パスで指定します。パラメータ省略時は、メッセージボディは設定されません。

- response_header_file

HTTPレスポンスのヘッダ情報を保存するファイルを絶対パスで指定します。パラメータ省略時は、ヘッダ情報のファイルへの出力は行いません。また、指定ファイルが存在していた場合は上書きします。ヘッダ情報は、以下の形式で指定ファイルに保存されます。

- response_body_file

HTTPレスポンスのメッセージボディを保存するファイルを絶対パスで指定します。パラメータ省略時は、メッセージボディのファイルへの出力は行いません。また、指定ファイルが存在していた場合は上書きします。

- response_body_stdout

HTTPレスポンスのメッセージボディを標準出力に出力する最大量(bytes)を指定します。パラメータ省略時は、メッセージボディの標準出力への出力は行いません。

3.1.3. HTTPリクエストパラメータ

HTTPリクエストパラメータの設定方法を説明します。

本項目における HTTPリクエストパラメータ は、以下の2種類を指します。

- URLクエリパラメータ

- HTTPヘッダパラメータ

3.1.3.1. URLクエリパラメータ

URLクエリパラメータは、HTTPサーバに情報を送るために、以下のように **変数名=値** の形式でURL末尾に付け加える情報です。

```
http://api.example.com/foo/var.cgi?param1=value1&param2=value2
```

HTTP接続部品でURLクエリパラメータを付与する場合は、スクリプト編集画面に一行一件で **変数名=値** の形式で記述します。

```
param1 = value1
```

```
param2 = value2
```



パラメータ名および値は、それぞれの前後空白文字を取り除いた文字列が設定されます。

パラメータ名および値は、HTTPリクエスト発行時にURLエンコードされます。スクリプト編集画面での入力値は、URLエンコードされていない値を記述して下さい。

同名パラメータが複数回記述された場合、最後に設定されたパラメータ値が設定されます。

3.1.3.2. HTTPヘッダパラメータ

HTTPヘッダパラメータは、HTTPサーバに情報を送るために、HTTPリクエストヘッダに以下のように **変数名: 値** の形式で設定される情報です。

```
param1: value1
param2: value2
```

HTTP接続部品は、標準で以下のHTTPヘッダを設定します。"で囲む表記は、固定の値が設定されることを示します。

ヘッダ名	値
Host	接続先のホスト名およびポート番号を設定します。
User-Agent	'JobCenter HTTP Connection Job'
Accept	'*/*'
Content-Length	request_body_file を指定している場合、当該ファイルのサイズを自動的に設定します。未指定の場合はこのヘッダは設定されません。
Authorization	authorization が basic / digest / bearer の場合、認証情報を設定します。

これらのHTTPヘッダを変更、または新しいHTTPヘッダを追加する場合は、スクリプト編集画面に一行一件で **Header = 変数名: 値** の形式で記述します。

```
Header = param1 : value1
Header = param2 : value2
```



パラメータ名および値は、それぞれの前後空白文字を取り除いた文字列が設定されます。

同名パラメータが複数回記述された場合、最後に設定されたパラメータ値が設定されます。

3.1.3.3. HTTPリクエストパラメータの環境変数展開

HTTPリクエストパラメータには環境変数を利用することができます。利用方法はスクリプト編集画面に「{{{環境変数名}}}'という記法を用いることで環境変数値の展開ができます。

3.1.3.3.1. HTTPリクエストパラメータの環境変数展開の利用例

環境変数とスクリプト編集画面が以下の設定がされている場合の例を記載します。

環境変数名	環境変数値
request_id	req-99999

環境変数名	環境変数値
HeaderAccept	Header = Accept : application/json

```
{{HeaderAccept}}
request_id={{request_id}}
```

設定されている環境変数を参照し展開するため、スクリプト編集画面で以下を記載したことと同じ意味になります。

```
Header = Accept : application/json
request_id=req-99999
```



スクリプト編集画面に記載した値が環境変数に存在しなかった場合は空文字列に置き換わります。



{ および } を環境変数展開と認識させないようにするエスケープ記法は用意しないため、パラメータ自体に { や } を入れたい場合には、環境変数の値に { や } を指定する必要があります。

3.1.4. 実行結果

HTTP接続部品の標準出力は、以下の形式で出力されます。<>で囲む表記は、実行時の値が反映されます。

```
--- General -----
Request URL: <リクエスト先URL>
Request Method: <リクエストメソッド>
Status Code: <レスポンスコード>
--- HTTP response headers -----
<ヘッダ名1>: <ヘッダ値1>
<ヘッダ名2>: <ヘッダ値2>
      : (設定されたヘッダを全て表示)
--- HTTP response body -----
<レスポンスボディーの内容>
```



response_body_stdout パラメータが未指定、または値が0の場合、HTTP response body 行以降は出力されません。

レスポンスボディーの内容は、response_body_stdout パラメータで指定したbyte数以内で表示されます。

レスポンスボディーの内容は、文字コード変換の処理は行われませんので、マルチバイト文字を含む内容は文字化けが発生する場合があります。



レスポンスボディーの内容は、制御文字を'?'(クエスション)に置換した内容で表示されます。対象となる制御文字は次のとおりです。

0x00 (NUL)	0x08 (BS)	0x12 (DC2)	0x1A (SUB)
0x01 (SOH)	0x09 (TAB)	0x13 (DC3)	0x1B (ESC)

0x02 (STX)	0x0B (VT)	0x14 (DC4)	0x1C (FS)
0x03 (ETX)	0x0C (FF)	0x15 (NAK)	0x1D (GS)
0x04 (EOT)	0x0E (SO)	0x16 (SYN)	0x1E (RS)
0x05 (ENQ)	0x0F (SI)	0x17 (ETB)	0x1F (US)
0x06 (ACK)	0x10 (DLE)	0x18 (CAN)	0x7F (DEL)
0x07 (BEL)	0x11 (DC1)	0x19 (EM)	

3.1.5. 主要メッセージ

メッセージ	内容
Network is unreachable	接続先のHTTPサーバ名、もしくはポート番号の指定が誤っている場合に出力されます。 retry_count および retry_interval のパラメータを指定している場合、再送処理が行われます。
Service Temporarily Unavailable	接続先HTTPサーバから、503エラーが返却された場合に出力されます。
invalid "url" parameter.	url パラメータが指定されていない場合に出力されます。
Unsupported grant_type	認証方式に oauth2 を指定している時、oauth_grant_type パラメータにHTTP接続部品が対応していないOAuth2認証フロー種別を指定した場合に出力されます。
Failed to get access token	認証方式に oauth2 を指定している時、OAuth2アクセストークンが取得できなかった場合に出力されます。

3.1.6. 注意事項

アップロードした部品(Systemグループ配下にある部品)は、スクリプト(HTTPリクエストパラメータ)を編集することができません。 定型のパラメータを設定した部品を作成する場合は、Userグループ配下に必要となる数だけコピーした後、スクリプト編集してご利用ください。

3.2. HTTP-GET部品

3.2.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpGet

■表示アイコン



HTTP-GET 部品の詳細は、「[3.1 HTTP接続部品共通仕様](#)」を参照してください。

3.3. HTTP-PUT部品

3.3.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpPut

■表示アイコン



HTTP-PUT 部品の詳細は、「[3.1 HTTP接続部品共通仕様](#)」を参照してください。

3.4. HTTP-POST部品

3.4.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpPost

■表示アイコン



HTTP-POST 部品の詳細は、「[3.1 HTTP接続部品共通仕様](#)」を参照してください。

3.5. HTTP-DELETE部品

3.5.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpDelete

■表示アイコン



HTTP-DELETE 部品の詳細は、[「3.1 HTTP接続部品共通仕様」](#)を参照してください。

3.6. HTTP-HEAD部品

3.6.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpHead

■表示アイコン



HTTP-HEAD 部品の詳細は、[「3.1 HTTP接続部品共通仕様」](#)を参照してください。

3.7. HTTP-PATCH部品

3.7.1. 部品情報

■格納パス
/Root/System/Tools/HTTP

■定義名
HttpPatch

■表示アイコン



HTTP-PATCH 部品の詳細は、[「3.1 HTTP接続部品共通仕様」](#)を参照してください。

4. Amazon Web Services 連携部品

Amazon Web Services(以下、AWSと記します)連携部品について説明します。

4.1. EC2

Elastic Compute Cloudの部品について説明します。

4.1.1. EC2部品共通のカスタムパラメータ

EC2部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.1.1.1. カスタムパラメータ設定

必須	変数	説明	設定例
	AccessKeyID	APIを利用するためのアクセスキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	APIを利用するためのシークレットキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfk8klsjdf23oisAKew12
○	endpoint	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、ec2.ap-northeast-1.amazonaws.comが指定されています。	ec2.ap-northeast-1.amazonaws.com
	proxy	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.nichiden.co.jp:8080

4.1.2. インスタンス起動

4.1.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
StartInstances

■表示アイコン



4.1.2.2. 部品説明

インスタンスを起動します。

対象となるインスタンスは、インスタンスIDでの指定の他、タグで検索することができます。

4.1.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.2.4. カスタムパラメータ設定

必須	変数	説明	設定例
	DryRun	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
△	InstanceId	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはtagの指定が必要です。	i-123456ab i-123456ab;i-45678cd
△	tag	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはInstanceIdの指定が必要です。	AutoShutDown=ON
	interval	EC2インスタンスの状態確認を行う間隔を秒単位で指定します。デフォルト値として5が指定されています。また、空欄の場合は30として扱います。	10
	timeout	EC2インスタンスの起動完了の待ち合わせを行う際のタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。	30

必須	変数	説明	設定例
	StatusCheck	EC2インスタンスがrunningの状態になりステータスチェックが完了するまでジョブの終了を待ち受ける場合はtrueを指定します。	true false

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.1.2.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

インスタンス開始対象となるインスタンスを特定するために、DescribeInstancesを実行し、DescribeInstancesResponseに格納されます。インスタンス開始結果は、StartInstancesResponseに格納されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.1.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.1.2.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	StartInstances
EC2	DescribeInstanceStatus

4.1.2.8. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■Instanceld、tagを両方を指定した場合はANDとして解釈されます。

4.1.3. インスタンス停止

4.1.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
StopInstances

■表示アイコン



4.1.3.2. 部品説明

インスタンスを停止します。

対象となるインスタンスは、インスタンスIDでの指定の他、タグで検索することができます。

4.1.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.3.4. カスタムパラメータ設定

必須	変数	説明	設定例
	DryRun	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
△	InstanceId	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはtagの指定が必要です。	i-123456ab i-123456ab;i-45678cd
△	tag	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータまたはInstanceIdの指定が必要です。	AutoShutDown=ON
	interval	EC2インスタンスの状態確認を行う間隔を秒単位で指定します。デフォルト値として5が指定されています。また、空欄の場合は30として扱います。	10
	timeout	EC2インスタンスの停止完了の待ち合わせを行う際のタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。	30

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.1.3.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

インスタンス停止対象となるインスタンスを特定するために、DescribeInstancesを実行し、DescribeInstancesResponseに格納されます。インスタンス停止結果は、StopInstancesResponseに格納されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.1.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.1.3.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	StopInstances
EC2	DescribeInstanceStatus

4.1.3.8. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■InstanceId、tagを両方を指定した場合はANDとして解釈されます。

4.1.4. インスタンス情報取得

4.1.4.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
DescribeInstances

■表示アイコン



4.1.4.2. 部品説明

インスタンスの情報取得を行います。

インスタンスIDまたはタグを指定することで、取得対象となるインスタンスを絞り込むことができます。

4.1.4.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには`{{変数名}}`を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、`{{#セクション名}}{{変数名}}`と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.4.4. カスタムパラメータ設定

必須	変数	説明	設定例
	InstanceId	EC2インスタンスIDを指定します。複数入力する場合セミコロン ; 区切りで指定します。	i-123456ab i-123456ab;i-45678cd
	tag	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン ; 区切りで指定します。	AutoShutDown=ON
	noTag	EC2インスタンスのタグを指定します。検索結果を、指定したタグを持たない、または持っていない値が空であるインスタンスに絞り込みます。複数入力する場合セミコロン ; 区切りで指定します。	Owner AutoShutDown; AutoBoot

4.1.4.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DescribeInstancesResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.1.4.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。

メッセージ	内容
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.1.4.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances

4.1.4.8. 注意事項

- テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。
- InstanceId、tagを両方指定した場合はANDとして解釈されます。
- noTagでタグを複数指定した場合はORとして解釈されます。

4.1.5. インスタンスプロパティ変更

4.1.5.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
ModifyInstanceAttribute

■表示アイコン



4.1.5.2. 部品説明

インスタンスのインスタンスタイプを変更します。

変更対象のインスタンスは、インスタンスIDで指定することができます。

4.1.5.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.5.4. カスタムパラメータ設定

必須	変数	説明	設定例
	DryRun	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	InstanceId	EC2インスタンスIDを指定します。	i-123456ab
○	InstanceType.Value	EC2インスタンスのタイプを指定します。	t2.micro

4.1.5.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、ModifyInstanceAttributeResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.1.5.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.1.5.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	ModifyInstanceAttribute

4.1.5.8. 注意事項

特になし

4.1.6. AMI作成

4.1.6.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
CreateImage

■表示アイコン



4.1.6.2. 部品説明

AMIを作成します。

対象となるインスタンスは、インスタンスIDで指定することができます。

4.1.6.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには`{{変数名}}`を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、`{#セクション名}{{変数名}}{/セクション名}`と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.6.4. カスタムパラメータ設定

必須	変数	説明	設定例
	DryRun	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	InstanceId	EC2インスタンスIDを指定します。	i-123456ab
○	Name	AMIの名前を指定します。マルチバイト文字は指定できません。	Test AMI

4.1.6.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、CreateImageResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.1.6.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.1.6.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	CreateImage

4.1.6.8. 注意事項

特になし

4.1.7. 世代管理付きAMI作成

4.1.7.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
CreateImageWithGeneration

■表示アイコン



4.1.7.2. 部品説明

AMIを作成します。

対象となるインスタンスは、インスタンスIDでの指定の他、タグで検索することができます。

この部品で作成されたAMIには世代管理用のタグが付与され、世代数の古いAMIを削除することができます。

4.1.7.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{変数名}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{変数名}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.7.4. カスタムパラメータ設定

必須	変数	説明	設定例
	DryRun	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
△	InstanceId	EC2インスタンスIDを指定します。複数入力する場合セミコロン ; 区切りで指定します。 このパラメータまたはInstancetagの指定が必要です。	i-123456ab i-123456ab;i-45678cd
△	Instancetag	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン ; 区切りで指定します。 このパラメータまたはInstanceIdの指定が必要です。	AMIBackup=Yes
	Description	AMIの概要説明を指定します。マルチバイト文字は指定できません。 本パラメータは以下のマクロに対応しています。 パラメータ中にマクロ文字列を記述した場合、対応する値に置換されます。 %Y : 西暦(4桁)	CreateImage %Y/%m/%d %H:%M: %S(%i)

必須	変数	説明	設定例
		%m : 月(2桁) %D : 日(2桁) %H : 時間(2桁) %M : 分(2桁) %S : 秒(2桁) %i : 作成元となるECインスタンスのインスタンスID %% : %自身(エスケープ)	
○	generation	AMIの世代数を指定します。デフォルト値として1が指定されています。 0を指定した場合は古い世代のAMIの削除を行いません。	5
	managementName	世代管理を区別する名前をUTF-8換算で63バイト以内で指定します。 このパラメータは同じインスタンスに対して複数のパターンで世代管理を別々に行う場合に指定します。	daily 月次バックアップ
○	timeout	AMIの作成完了の待ち合わせを行う際のタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。	30
○	NoReboot	AMIを作成する際に作成元のEC2インスタンスを再起動するかどうかを指定します。インスタンスを再起動しない場合はtrue、再起動する場合はfalseを指定します。デフォルト値としてtrueが指定されています。	true

4.1.7.5. 部品詳細

本部品は大きく分けて5つのステップの動作があります。

1. AMI作成対象となるインスタンス一覧の取得
2. AMIの作成要求
3. 作成要求成功したAMIへ世代管理を行うためのタグの付与
4. AMIの作成完了待ち合わせ
5. 旧世代のAMIの削除

本部品の動作の詳細を説明します。

1. InstanceId/パラメータおよびInstancetag/パラメータを用いて、AMI作成対象となるインスタンスの一覧を取得します。

 インスタンスの一覧を取得できなかった場合や、インスタンスがひとつも見つからなかった場合、部品はコード1で終了します。
2. 取得したインスタンス一覧に対してAMIの作成要求を行います。

ここで、AMI名は"YYYYMMDDhhmmss(<インスタンスID>)"という名前が自動的に指定されます。

また、AMIの説明はDescription/パラメータで指定された文字列(マクロ文字列を記述していた場合はマクロ展開後の文字列)が設定されます。

作成要求に成功したAMIの一覧が次の処理の対象となります。

インスタンス一覧のすべてのインスタンスに対してAMIの作成要求が失敗した場合、部品はコード1で終了します。

3. 作成要求に成功したAMI一覧の各AMIに以下のタグを付与します。

■ JCSourceInstanceId=<作成元となったインスタンスのインスタンスID>

このタグは世代管理に必要となります。

■ JCGenerationManagement

■ managementName/パラメータを指定した場合

JCGenerationManagement=Yes:<managementName/パラメータで指定した値>

■ managementName/パラメータを指定しない場合

JCGenerationManagement=Yes

このタグは世代管理に必要となります。

■ Name=<作成元となったインスタンスのNameタグの値>

タグの付与に成功したAMIの一覧が次の処理の対象となります。

AMI一覧のすべてのAMIに対してタグの付与に失敗した場合、部品はコード1で終了します。

4. タグの付与に成功したAMI一覧に対して、作成完了を待ち合わせます。

約30秒間隔でAMIの状態を取得し、作成完了したかどうかをチェックします。

以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

■ AMI一覧のすべてのAMIの状態が"available"または"failed"のいずれかとなったとき

すべてのAMIの状態が"failed"となった場合、部品はコード1で終了します。

■ 待ち合わせ開始時刻からtimeout/パラメータで指定した時間が経過したとき

状態が"available"であるAMIがひとつもなかった場合、部品はコード1で終了します。

■ AMIの状態取得に失敗した場合

この場合、部品はコード1で終了します。

待ち合わせ終了時に、状態が"available"となったAMIの一覧が次の処理の対象となります。

5. generation/パラメータが0よりも大きい場合、AMIの世代管理、すなわち古い世代のAMIの削除を行います。

状態が"available"となったAMI一覧の各AMIに対して、以下の処理を行います。

■ 以下のタグをすべて持つAMIの一覧を取得します。

JCSourceInstanceId=<作成元となったインスタンスのインスタンスID>

- managementName/パラメータを指定した場合

JCGenerationManagement=Yes:<managementName/パラメータで指定した値>

- managementName/パラメータを指定しない場合

JCGenerationManagement=Yes

- 上記のAMI一覧から、状態が"available"であるもののみを取り出し、AMI名を利用して作成日時順にソートします。

- 上記のソートされたAMI一覧のうち、作成日時が新しいものからgeneration/パラメータで指定した数だけを残し、他のAMIを削除します。

また、削除したAMIに紐づいたスナップショットを削除します。

6. AMI作成対象となったインスタンスすべてに対して、AMIの作成処理および世代管理処理が成功した場合、部品はコード0で終了します。

そうでない場合、部品はコード1で終了します。

4.1.7.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、オブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.1.7.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.1.7.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	CreateImage
EC2	DescribeImages
EC2	CreateTags
EC2	DeregisterImage
EC2	DeleteSnapshot

4.1.7.9. 注意事項

- タグの付与に失敗したAMI、および作成に失敗して状態が"failed"となったAMIは、世代管理の対象にはならず残存し続けます。そのようなAMIは手動で削除する必要があります。

4.1.8. タグ作成

4.1.8.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticComputeCloud

■定義名
CreateTags

■表示アイコン



4.1.8.2. 部品説明

EC2の各リソースのタグ作成を行います。

対象となるリソースはリソースIDで指定します。

4.1.8.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには`{{変数名}}`を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、`{{#セクション名}}{{変数名}} {{/セクション名}}`と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.8.4. カスタムパラメータ設定

必須	変数	説明	設定例
	DryRun	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	ResourceId	リソースIDを指定します。複数入力する場合セミコロン ; 区切りを指定します。	snap-123456ab i-123456ab; snap-45678cd
○	Tag	作成するタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン ; 区切りで指定します。	Name=Snapshot1

4.1.8.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、CreateTagsResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.1.8.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	Tagの指定方法が間違っている場合に表示されます。

4.1.8.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	CreateTags

4.1.8.8. 注意事項

特になし

4.1.9. スナップショット作成

4.1.9.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
CreateSnapshot

■表示アイコン



4.1.9.2. 部品説明

EBSボリュームのスナップショットを作成します。

対象となるEBSボリュームは、ボリュームIDでの指定の他、インスタンスID、インスタスタグ、ボリュームタグで検索することができます。

世代数の古いスナップショットを削除することができます。

4.1.9.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}} {{{変数名}}} {{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.9.4. カスタムパラメータ設定

必須	変数	説明	設定例
	DryRun	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	Description	スナップショットの概要説明を指定します。マルチバイト文字は指定できません。 本パラメータは以下のマクロに対応しています。パラメータ中にマクロ文字列を記述した場合、対応する値に置換されます。 %Y：西暦(4桁) %m：月(2桁) %D：日(2桁) %H：時間(2桁) %M：分(2桁) %S：秒(2桁)	CreateSnapshot %Y/%m/%d %H:%M: %S(%i, %v)

必須	変数	説明	設定例
		%i : 作成元となるECインスタンスのインスタンスID(ボリュームがインスタンスにアタッチされていない場合など、インスタンスIDが取得できない場合は空文字列) %v : 作成元となるボリュームのボリュームID %% : %自身(エスケープ)	
△	Volumeld	ボリュームIDを指定します。複数入力する場合セミコロン ; 区切りを指定します。 InstanceId、InstanceTag、Volumeld、Volumetagのいずれか1つを指定する必要があります。	vol-123456ab vol-123456ab; vol-45678cd
△	Volumetag	ボリュームのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン ; 区切りで指定します。 InstanceId、InstanceTag、Volumeld、Volumetagのいずれか1つを指定する必要があります。	Name=Vol1
△	InstanceId	EC2インスタンスIDを指定します。複数入力する場合セミコロン ; 区切りを指定します。 InstanceId、InstanceTag、Volumeld、Volumetagのいずれか1つを指定する必要があります。	i-123456ab i-123456ab;i-45678cd
△	InstanceTag	EC2インスタンスのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン ; 区切りを指定します。 InstanceId、InstanceTag、Volumeld、Volumetagのいずれか1つを指定する必要があります。	AutoShutDown=ON
○	generation	スナップショットの世代数を指定します。デフォルト値として1が指定されています。 0を指定した場合は古い世代のスナップショットの削除を行いません。	5
○	timeout	スナップショットの作成完了の待ち合わせを行う際のタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。	30
○	checkInterval	スナップショットの作成完了の待ち合わせを行う際のステータスチェックの間隔を秒単位で指定します。デフォルト値として30が指定されています。	60

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.1.9.5. 部品詳細

本部品は大きく分けて5つのステップの動作があります。

1. スナップショット作成対象となるEBSボリューム一覧の取得

2. スナップショットの作成要求
3. 作成要求成功したスナップショットへタグの付与
4. スナップショットの作成完了待ち合わせ
5. 旧世代のスナップショットの削除

本部品の動作の詳細を説明します。

1. InstanceId/パラメータ、Instancetag/パラメータ、Volumeld/パラメータ、Volumetag/パラメータのうち、指定されたものを用いて、スナップショット作成対象となるEBSボリュームの一覧を取得します。

EBSボリュームの一覧を取得できなかった場合や、EBSボリュームがひとつも見つからなかった場合、部品はコード1で終了します。

2. 取得したEBSボリューム一覧に対してスナップショットの作成要求を行います。

スナップショットの説明はDescription/パラメータで指定された文字列(マクロ文字列を記述していた場合はマクロ展開後の文字列)が設定されます。

作成要求が成功したスナップショットの一覧が次の処理の対象となります。

EBSボリューム一覧のすべてのEBSボリュームに対してスナップショットの作成要求が失敗した場合、部品はコード1で終了します。

3. 作成要求が成功したスナップショットに以下のタグを付与します。

■Name=<作成元となったEBSボリュームのNameタグの値>

■InstanceName=<作成元となったEBSボリュームをアタッチしているインスタンスのNameタグの値>



■上記タグは部品利用者がスナップショットの識別を行いやすくする目的で付与しています。そのため、付与されたタグを後から編集、削除しても問題ありません。

■EBSボリュームにNameタグが存在しない、またはNameタグの値が空である場合、スナップショットにNameタグは付与されません。

■EBSボリュームをアタッチしているインスタンスが存在しない、インスタンスにNameタグが存在しない、または値が空である場合、スナップショットにInstanceNameタグは付与されません。



タグの付与に失敗した場合、以下の警告メッセージを標準出力に出力します。

Warning: CreateTags request failure. target snapshotId=<スナップショットID>

なお、タグの付与に失敗しても部品は実行を継続します。また、タグの付与の結果は部品の終了コードに影響しません。

4. 作成要求が成功したスナップショット一覧に対して、作成完了を待ち合わせます。

checkInterval/パラメータで指定した秒間隔でスナップショットの状態を取得し、作成完了したかどうかをチェックします。

以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

■スナップショット一覧のすべてのスナップショットの状態が"completed"または"error"のいずれかとなったとき

すべてのスナップショットの状態が"error"となった場合、部品はコード1で終了します。

■待ち合わせ開始時刻からtimeout/パラメータで指定した時間が経過したとき

状態が"completed"であるスナップショットがひとつもなかった場合、部品はコード1で終了します。

■スナップショットの状態取得に失敗した場合

この場合、部品はコード1で終了します。

待ち合わせ終了時に、状態が"completed"となったスナップショットの一覧が次の処理の対象となります。

5. generation/パラメータが0よりも大きい場合、スナップショットの世代管理、すなわち古い世代のスナップショットの削除を行います。

状態が"completed"となったスナップショット一覧の各スナップショットに対して、以下の処理を行います。

■そのスナップショットの作成元となったEBSボリュームから作られたスナップショット一覧を取得します。

■上記のスナップショット一覧から、状態が"completed"であるもののみを取り出し、作成日時順にソートします。

■上記のソートされたスナップショット一覧のうち、作成日時が新しいものからgeneration/パラメータで指定した数だけを残し、他のスナップショットを削除します。



削除しようとしたスナップショットがAMIに使用されている場合、そのスナップショットは削除できません。また、その理由によりスナップショットの削除が行えなかった場合、本部品はそれをエラー扱いしません。

6. スナップショット作成対象となったEBSボリュームすべてに対して、スナップショットの作成処理および世代管理処理が成功した場合、部品はコード0で終了します。

そうでない場合、部品はコード1で終了します。

4.1.9.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

スナップショットの作成対象となるボリュームを特定するために、DescribeInstancesまたはDescribeVolumesを実行しますが、それぞれDescribeInstancesResponse、DescribeVolumesResponseに格納されます。スナップショット作成結果は、JobCenterResultというオブジェクト配下に、VolumeId、CreateSnapshotResponse、DescribeSnapshotsResponseとして格納されます。ローテーション結果として、削除したスナップショットが存在する場合には、jc_deletedという変数がtrueに設定されます。また削除ができなかった場合にはjc_delete_error_messageにエラーメッセージが保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.1.9.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

メッセージ	内容
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.1.9.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeInstances
EC2	DescribeVolumes
EC2	CreateSnapshot
EC2	CreateTags
EC2	DescribeSnapshots
EC2	DeleteSnapshot

4.1.9.9. 注意事項

- Instanceld、Instancetag、Volumeld、Volumetagをいずれか1つではなく、複数指定した場合は Instanceld、Instancetagの設定が優先されます。Id、tagを両方を指定した場合はANDとして解釈されま
す。
- テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多
すぎる場合には表示されない場合があります。

4.1.10. スナップショット削除

4.1.10.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
DeleteSnapshot

■表示アイコン


4.1.10.2. 部品説明

スナップショットを削除します。

削除対象となるスナップショットは、スナップショットIDで指定することができます。

4.1.10.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには`{{変数名}}`を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、`{#{セクション名}}
{{変数名}}
{/セクション名}}`と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.10.4. カスタムパラメータ設定

必須	変数	説明	設定例
	DryRun	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	SnapshotId	スナップショットIDを指定します。	snap-123456ab

4.1.10.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DeleteSnapshotResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.1.10.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.1.10.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DeleteSnapshot

4.1.10.8. 注意事項

特になし

4.1.11. スナップショット情報取得

4.1.11.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
DescribeSnapshots

■表示アイコン


4.1.11.2. 部品説明

スナップショットの情報取得を行います。

スナップショットIDまたはタグを指定することで、取得対象となるスナップショットを絞り込むことができます。

4.1.11.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.11.4. カスタムパラメータ設定

必須	変数	説明	設定例
	SnapshotId	スナップショットIDを指定します。複数入力する場合セミコロン；区切りを指定します。	snap-123456ab snap-123456ab; snap-45678cd
	tag	スナップショットのタグを タグ名=値 の形式で指定します。複数入力する場合セミコロン；区切りで指定します。	Name=Snapshot1

4.1.11.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DescribeSnapshotsResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.1.11.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。

メッセージ	内容
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

4.1.11.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeSnapshots

4.1.11.8. 注意事項

- テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。
- Id、tagを両方を指定した場合はANDとして解釈されます。

4.1.12. スナップショットコピー

4.1.12.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
CopySnapshot

■表示アイコン


4.1.12.2. 部品説明

スナップショットをコピーします。

コピー対象となるスナップショットは、スナップショットIDでの指定の他、タグで検索することができます。

コピー先のリージョンは、エンドポイントが表すリージョンとなります。

4.1.12.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには`{{変数名}}`を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、`{{#セクション名}}{{変数名}}{/セクション名}`と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.12.4. カスタムパラメータ設定

必須	変数	説明	設定例
	DryRun	実際には処理を行わない場合はtrueを指定します。デフォルト値としてfalseが指定されています。空欄またはtrue以外を指定した場合はfalseとして扱います。	true false
○	Description	スナップショットの概要説明を指定します。マルチバイト文字は指定できません。	CopySnapshot
△	SourceSnapshotId	コピー元のスナップショットIDを指定します。	snap-123456ab
△	SourceSnapshottag	コピー元のスナップショットのタグを指定します。 このパラメータを指定した場合、SourceRegion/パラメータの値はエンドポイントが表すリージョンを指定する必要があります。	
○	SourceRegion	コピー元のリージョンを指定します。 SourceSnapshottag/パラメータを指定した場合、このパラメータの値はエンドポイントが表すリージョンを指定する必要があります。	ap-northeast-1

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.1.12.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、CopySnapshotResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.1.12.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.1.12.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeSnapshots
EC2	CopySnapshot

4.1.12.8. 注意事項

特になし

4.1.13. EBSボリュームサイズ拡張

4.1.13.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
ExtendVolumeSize

■表示アイコン



4.1.13.2. 部品説明

EBSボリュームのサイズを拡張します。

対象となるEBSボリュームは、ボリュームIDで指定します。

4.1.13.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.13.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	Volumeld	EBSボリュームIDを指定します。	vol-123456ab
○	Size	EBSボリュームの拡張後のサイズをGiB単位で指定します。	200

4.1.13.5. 部品詳細

本部品は大きく分けて2つのステップの動作があります。

1. サイズ拡張対象となるEBSボリュームの情報取得
2. ボリュームサイズの拡張

本部品の動作の詳細を説明します。

1. Volumeld/パラメータを用いて、DescribeVolumes APIを発行しサイズ拡張対象となるEBSボリュームの情報を取得します。

情報を取得できなかった場合、部品はコード1で終了します。

2. ModifyVolume APIを発行しボリュームに対してサイズの拡張要求を行います。

サイズの拡張要求に失敗した場合、またはSize/パラメータで指定した値が現在のボリュームサイズ以下であった場合、部品はコード1で終了します。

4.1.13.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、オブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.1.13.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.1.13.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeVolumes
EC2	ModifyVolume

4.1.13.9. 注意事項

特になし

4.1.14. EBSボリュームタイプ変更

4.1.14.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticBlockStore

■定義名
ChangeVolumeType

■表示アイコン



4.1.14.2. 部品説明

EBSボリュームのタイプを変更します。

対象となるEBSボリュームは、ボリュームIDで指定します。

4.1.14.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.1.14.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	Volumeld	EBSボリュームIDを指定します。	vol-123456ab
○	VolumeType	EBSボリュームの変更後のタイプを指定します。 ただし、io1 を指定することはできません。	standard, gp2, sc1, st1

4.1.14.5. 部品詳細

本部品は大きく分けて2つのステップの動作があります。

1. タイプ変更対象となるEBSボリュームの情報取得
2. ボリュームタイプの変更

本部品の動作の詳細を説明します。

1. Volumeld/パラメータを用いて、DescribeVolumes APIを発行しタイプ変更対象となるEBSボリュームの情報を取得します。

情報を取得できなかった場合、部品はコード1で終了します。

2. ModifyVolume APIを発行しボリュームに対してタイプの変更要求を行います。

タイプの変更要求に失敗した場合、または現在のボリュームタイプとVolumeTypeパラメータで指定した値が同じであった場合、部品はコード1で終了します。

4.1.14.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、オブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.1.14.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.1.14.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
EC2	DescribeVolumes
EC2	ModifyVolume

4.1.14.9. 注意事項

本部品はボリュームタイプ io1 への変更をサポートしていません。

4.2. S3

Simple Storage Serviceの部品について説明します。

4.2.1. S3部品共通のカスタムパラメータ

S3部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.2.1.1. カスタムパラメータ設定

必須	変数	説明	設定例
	AccessKeyId	APIを利用するためのアクセスキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	APIを利用するためのシークレットキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、s3-ap-northeast-1.amazonaws.comが指定されています。	s3-ap-northeast-1.amazonaws.com
	proxy	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.nichiden.co.jp:8080

4.2.2. S3部品共通仕様

S3部品に共通の仕様について説明します。

4.2.2.1. ワイルドカード設定

S3部品の一部パラメータではワイルドカードを利用することができます。

ワイルドカードとして利用できる文字は以下になります。

利用可能文字	説明
*	0文字以上の任意の文字列と一致するパターン
?	任意の1文字と一致するパターン

4.2.3. ファイルダウンロード

4.2.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/SimpleStorageService

■定義名
GetObject

■表示アイコン



4.2.3.2. 部品説明

S3上のファイルをジョブ実行マシン上にダウンロードします。

4.2.3.3. スクリプト

指定する必要はありません。

4.2.3.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	bucket	S3のバケット名を指定します。	bucket-directory
△	object	ダウンロードするS3のオブジェクト名を指定します。 このパラメータまたはprefixパラメータのどちらか一方が必須です。 wild_cardパラメータがtrueの場合、ワイルドカードを使用できます。その場合、パターンと一致したファイルを全てダウンロードします。	test.txt test?.txt
	file_path	objectパラメータ指定時のオブジェクトの格納先を指定します。 ■ワイルドカードが無効の場合 ダウンロードしたオブジェクトを格納するファイルパスを絶対パスで指定します。指定しない場合は標準出力に内容を表示します。 ■ワイルドカードが有効の場合 ダウンロードしたオブジェクトを格納するディレクトリを絶対パスで指定します。指定しない場合は標準出力に内容を表示します。	C:\test.txt /tmp/test.txt
△	prefix	ダウンロードするS3フォルダ名を絶対パスで指定します。 指定したフォルダ直下のファイルをダウンロードします。サブフォルダのダウンロードには対応していません。 このパラメータまたはprefixパラメータのどちらか一方が必須です。	test/log/

必須	変数	説明	設定例
	dir_path	prefixパラメータ指定時のダウンロード先ディレクトリ名を絶対パスで指定します。指定しない場合は標準出力に内容を表示します。 prefixパラメータが入力されていない場合、このパラメータの値は利用されません。	C:\test\log
	wild_card	objectパラメータでワイルドカードを有効にするかどうかを指定します。有効にする場合はtrueを指定します。無効にする場合は空欄またはfalseを指定します。	true false
	exclude_file	ダウンロード対象から除外するファイルを指定します。 このパラメータはwild_cardパラメータの設定値に関係なくワイルドカードが有効となります。また、複数指定する場合は,(カンマ)で区切って記述します。	tmp?.txt tmp*.txt tmp1.txt,tmp2.txt



[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

prefixパラメータとobjectパラメータの両方に値が入力されている場合、prefixパラメータが優先されます。

4.2.3.5. 実行結果

ダウンロードした結果を出力します。

4.2.3.5.1. 実行例

S3上のbucketに以下のファイルが登録されている場合の例を説明します。

```

+--var/
+----log/
+-----error1.txt
+-----error2.txt
+-----log2017.tar.gz
+-----log20181031.log
+-----log20181101.log
+-----log20181102.log
+-----log20181103.log
+-----log20181104.log
+-----log20181105.log

```

4.2.3.5.1.1. prefixパラメータを用いた複数ファイルのダウンロード例

var/log配下の「error1.txt」、「error2.txt」、「log2017.tar.gz」ファイル以外のすべてのファイルをダウンロードし、/tmp/log配下のディレクトリに格納したい場合、カスタムパラメータを以下の値に設定することで目的のファイルを取得できます。

変数	設定値
bucket	<bucket名>
object	<空白>

変数	設定値
file_path	<空白>
prefix	var/log
dir_path	/tmp/log
wild_card	<空白>
exclude_file	error?.txt,log2017.tar.gz

このカスタムパラメータで実行した結果、ダウンロードされるファイルは以下になります。

- log20181031.log
- log20181101.log
- log20181102.log
- log20181103.log
- log20181104.log
- log20181105.log

4.2.3.5.1.2. objectパラメータを用いた複数ファイルのダウンロード例

var/log配下のファイル名が「log2018」に前方一致するすべてのファイルをダウンロードし、/tmp/log配下のディレクトリに格納したい場合、 カスタムパラメータを以下の値に設定することで目的のファイルを取得できます。

変数	設定値
bucket	< bucket名>
object	var/log/log2018*
file_path	/tmp/log
prefix	<空白>
dir_path	<空白>
wild_card	true
exclude_file	<空白>

このカスタムパラメータで実行した結果、ダウンロードされるファイルは以下になります。

- log20181031.log
- log20181101.log
- log20181102.log
- log20181103.log
- log20181104.log
- log20181105.log

4.2.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。

メッセージ	内容
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
Not a file:XXX.	格納先がファイルではない場合に出力されます。
No such target object.	ダウンロード対象のオブジェクトがひとつも存在しない場合に出力します。

4.2.3.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
S3	ListBucket
S3	GetObject

4.2.3.8. 注意事項

- 取得したオブジェクトの内容を標準出力に表示する場合、JobCenterの言語設定と一致していないと文字化けする可能性があります。

4.2.4. ファイルアップロード

4.2.4.1. 部品情報

■格納パス
/Root/System/AmazonWebService/SimpleStorageService

■定義名
PutObject

■表示アイコン



4.2.4.2. 部品説明

ジョブ実行マシン上のファイルをS3上にアップロードします。

4.2.4.3. スクリプト

指定する必要はありません。

4.2.4.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	bucket	S3のバケット名を指定します。	bucket-directory
○	object	S3のオブジェクト名を指定します。 ■file_pathパラメータにファイルを指定した場合 (ワイルドカードが無効) S3のオブジェクト名にはobjectパラメータがそのまま利用されます。 ■file_pathパラメータにファイルを指定した場合 (ワイルドカードが有効) S3のオブジェクト名にはディレクトリ名にobjectパラメータ、ファイル名にfile_path/パラメータで対象となるファイルが利用されます。 ■file_pathパラメータにディレクトリを指定した場合 S3のオブジェクト名にはディレクトリ名にobjectパラメータ、ファイル名にfile_path/パラメータで対象となるファイルが利用されます。	test.txt var/log
○	file_path	バケットに格納するジョブ実行マシン上のファイルまたはディレクトリを絶対パスで指定します。 ■ファイルを指定した場合(ワイルドカードが無効) アップロードするファイルパスを絶対パスで指定します。 ■ファイルを指定した場合(ワイルドカードが有効) ファイルパスにワイルドカードを含めて絶対パスで指定します。	C:\test.txt /tmp/test.txt

必須	変数	説明	設定例
		条件に一致するファイルが対象となります。 ■ディレクトリを指定した場合 アップロードするディレクトリを絶対パスで指定します。 ディレクトリ直下のファイルが対象となります。サブディレクトリ配下のファイルは対象となりません。	
	wild_card	file_path/パラメータでワイルドカードを有効にするかどうかを指定します。有効にする場合はtrueを指定します。無効にする場合は空欄またはfalseを指定します。	true false
	exclude_file	アップロード対象から除外するファイルを指定します。 このパラメータはwild_card/パラメータの設定値に関係なくワイルドカードが有効となります。また、複数指定する場合は,(カンマ)で区切って記述します。	tmp?.txt tmp*.txt

4.2.4.5. 実行結果

アップロードした結果を出力します。

4.2.4.5.1. 実行例

JobCenterサーバ上に以下のファイルが保存されている場合の例を説明します。

```

+--/tmp/
+----log/
+-----error1.txt
+-----error2.txt
+-----log2017.tar.gz
+-----log20181031.log
+-----log20181101.log
+-----log20181102.log
+-----log20181103.log
+-----log20181104.log
+-----log20181105.log

```

4.2.4.5.1.1. file_path/パラメータにディレクトリを指定した場合の複数ファイルのアップロード例

/tmp/log配下の「error1.txt」、「error2.txt」、「log2017.tar.gz」ファイル以外のすべてのファイルをアップロードし、S3上のbucketのvar/log配下のフォルダに格納したい場合、カスタムパラメータを以下の値に設定することで目的のファイルをアップロードできます。

変数	設定値
bucket	<bucket名>
object	var/log
file_path	/tmp/log

変数	設定値
wild_card	<空白>
exclude_file	error?.txt,log2017.tar.gz

このカスタムパラメータで実行した結果、アップロードされるファイルは以下になります。

■log20181031.log
 ■log20181101.log
 ■log20181102.log
 ■log20181103.log
 ■log20181104.log
 ■log20181105.log

4.2.4.5.1.2. file_path/パラメータにワイルドカードを指定した場合の複数ファイルのアップロード例

/tmp/log配下のファイル名が「log2018」に前方一致するすべてのファイルをアップロードし、S3上のbucketのvar/log配下のフォルダに格納したい場合、カスタムパラメータを以下の値に設定することで目的のファイルをアップロードできます。

変数	設定値
bucket	<bucket名>
object	var/log/
file_path	/tmp/log/log2018*
wild_card	true
exclude_file	<空白>

このカスタムパラメータで実行した結果、アップロードされるファイルは以下になります。

■log20181031.log
 ■log20181101.log
 ■log20181102.log
 ■log20181103.log
 ■log20181104.log
 ■log20181105.log

4.2.4.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。
No such file or directory.	格納するファイルがファイルまたはディレクトリではない、もしくは存在しないパスである場合に出力されます。
No such target object.	アップロード対象のオブジェクトがひとつも存在しない場合に出力します。

4.2.4.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
S3	PutObject

4.2.4.8. 注意事項

特になし

4.2.5. ファイル削除

4.2.5.1. 部品情報

■格納パス
/Root/System/AmazonWebService/SimpleStorageService

■定義名
DeleteObject

■表示アイコン



4.2.5.2. 部品説明

S3上のファイルを削除します。

4.2.5.3. スクリプト

指定する必要はありません。

4.2.5.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	bucket	S3のバケット名を指定します。	bucket-directory
○	object	S3のオブジェクト名を指定します。	test.txt

4.2.5.5. 実行結果

ファイル削除した結果を出力します。

4.2.5.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.2.5.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
S3	DeleteObject

4.2.5.8. 注意事項

■存在しないオブジェクトを削除した場合、ジョブとしては正常終了します。

4.2.6. S3部品共通の注意事項

■UNIX環境では各ジョブはroot権限で実行されるため、S3部品のファイルアクセス権限はrootとなります。

4.3. ELB

Elastic Load Balancingの部品について説明します。

4.3.1. ELB部品共通のカスタムパラメータ

ELB部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.3.1.1. カスタムパラメータ設定

必須	変数	説明	設定例
	AccessKeyID	APIを利用するためのアクセスキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	APIを利用するためのシークレットキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	akdsjfk8klsjdf23oisAKew12
○	endpoint	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、elasticloadbalancing.ap-northeast-1.amazonaws.comが指定されています。	elasticloadbalancing.ap-northeast-1.amazonaws.com
	proxy	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.nichiden.co.jp:8080

4.3.2. クラシックロードバランサーへのインスタンス登録

4.3.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticLoadBalancing

■定義名
RegisterInstancesWithLoadBalancer

■表示アイコン



4.3.2.2. 部品説明

クラシックロードバランサーに指定したインスタンスを登録します。

4.3.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.3.2.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	InstanceId	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りを指定します。	i-123456ab i-123456ab;i-45678cd
○	LoadBalancerName	ロードバランサー名を指定します。	my-http-loadbalancer

4.3.2.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、RegisterInstancesWithLoadBalancerResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.3.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.3.2.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
ELB	RegisterInstancesWithLoadBalancer

4.3.2.8. 注意事項

特になし

4.3.3. クラシックロードバランサーからのインスタンス登録解除

4.3.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/ElasticLoadBalancing

■定義名
DeregisterInstancesFromLoadBalancer

■表示アイコン



4.3.3.2. 部品説明

クラシックロードバランサーから指定したインスタンスを登録解除します。

4.3.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{{変数名}}}/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.3.3.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	InstanceId	EC2インスタンスIDを指定します。複数入力する場合セミコロン；区切りを指定します。	i-123456ab i-123456ab;i-45678cd
○	LoadBalancerName	ロードバランサー名を指定します。	my-http-loadbalancer

4.3.3.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DeregisterInstancesFromLoadBalancerResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.3.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.3.3.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
ELB	DeregisterInstancesFromLoadBalancer

4.3.3.8. 注意事項

特になし

4.4. CF

CloudFormationの部品について説明します。

4.4.1. CF部品共通のカスタムパラメータ

CF部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.4.1.1. カスタムパラメータ設定

必須	変数	説明	設定例
	AccessKeyID	APIを利用するためのアクセスキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	APIを利用するためのシークレットキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、cloudformation.ap-northeast-1.amazonaws.comが指定されています。	cloudformation.ap-northeast-1.amazonaws.com
	proxy	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.nichiden.co.jp:8080

4.4.2. スタック作成

4.4.2.1. 部品情報

■格納パス
/Root/System/AmazonWebService/CloudFormation

■定義名
CreateStack

■表示アイコン



4.4.2.2. 部品説明

指定したテンプレートからスタックを作成します。

4.4.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{変数名}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{変数名}}{{/セクション名}}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.4.2.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	StackName	作成するスタック名を指定します。	
○	TemplateURL	スタック作成に使用するテンプレートファイルがあるURLを指定します。 URLはAmazon S3バケット上にあるファイルを指定する必要があります。	https://s3-ap-northeast-1.amazonaws.com/cf-templates-xxxxxxxxxx-ap-northeast-1/cf_template.json

4.4.2.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、CreateStackResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.4.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.4.2.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudFormation	CreateStack
CloudFormation	DescribeStacks

4.4.2.8. 注意事項

特になし

4.4.3. スタック削除

4.4.3.1. 部品情報

■格納パス
/Root/System/AmazonWebService/CloudFormation

■定義名
DeleteStack

■表示アイコン



4.4.3.2. 部品説明

指定したスタックを削除します。

4.4.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{#{セクション名}} {{{変数名}}} {/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.4.3.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	StackName	削除するスタック名を指定します。	

4.4.3.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DeleteStackResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.4.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.4.3.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudFormation	DeleteStack

4.4.3.8. 注意事項

特になし

4.4.4. スタック情報取得

4.4.4.1. 部品情報

■格納パス
/Root/System/AmazonWebService/CloudFormation

■定義名
DescribeStacks

■表示アイコン



4.4.4.2. 部品説明

指定したスタックの情報取得を行います。

4.4.4.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{#{セクション名}} {{{変数名}}} {/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.4.4.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	StackName	情報取得するスタック名を指定します。	

4.4.4.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DescribeStacksResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。

4.4.4.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.4.4.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudFormation	DescribeStacks

4.4.4.8. 注意事項

特になし

4.5. CW

CloudWatchの部品について説明します。

4.5.1. CW部品共通のカスタムパラメータ

CW部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.5.1.1. カスタムパラメータ設定

必須	変数	説明	設定例
	AccessKeyID	APIを利用するためのアクセスキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	APIを利用するためのシークレットキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、monitoring.ap-northeast-1.amazonaws.comが指定されています。	monitoring.ap-northeast-1.amazonaws.com
	proxy	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.nichiden.co.jp:8080

4.5.2. アラーム情報取得

4.5.2.1. 部品情報

■格納パス

/Root/System/AmazonWebService/CloudWatch

■定義名

DescribeAlarms

■表示アイコン



4.5.2.2. 部品説明

アラームの情報取得を行います。

アラーム名を直接指定またはアラーム名のプレフィックスを指定することで、取得対象となるアラームを絞り込むことができます。

4.5.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには`{{変数名}}`を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、`{#{セクション名}} {{変数名}} {/セクション名}}`と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.5.2.4. カスタムパラメータ設定

必須	変数	説明	設定例
	AlarmName	アラーム名を指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータとAlarmNamePrefixパラメータを同時に指定した場合、AlarmNamePrefixパラメータは無視されます。	alarm001 instanceA_CPUUtil; instanceB_CPUUtil
	AlarmNamePrefix	アラーム名を前方一致で検索します。 このパラメータとAlarmNameパラメータを同時に指定した場合、このパラメータは無視されます。	instanceA_

4.5.2.5. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果が、DescribeAlarmsResponseオブジェクトとして保存されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.5.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。

メッセージ	内容
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.5.2.7. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudWatch	DescribeAlarms

4.5.2.8. 注意事項

- テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.5.3. アラーム状態待ち合わせ

4.5.3.1. 部品情報

■格納パス

/Root/System/AmazonWebService/CloudWatch

■定義名

WaitAlarmState

■表示アイコン



4.5.3.2. 部品説明

アラームの状態がOKとなるまで待ち合わせます。

対象となるアラームは、アラーム名での指定の他、アラーム名のプレフィックスで検索することができます。

4.5.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{#{セクション名}} {{{変数名}}} {/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.5.3.4. カスタムパラメータ設定

必須	変数	説明	設定例
△	AlarmName	アラーム名を指定します。複数入力する場合セミコロン；区切りで指定します。 このパラメータとAlarmNamePrefix/パラメータを同時に指定した場合、AlarmNamePrefix/パラメータは無視されます。 このパラメータまたはAlarmNamePrefixの指定が必要です。	alarm001 instanceA_CPUUtil; instanceB_CPUUtil
△	AlarmNamePrefix	アラーム名を前方一致で検索します。 このパラメータとAlarmName/パラメータを同時に指定した場合、このパラメータは無視されます。 このパラメータまたはAlarmNameの指定が必要です。	instanceA_
	interval	アラームの状態確認を行う間隔を秒単位で指定します。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。	10
	timeout	アラームの状態がOKとなるまで待ち合わせを行う際のタイムアウト時間を分単位で指定します。デフォルト値として30が指定されています。また、空欄の場合は30として扱います。	60

[必須]列の△は、そのパラメータが条件付き必須のパラメータであることを表します。

4.5.3.5. 部品詳細

本部品は大きく分けて2つのステップの動作があります。

1. 状態チェック対象となるアラーム一覧の取得
2. アラームの状態がOKになるまで待ち合わせ

本部品の動作の詳細を説明します。

1. AlarmNameパラメータ、AlarmNamePrefixパラメータのうち、指定されたものを用いて、状態チェック対象となるアラームの一覧を取得します。

アラームの一覧を取得できなかった場合や、アラームがひとつも見つからなかった場合、部品はコード1で終了します。

2. 前ステップで取得したアラーム一覧に対して、状態がOKになるのを待ち合わせます。

interval/パラメータで指定した秒間隔でアラームの状態を取得し、状態がOKかどうかをチェックします。

以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

■アラーム一覧のすべてのアラームの状態が"OK"となったとき

この場合、部品はコード0で終了します。

■待ち合わせ開始時刻から状態が"OK"とならないアラームが存在したままtimeoutパラメータで指定した時間が経過したとき

部品はコード1で終了します。

■アラームの状態取得に失敗した場合

この場合、部品はコード1で終了します。

4.5.3.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

状態チェック対象となるアラームを特定するために、DescribeAlarmsを実行し、DescribeAlarmsResponseに格納されます。詳しくは、部品に記述されているサンプルを参照してください。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.5.3.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.5.3.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
CloudWatch	DescribeAlarms

4.5.3.9. 注意事項

- テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

4.6. RDS

Relational Database Serviceの部品について説明します。

4.6.1. RDS部品共通のカスタムパラメータ

RDS部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

4.6.1.1. カスタムパラメータ設定

必須	変数	説明	設定例
	AccessKeyID	APIを利用するためのアクセスキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	AKIAIOSFODNN7EXAMPLE
	SecretAccessKey	APIを利用するためのシークレットキーを指定します。 IAM Role利用の場合はIAMRoleまたは空白を指定します。	aksdjfka8klsjadf23oisAKew12
○	endpoint	APIリクエストする先のエンドポイントを指定する際に指定します。デフォルト値として、rds.ap-northeast-1.amazonaws.comが指定されています。	rds.ap-northeast-1.amazonaws.com
	proxy	プロキシサーバ経由でリクエストを投げる場合に指定します。プロトコル名://サーバ名:ポート番号の形式で指定します。	http://proxy.nichiden.co.jp:8080

4.6.2. インスタンス起動

4.6.2.1. 部品情報

■格納パス

/Root/System/AmazonWebService/RelationalDatabaseService

■定義名

StartDBInstance

■表示アイコン



4.6.2.2. 部品説明

DBインスタンスを起動します。

対象とするDBインスタンスはDBインスタンスの名前で指定します。

4.6.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには`{{変数名}}`を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、`{{#セクション名}}{{変数名}}``{{/セクション名}}`と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.6.2.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	DBInstanceIdentifier	起動させたいDBインスタンスの名前を指定します。	db001
	interval	DBインスタンスのステータス確認を行う間隔を秒単位で指定します。デフォルト値として30が指定されています。また、空欄の場合は30として扱います。 なお、5より小さい数値を指定することはできません。5より小さい値が指定された場合は5が設定されたものとして扱います。	10
	timeout	DBインスタンスが利用可能になるまでの待ち合わせを行うタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。 なお、1より小さい数値を指定することはできません。1より小さい値が指定された場合は1が設定されたものとして扱います。	30

4.6.2.5. 部品詳細

本部品は大きく分けて3つのステップの動作があります。

1. DBインスタンスのステータスの確認

2. DBインスタンスの起動要求

3. DBインスタンスが利用可能になるまで待ち合わせ

本部品の動作の詳細を説明します。

1. DBInstanceIdentifierで指定されたDBインスタンスについて、DescribeDBInstances APIでステータスを取得します。

DBインスタンスのステータスが取得できなかった場合は、部品はコード1で終了します。

DBインスタンスのステータスが既にavailable(利用可能)である場合、部品はコード0で終了します。

DBインスタンスのステータスが既にstarting(開始中)である場合、DBインスタンスがavailable(利用可能)になるまで待ち合わせます。

DBインスタンスのステータスが他のステータスである場合、DBインスタンスの起動要求を行います。

2. DBInstanceIdentifierで指定されたDBインスタンスについて、StartDBInstance APIで起動の要求を行います。

StartDBInstance APIによる起動の要求が失敗した場合、部品はコード1で終了します。

3. DBInstanceIdentifierで指定されたDBインスタンスについて、ステータスがavailable(利用可能)になるまで待ち合わせます。

interval/パラメータで指定した秒の間隔で、DescribeDBInstances APIでDBインスタンスのステータスを取得し、available(利用可能)かどうかをチェックします。

以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

■DBインスタンスのステータスがavailable(利用可能)となったとき

この場合、部品はコード0で終了します。

■待ち合わせ開始時刻からtimeout/パラメータで指定した時間が経過したとき

この場合、部品はコード1で終了します。

■DBインスタンスのステータスの取得に失敗したとき

この場合、部品はコード1で終了します。

4.6.2.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果として、各ステップで実行したAPIの結果がそれぞれ以下の通りオブジェクトに保存されます。詳しくは、部品に記述されているサンプルを参照してください。

■DBインスタンスのステータスの確認

DescribeDBInstances APIの結果がBeforeDescribeDBInstancesResponseオブジェクトに保存されます。

■DBインスタンスの起動要求

StartDBInstance APIの結果がStartDBInstanceResponseオブジェクトに保存されます。

■DBインスタンスが利用可能になるまで待ち合わせ

待ち合わせ完了時点の最終的なDescribeDBInstances APIの結果がDescribeDBInstancesResponseオブジェクトに保存されます。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには表示されない場合があります。

4.6.2.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] is not a number.	数値を指定する必要があるパラメータに数値以外が指定された場合に出力されます。
Successfully XXX.	APIが成功した場合に出力されます。ただし待ち合わせ中のDescribeDBInstances APIについては出力されず、待ち合わせ完了時点での最後のDescribeDBInstances APIについてのみ出力されます。
Failed to XXX. Message : XXX Code : XXX	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.6.2.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
RDS	DescribeDBInstances
RDS	StartDBInstance

4.6.2.9. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには表示されない場合があります。

■部品実行時点でDBインスタンスのステータスがavailable(利用可能)の場合、その時点で部品は正常終了します。

■部品実行時点でDBインスタンスのステータスがstarting(開始中)の場合、DBインスタンスの起動要求は行わずにステータスがavailable(利用可能)まで待ち合わせを行います。

■待ち合わせの処理では、intervalで指定された間隔でDBインスタンスのステータスのチェックを行っています。

この間隔の間にステータスがavailable(利用可能)になり、さらに他のステータスに遷移した場合、ステータスのチェックの時点でステータスがavailable(利用可能)ではないため、起動部品は実行中のままとなります。

例えばDBインスタンスがavailable(利用可能)となった直後に停止操作が行われた場合、別途操作がない限りステータスがavailable(利用可能)に戻らないため、起動部品はtimeoutで指定した時間で打ち切られるまで実行中となる可能性があります。

■CL/Webを用いてInternet ExplorerからRDS部品を含むジョブネットワークのフローを参照した場合、RDS部品の表示アイコンは黒色で表示されます。

4.6.3. インスタンス停止

4.6.3.1. 部品情報

■格納パス

/Root/System/AmazonWebService/RelationalDatabaseService

■定義名

StopDBInstance

■表示アイコン



4.6.3.2. 部品説明

DBインスタンスを停止します。

対象とするDBインスタンスはDBインスタンスの名前で指定します。

4.6.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{#{セクション名}} {{{変数名}}} {/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

4.6.3.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	DBInstanceIdentifier	停止させたいDBインスタンスの名前を指定します。	db001
	interval	DBインスタンスのステータス確認を行う間隔を秒単位で指定します。デフォルト値として30が指定されています。また、空欄の場合は30として扱います。 なお、5より小さい数値を指定することはできません。5より小さい値が指定された場合は5が設定されたものとして扱います。	10
	timeout	DBインスタンスが停止になるまでの待ち合わせを行うタイムアウト時間を分単位で指定します。デフォルト値として60が指定されています。また、空欄の場合は60として扱います。 なお、1より小さい数値を指定することはできません。1より小さい値が指定された場合は1が設定されたものとして扱います。	30
	createSnapshot	DBインスタンスの停止時にスナップショットを作成する場合はtrueを指定します。trueが指定された場合は「<DBインスタンスID>-<YYYYYYMMDDHHmmSS>」をスナップショット名としてスナップショットを作成します。	true

必須	変数	説明	設定例
		このとき<YYYYMMDDHHmmSS>の時刻はUTC時刻となります。 また、空欄の場合やtrue以外の値が設定されている場合はスナップショットは作成しません。	

4.6.3.5. 部品詳細

本部品は大きく分けて3つのステップの動作があります。

1. DBインスタンスのステータスの確認
2. DBインスタンスの停止要求
3. DBインスタンスが停止になるまで待ち合わせ

本部品の動作の詳細を説明します。

1. DBInstanceIdentifierで指定されたDBインスタンスについて、DescribeDBInstances APIでステータスを取得します。

DBインスタンスのステータスが取得できなかった場合は、部品はコード1で終了します。

DBインスタンスのステータスが既にstopped(停止)である場合、部品はコード0で終了します。

DBインスタンスのステータスが既にstopping(停止中)である場合、DBインスタンスがstopped(停止)になるまで待ち合わせます。

DBインスタンスのステータスが他のステータスである場合、DBインスタンスの停止要求を行います。

2. DBInstanceIdentifierで指定されたDBインスタンスについて、StopDBInstance APIで停止の要求を行います。

StopDBInstance APIによる停止の要求が失敗した場合、部品はコード1で終了します。

3. DBInstanceIdentifierで指定されたDBインスタンスについて、ステータスがstopped(停止)になるまで待ち合わせます。

interval/パラメータで指定した秒の間隔で、DescribeDBInstances APIでDBインスタンスのステータスを取得し、stopped(停止)かどうかをチェックします。

以下のいずれかの条件を満たしたとき、待ち合わせは終了します。

■DBインスタンスのステータスがstopped(停止)となったとき

この場合、部品はコード0で終了します。

■待ち合わせ開始時刻からtimeout/パラメータで指定した時間が経過したとき

この場合、部品はコード1で終了します。

■DBインスタンスのステータスの取得に失敗したとき

この場合、部品はコード1で終了します。

4.6.3.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

実行した結果として、各ステップで実行したAPIの結果がそれぞれ以下の通りオブジェクトに保存されます。詳しくは、部品に記述されているサンプルを参照してください。

■DBインスタンスのステータスの確認

DescribeDBInstances APIの結果がBeforeDescribeDBInstancesResponseオブジェクトに保存されます。

■DBインスタンスの停止要求

StopDBInstance APIの結果がStopDBInstanceResponseオブジェクトに保存されます。

■DBインスタンスが停止になるまで待ち合わせ

待ち合わせ完了時点の最終的なDescribeDBInstances APIの結果がDescribeDBInstancesResponseオブジェクトに保存されます。

テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎると表示されない場合があります。

4.6.3.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] is not a number.	数値を指定する必要があるパラメータに数値以外が指定された場合に出力されます。
Successfully XXX.	APIが成功した場合に出力されます。ただし待ち合わせ中のDescribeDBInstances APIについては出力されず、待ち合わせ完了時点での最後のDescribeDBInstances APIについてのみ出力されます。
Failed to XXX. Message : XXX Code : XXX	APIが失敗した場合に出力されます。Message、CodeにAWSから返されたメッセージが記録されますので参照してください。

4.6.3.8. 使用API

本部品が使用するAWSのAPIの一覧は以下の通りです。

部品の実行前にあらかじめIAMのアクセス許可を設定してください。

サービス	アクション
RDS	DescribeDBInstances
RDS	StopDBInstance

4.6.3.9. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎるときには表示されない場合があります。

■部品実行時点でDBインスタンスのステータスがstopped(停止)の場合、その時点で部品は正常終了します。

■部品実行時点でDBインスタンスのステータスがstopping(停止中)の場合、DBインスタンスの停止要求は行わずにステータスがstopped(停止)まで待ち合わせを行います。

■待ち合わせの処理では、intervalで指定された間隔でDBインスタンスのステータスのチェックを行っています。

この間隔の間にステータスがstopped(停止)になり、さらに他のステータスに遷移した場合、ステータスのチェックの時点でステータスがstopped(停止)ではないため、停止部品は実行中のままとなります。

例えばDBインスタンスがstopped(停止)となった直後に起動操作が行われた場合、別途操作がない限りステータスがstopped(停止)に戻らないため、停止部品はtimeoutで指定した時間で打ち切られるまで実行中となる可能性があります。

- AWS側の仕様としてDBインスタンスを停止した後、7日経過すると自動的に停止したインスタンスが起動されます。これについてはRDSの停止部品を用いてJobCenterからDBインスタンスを停止した場合も同様の動作となります。
- CL/Webを用いてInternet ExplorerからRDS部品を含むジョブネットワークのフローを参照した場合、RDS部品の表示アイコンは黒色で表示されます。

5. Microsoft Azure(V2) 連携部品

本章ではAzure Resource Manager(以下、Azure(V2)と記します)連携部品について説明します。

5.1. 部品共通のカスタムパラメータ

Azure(V2)連携部品に共通のカスタムパラメータについて説明します。

個々の部品の説明では、下記パラメータの説明は省略していますのでご注意ください。

5.1.1. カスタムパラメータ設定

必須	変数	説明	設定例
○	client_id	AzureのAPIの認証に必要です。あらかじめ取得したクライアントIDを指定してください。	
○	username	AzureのAPIの認証に必要です。あらかじめ作成したユーザの名前を指定してください。	jobcenteruser@<ドメイン名>
○	password	AzureのAPIの認証に必要です。あらかじめ作成したユーザのパスワードを指定してください。	
○	endpoint	AzureのAPIのエンドポイントを指定します。デフォルト値としてhttps://management.azure.com/が指定されています。	https://management.azure.com/
○	token_endpoint	AzureのAPIを使用するためのトークンを取得するためのエンドポイントを指定します。あらかじめ取得したOAUTH 2.0トークン エンドポイントを指定してください。	https://login.microsoftonline.com/xxxxxxx/oauth2/token
	proxy	プロキシ経由でAzureのAPIを使用する場合に指定してください。	http://proxy.nichiden.co.jp:8080

5.2. 仮想マシン起動部品

5.2.1. 部品情報

■格納パス
/Root/System/Azure/V2

■定義名
StartVM

■表示アイコン


5.2.2. 部品説明

指定した仮想マシンの起動要求を行います。

対象となる仮想マシンは、名前を直接指定する他、タグで検索することができます。

5.2.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{変数名}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.2.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	subscription_id	AzureのサブスクリプションIDを指定します。	
○	resource_group	起動したい仮想マシンが属するリソースグループの名前を指定します。	ResourceGroup1
△	vm_name	起動したい仮想マシンの名前を指定します。複数指定する場合は、名前をセミコロン(;)で区切ってください。 このパラメータまたはtagパラメータのどちらかが必須です。	vmtest1 vmtest1;vmtest2
△	tag	起動したい仮想マシンをタグで検索します。タグ名=値 の形式で指定してください。 このパラメータまたはvm_nameパラメータのどちらかが必須です。	AutoStart=ON
	wait	仮想マシンの起動要求を行った後、実際に仮想マシンが起動完了するまで部品の終了を待ち合わせる場合はtrueを指定してください。待ち合わせない場合は空欄またはfalseを指定してください。 デフォルト値としてtrueが指定されています。	true false
	interval	waitパラメータにtrueを指定した場合に使用される値です。	10

必須	変数	説明	設定例
		仮想マシンの起動完了チェックを行う間隔を秒数で指定します。未指定時はデフォルト値として5が使用されます。	
	timeout	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの起動完了チェックのタイムアウト時間を秒数で指定します。未指定時はデフォルト値として300が使用されます。	600

5.2.5. 実行結果

■waitパラメータが未指定、またはfalseを指定した場合

起動要求が正常終了した場合は0、エラー時には1が終了値となります。

なお、複数の仮想マシンに対して起動要求を行った場合、すべての起動要求が成功した場合は0、そうでない場合は1が終了値となります。

■waitパラメータにtrueを指定した場合

起動要求および起動チェックが正常終了した場合は0、タイムアウトした場合やエラー時には1が終了値となります。

なお、複数の仮想マシンが対象の場合、すべての起動要求および起動チェックが成功した場合は0、そうでない場合は1が終了値となります。

5.2.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

5.2.7. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■vm_nameパラメータ、tagパラメータの両方を指定した場合はvm_nameが優先されます。

5.3. 仮想マシン停止およびリソース解放部品

5.3.1. 部品情報

■格納パス
/Root/System/Azure/V2

■定義名
StopVM

■表示アイコン


5.3.2. 部品説明

指定した仮想マシンの停止要求を行います。

対象となる仮想マシンは、名前を直接指定する他、タグで検索することができます。

停止に成功した場合、仮想マシンの状態は"Stopped (deallocated)"の状態となります。

5.3.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{{変数名}}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{#{セクション名}} {{{変数名}}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.3.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	subscription_id	AzureのサブスクリプションIDを指定します。	
○	resource_group	停止したい仮想マシンが属するリソースグループの名前を指定します。	ResourceGroup1
△	vm_name	停止したい仮想マシンの名前を指定します。複数指定する場合は、名前をセミコロン(;)で区切ってください。 このパラメータまたはtagパラメータのどちらかが必須です。	vmtest1 vmtest1;vmtest2
△	tag	停止したい仮想マシンをタグで検索します。タグ名=値 の形式で指定してください。 このパラメータまたはvm_nameパラメータのどちらかが必須です。	AutoStart=ON
	wait	仮想マシンの停止要求を行った後、実際に仮想マシンが停止完了するまで部品の終了を待ち合わせの場合はtrueを指定してください。待ち合わせない場合は空欄またはfalseを指定してください。 デフォルト値としてtrueが指定されています。	true false

必須	変数	説明	設定例
	interval	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの停止完了チェックを行う間隔を秒数で指定します。未指定時はデフォルト値として5が使用されます。	10
	timeout	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの停止完了チェックのタイムアウト時間を秒数で指定します。未指定時はデフォルト値として300が使用されます。	600

5.3.5. 実行結果

■waitパラメータが未指定、またはfalseを指定した場合

停止要求が正常終了した場合は0、エラー時には1が終了値となります。

なお、複数の仮想マシンに対して停止要求を行った場合、すべての停止要求が成功した場合は0、そうでない場合は1が終了値となります。

■waitパラメータにtrueを指定した場合

停止要求および停止チェックが正常終了した場合は0、タイムアウトした場合やエラー時には1が終了値となります。

なお、複数の仮想マシンが対象の場合、すべての停止要求および停止チェックが成功した場合は0、そうでない場合は1が終了値となります。

5.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

5.3.7. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■vm_nameパラメータ、tagパラメータの両方を指定した場合はvm_nameが優先されます。

5.4. 仮想マシン停止部品

5.4.1. 部品情報

- 格納パス
/Root/System/Azure/V2
- 定義名
PowerOffVM
- 表示アイコン


5.4.2. 部品説明

指定した仮想マシンの停止要求を行います。

対象となる仮想マシンは、名前を直接指定する他、タグで検索することができます。

停止に成功した場合、仮想マシンの状態は"Stopped"の状態となります。



"Stopped"の状態ではネットワークリソースなどが解放されず、課金状態が維持されます。

5.4.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには`{{変数名}}`を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、`{#{セクション名}} {{変数名}} {{/セクション名}}`と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.4.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	subscription_id	AzureのサブスクリプションIDを指定します。	
○	resource_group	停止したい仮想マシンが属するリソースグループの名前を指定します。	ResourceGroup1
△	vm_name	停止したい仮想マシンの名前を指定します。複数指定する場合は、名前をセミコロン(;)で区切ってください。 このパラメータまたはtagパラメータのどちらかが必須です。	vmtest1 vmtest1;vmtest2
△	tag	停止したい仮想マシンをタグで検索します。タグ名=値 の形式で指定してください。 このパラメータまたはvm_nameパラメータのどちらかが必須です。	AutoStart=ON
	wait	仮想マシンの停止要求を行った後、実際に仮想マシンが停止完了するまで部品の終了を待ち合わせ	true

必須	変数	説明	設定例
		る場合はtrueを指定してください。待ち合わせない場合は空欄またはfalseを指定してください。 デフォルト値としてtrueが指定されています。	false
	interval	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの停止完了チェックを行う間隔を秒数で指定します。未指定時はデフォルト値として5が使用されます。	10
	timeout	waitパラメータにtrueを指定した場合に使用される値です。 仮想マシンの停止完了チェックのタイムアウト時間を秒数で指定します。未指定時はデフォルト値として300が使用されます。	600

5.4.5. 実行結果

■waitパラメータが未指定、またはfalseを指定した場合

停止要求が正常終了した場合は0、エラー時には1が終了値となります。

なお、複数の仮想マシンに対して停止要求を行った場合、すべての停止要求が成功した場合は0、そうでない場合は1が終了値となります。

■waitパラメータにtrueを指定した場合

停止要求および停止チェックが正常終了した場合は0、タイムアウトした場合やエラー時には1が終了値となります。

なお、複数の仮想マシンが対象の場合、すべての停止要求および停止チェックが成功した場合は0、そうでない場合は1が終了値となります。

5.4.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。
Invalid tag format: XXX	tagの指定方法が間違っている場合に表示されます。

5.4.7. 注意事項

■テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

■vm_nameパラメータ、tagパラメータの両方を指定した場合はvm_nameが優先されます。

5.5. 仮想マシン情報(モデルビュー)取得

5.5.1. 部品情報

- 格納パス
/Root/System/Azure/V2
- 定義名
ModelView
- 表示アイコン


5.5.2. 部品説明

指定した仮想マシンの情報取得を行います。

対象となる仮想マシンは名前で指定します。

情報取得に成功した場合、ジョブの出力結果に情報を出力します。

取得できる情報としては、タグ、仮想マシンのサイズなどがあります。

その他、取得できる情報については下記サイトを参照してください。

<https://msdn.microsoft.com/ja-jp/library/azure/mt163682.aspx>

5.5.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{変数名}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{変数名}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.5.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	subscription_id	AzureのサブスクリプションIDを指定します。	
○	resource_group	情報取得したい仮想マシンが属するリソースグループの名前を指定します。	ResourceGroup1
○	vm_name	情報取得したい仮想マシンの名前を指定します。	vmtest1 vmtest1;vmtest2

5.5.5. 実行結果

情報取得が正常終了した場合は0、エラー時には1が終了値となります。

以下に、取得した情報の出力例を示します。

```
Successfully model_view
```

```

----- ModelViewResult -----
type: Microsoft.Compute/virtualMachines
name: WS2012R2-test
id: /subscriptions/XXXXXXXXXX/resourceGroups/rg-test1/providers/Microsoft.Compute/virtualMachines/
WS2012R2-test
location: japaneast
----- tags -----
AutoShutdown=Yes
AutoStart=Yes

----- properties -----
osProfile:
  computerName: WS2012R2-test
  adminUsername: JobCenter
networkProfile:
  id: /subscriptions/XXXXXXXXXX/resourceGroups/rg-test1/providers/Microsoft.Compute/
virtualMachines/WS2012R2-test
hardwareProfile:
  vmSize: Standard_DS1
storageProfile:
  dataDisks:
  imageReference:
    version: latest
    publisher: MicrosoftWindowsServer
    sku: 2012-R2-Datacenter
    offer: WindowsServer
  osDisk:
    name: WS2012R2-test
    vhd:
      uri: https://XXXXXXXXXX.blob.core.windows.net/vhds/WS2012R2-test.vhd
    caching: ReadWrite
    osType: Windows
    createOption: FromImage

provisioningState: Succeeded

```

5.5.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。

5.5.7. 注意事項

- テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

5.6. 仮想マシン情報(インスタンスビュー)取得

5.6.1. 部品情報

■格納パス
/Root/System/Azure/V2

■定義名
InstanceView

■表示アイコン


5.6.2. 部品説明

指定した仮想マシンの情報取得を行います。

対象となる仮想マシンは名前で指定します。

情報取得に成功した場合、ジョブの出力結果に情報を出力します。

取得できる情報としては、仮想マシンの電源状態、ゲスト エージェントの状態などがあります。

その他、取得できる情報については下記サイトを参照してください。

<https://msdn.microsoft.com/ja-jp/library/azure/mt163682.aspx>

5.6.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには{{変数名}}を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、{{#セクション名}}{{変数名}}{/セクション名}と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

5.6.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	subscription_id	AzureのサブスクリプションIDを指定します。	
○	resource_group	情報取得したい仮想マシンが属するリソースグループの名前を指定します。	ResourceGroup1
○	vm_name	情報取得したい仮想マシンの名前を指定します。	vmtest1 vmtest1;vmtest2

5.6.5. 実行結果

情報取得が正常終了した場合は0、エラー時には1が終了値となります。

以下に、取得した情報の出力例を示します。

```
Successfully instance_view
```

```
----- vmAgent -----
message: VM Agent is unresponsive.
time: 2016-01-19T08:32:50+00:00
code: ProvisioningState/Unavailable
level: Warning
displayStatus: Not Ready
vmAgentVersion: Unknown

----- disks -----
name: WS2012R2-test

level: Info
time: 2016-01-18T07:19:59.2814519+00:00
displayStatus: Provisioning succeeded
code: ProvisioningState/succeeded
-----

----- statuses -----
level: Info
time: 2016-01-18T07:19:59.343958+00:00
displayStatus: Provisioning succeeded
code: ProvisioningState/succeeded
-----

level: Info
displayStatus: VM deallocated
code: PowerState/deallocated
-----
```

5.6.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
Failed to XXX.	APIが失敗した場合に出力されます。Message、CodeにAzureから返されたメッセージが記録されますので参照してください。

5.6.7. 注意事項

- テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。データが多すぎる場合には表示されない場合があります。

6. UiPath 連携部品

UiPath 連携部品について説明します。

6.1. ジョブ開始

6.1.1. 部品情報

- 格納パス
/Root/System/UiPath/Jobs
- 定義名
StartJobs
- 表示アイコン


6.1.2. 部品説明

UiPath Orchestratorで定義済みのプロセスからジョブを起動し、その実行を完了まで待ち合わせます。

なお、本部品でジョブを起動できるのはUnattendedロボットのみです。

6.1.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

値を参照するには`{{変数名}}`を記述します。

また複数の値を参照する場合にはセクションと呼ばれる記述を行う必要があり、`{#{セクション名}}
{{変数名}}{/セクション名}}`と記述します。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

6.1.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	endpoint	UiPath OrchestratorのAPIエンドポイントのURLを指定します。	http://myuipath.local
	proxy	プロキシサーバ経由でAPIリクエストを送信する場合に指定します。 プロトコル名://サーバ名:ポート番号 の形式で指定します。	http://proxy:8080
	tenancyName	テナント名を指定します。	tenant1
	organizationUnitName	組織単位名を指定します。 組織単位を有効にしている環境の場合は必須です。	OU1
○	usernameOrEmailAddress	ユーザ名、またはメールアドレスを指定します。	user1
○	password	パスワードを指定します。	password123 \$user1_password\$ (暗号化パラメータ指定)
	expirationTime	APIアクセストークンの有効期限を分単位で指定します。 省略時は30分となります。	60
○	ProcessName	プロセスの名前を指定します。	process1
	EnvironmentName	ロボットグループの名前を指定します。	TESTENV

必須	変数	説明	設定例
	Strategy	ジョブの実行方法を指定します。 "All", "Specific", "JobsCount" のいずれかを指定してください。 省略時は "All" となります。	Specific
	RobotNames	ジョブを実行するロボット名を指定します。 Strategyパラメータで "Specific" を指定した場合は必須です。 ロボット名を複数指定する場合は、RobotNamesSeparatorパラメータで指定した文字で区切ってください。	robot1 robot2;robot3 (RobotNamesSeparator パラメータに ; を指定) robot4 robot5 robot6 (RobotNamesSeparator パラメータに %s を指 定)
	RobotNamesSeparator	RobotNamesパラメータで複数のロボット名を指定する際の区切り文字を指定します。 Lua言語のパターン表記で指定します。	; %s (空白文字を区切り 文字に指定)
	JobsCount	実行するジョブ数を指定します。 Strategyパラメータで "JobsCount" を指定した場合は必須です。	3
	Source	ジョブのソースを指定します。 "Manual", "Schedule" のいずれかを指定してください。 省略時は "Manual" となります。	Manual
	InputArgumentsFile	入力パラメータを記載したJSONファイルを絶対パスで指定します。 なお、入力パラメータファイルの文字コードはJobCenterのセットアップ言語としてください。	C:\input\param1.json
	OutputJobIdsFile	ジョブIDを出力するファイルを絶対パスで指定します。 省略時はファイル出力を行いません。 既にファイルが存在した場合はファイルを上書きします。 なお、出力するジョブIDが複数ある場合はセミコロン(;)区切りで出力します。	C:\output\jobid1.txt
	interval	ジョブの状態確認を行う間隔を秒単位で指定します。 省略時は30秒となります。	60
	timeout	ジョブの実行完了の待ち合わせを行う際のタイムアウト時間を分単位で指定します。 省略時は1440分となります。	120

6.1.5. 部品詳細

本部品の動作の流れを説明します。

1. パラメータの事前チェック

各パラメータのチェックを行います。不正な指定があった場合、部品はここでコード1で終了します。

2. 組織単位IDの取得

organizationUnitNameパラメータが指定されていない場合、この処理はスキップします。

organizationUnitNameパラメータが指定されている場合、ユーザ情報取得APIリクエストを発行してユーザ情報を取得してorganizationUnitNameパラメータと一致する名前の組織単位のIDを取得します。

APIリクエストに失敗した場合や、APIレスポンスにorganizationUnitNameパラメータと一致する名前の組織単位情報が含まれていなかった場合は部品はここでコード1で終了します。

3. リリースキーの取得

ジョブを起動するためのリリースキーを取得します。

ProcessNameパラメータとEnvironmentNameパラメータを使用してリリース情報取得APIリクエストを発行し、そのレスポンスからリリースキーを取得します。

APIリクエストに失敗した場合や、APIレスポンスにリリース情報が含まれていなかった場合は部品はここでコード1で終了します。

4. ロボットIDの取得

Strategyパラメータが"Specific"ではない場合、この処理はスキップします。

Strategyパラメータが"Specific"の場合、ロボット情報取得APIリクエストを発行してRobotNamesパラメータで指定したロボット名に対応するロボットIDを取得します。

APIリクエストに失敗した場合や、指定した名前のロボットが見つからなかった場合は部品はここでコード1で終了します。

5. ジョブの起動

「リリースキーの取得」ステップで取得したリリースキーを使用して、ジョブ起動APIリクエストを発行してジョブを起動します。

APIリクエストに失敗した場合、部品はここでコード1で終了します。

6. ジョブIDの出力

OutputJobIdsFileパラメータが指定されていない場合、この処理はスキップします。

OutputJobIdsFileパラメータで指定したファイルに「ジョブの起動」ステップで起動したジョブIDを出力します。

ジョブIDの出力が失敗した場合、部品はここでコード1で終了します。

7. ジョブのステータス監視

intervalパラメータで指定した秒数間隔でジョブの情報を繰り返し取得し、ジョブのステータスを監視します。

ジョブのステータスが以下のいずれかとなる、timeoutパラメータで指定した時間を超える、またはAPIリクエストに失敗した場合にステータス監視処理を終了します。

■ Successful

■ Stopped

■ Faulted

8. 結果出力

部品の「スクリプト」に記述されたテンプレートを使用して結果を出力します。

部品の終了コードは「ジョブのステータス監視」ステップの終了理由によって以下になります。

終了コード	説明
0	■ 部品から実行されたジョブがすべてSuccessfulで終了した場合
1	■ 部品から実行されたジョブが一つでもStopped, Faultedで終了した場合 ■ 部品から実行されたジョブがステータス監視処理でtimeoutした場合

6.1.6. 実行結果

各APIを実行した結果をテンプレートエンジンで変換して標準出力に出力します。

以下に、出力例を示します。

```

===== summary =====
JobIds=76
JobCount=1
SuccessJobIds=76
SuccessJobCount=1
FaultedJobIds=
FaultedJobCount=0
StoppedJobIds=
StoppedJobCount=0
TimeoutJobIds=
TimeoutJobCount=0

===== error messages =====

===== detail =====
----- start jobs result -----
***** Id: 76 *****
State: Pending
CreationTime: 2019-11-01T08:10:03.421368Z
Key: c1d629eb-9db3-4ef7-9a61-c10bccb8eedd
ReleaseName: process1_TESTENV
BatchExecutionKey: af9fd7e0-70d3-410c-b1c8-afb93caf564e

----- get jobs result -----
***** Id: 76 *****
State: Successful
CreationTime: 2019-11-01T08:10:03.42Z
StartTime: 2019-11-01T08:10:03.67Z
EndTime: 2019-11-01T08:12:11.39Z
Key: c1d629eb-9db3-4ef7-9a61-c10bccb8eedd
ReleaseName: process1_TESTENV
BatchExecutionKey: af9fd7e0-70d3-410c-b1c8-afb93caf564e
RobotName: robot1
Info: ジョブは完了しました
OutputArguments: {"Division":"abc","total":1000}

```

6.1.7. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
parameter [XXX] should be a number.	数値を指定する必要があるパラメータに数値以外が指定された場合に出力されます。
parameter [XXX] is invalid.	パラメータに指定した値が不正である場合に出力されます。

メッセージ	内容
Failed to read file. (InputArgumentsFile=<InputArgumentsFile/パラメータの値 >)	入力パラメータファイルの読み込みに失敗した場合に出力されます。
File format is invalid. (InputArgumentsFile=<InputArgumentsFile/パラメータの値 >)	入力パラメータファイルのフォーマットが不正である場合に出力されます。
script is invalid.	部品の「スクリプト」に記述されたテンプレートのフォーマットが不正である場合に出力されます。
Failed to get organization unit information.	ユーザ情報取得APIリクエストに失敗した場合に出力されます。
Organization unit < organizationUnitName/パラメータの値 > not found.	ユーザ情報取得APIのレスポンスにorganizationUnitName/パラメータと一致する名前の組織単位情報が含まれていなかった場合に出力されます。
Failed to get releases.	リリース情報取得APIリクエストに失敗した場合に出力されます。
Any releases not found.	リリース情報取得APIのレスポンスにリリース情報が含まれていなかった場合に出力されます。
Failed to get robot ids.	ロボット情報取得APIリクエストに失敗した場合に出力されます。
Robot not found. (RobotNames=<RobotNamesで指定したロボット名 >)	指定したロボット名のロボット情報が見つからなかった場合に出力されます。
Failed to start jobs.	ジョブ起動APIリクエストに失敗した場合に出力されます。
Failed to write file. (OutputJobIdsFile=<OutputJobIdsFile/パラメータの値 >)	ジョブIDのファイルの書き込みに失敗した場合に出力されます。
Failed to get jobs.	ジョブ情報取得APIリクエストに失敗した場合に出力されます。
Job state check time out.	timeout/パラメータで指定した時間を超えたためジョブのステータス監視処理をタイムアウトした場合に出力されます。

6.1.8. 使用API

本部品が使用するUiPath OrchestratorのAPIの一覧は以下の通りです。

部品の実行前にあらかじめアクセス許可を設定してください。

API	使用用途
POST /api/Account/Authenticate	認証 アクセストークンの取得のため
GET /odata/Users/ UiPath.Server.Configuration.OData.GetCurrentUser	ユーザ情報取得 組織単位情報の取得のため
GET /odata/Releases	リリース情報取得 リリースキーの取得のため
GET /odata/Robots	ロボット情報取得 ロボットIDの取得のため
POST /odata/Jobs/ UiPath.Server.Configuration.OData.StartJobs	ジョブ起動 ジョブの起動のため

API	使用用途
GET /odata/Jobs	ジョブ情報取得 ジョブのステータス監視のため

6.1.9. 注意事項

- Sourceパラメータに "Schedule" を指定しても、ジョブは "Manual" で実行されます。
- テンプレートエンジンの出力対象となったデータは、json形式で標準エラー出力に出力します。ただしデータが多すぎる時には表示されない場合があります。

7. Box連携部品

本章ではBox連携部品について説明します。

7.1. Box連携部品概要

Box連携部品はBox社が提供しているBox APIを実行することができる機能を提供しています。

Box APIはBOX向けAPIトークンがあれば利用できる機能であり、本Box連携部品はJWT認証によるAPIトークンの発行やアップロード、ダウンロードなどのAPI実行機能を提供しています。



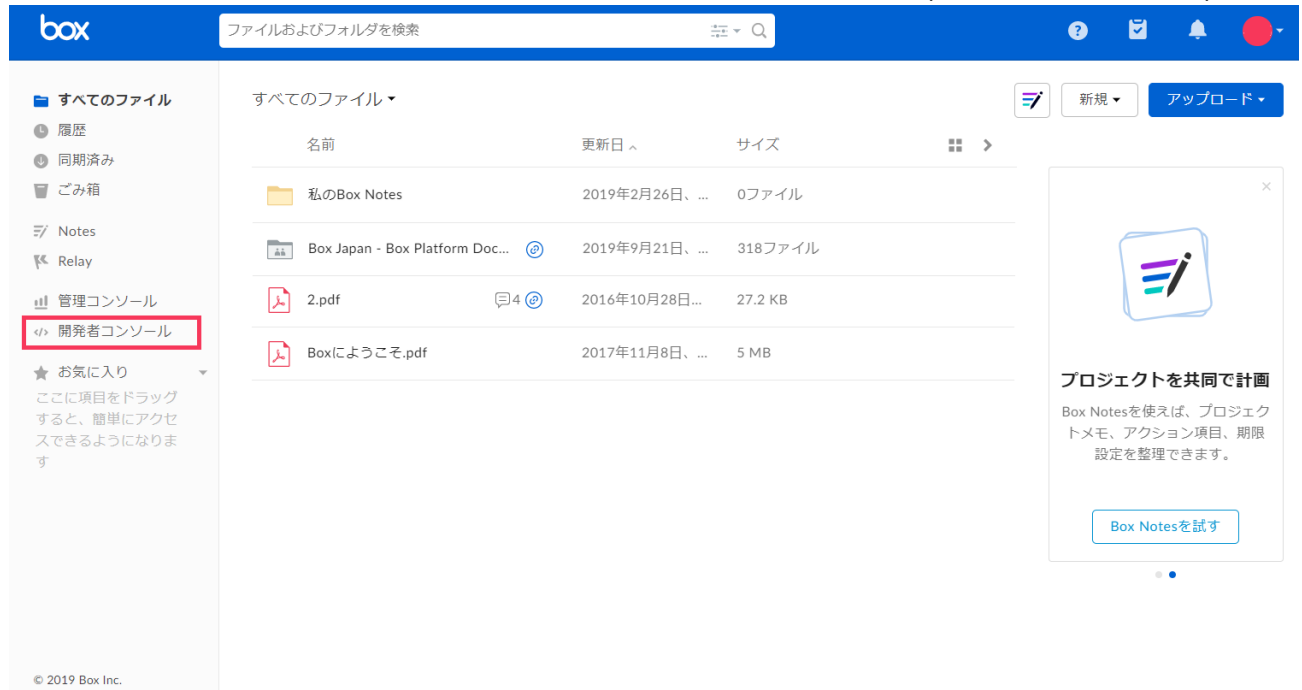
本部品はWindowsでご利用される場合、非UNICODEモードでセットアップを行ってご利用ください。UNICODEモードでセットアップされた場合、正しく動作が出来ない、文字化けするなどの問題が発生します。

7.1.1. 事前準備

JobCenterのBox連携部品を利用するためにはBoxのアプリ作成など事前準備が必要です。以下に必要となる事前準備について記載します。

■Boxアプリケーションの作成

Boxの管理者アカウントでログインを行い、開発者コンソールを開きます。（赤枠を押下で開きます）



開発者コンソールのマイアプリでアプリの新規作成を選び、カスタムアプリを新規作成します。（どちらの赤枠を押下で新規作成画面に遷移します）



認証方式として「JWTを使用したOAuth 2.0(サーバー認証)」を選択してください。(赤枠が選択されていることを確認してください)

新しいBOXアプリを作成する

認証方法

選択したアプリケーションの種類に基づいて認証方法を提案しました。以下から認証方式を変更できます。[詳細を表示](#)

カスタムアプリにおすすめ:

JWTを使用したOAuth 2.0 (サーバー認証)

ユーザーの資格情報の代わりに、デジタル署名されたJSONウェブトークンを使用してアプリから直接Boxを認証できます。サービスアカウントおよびApp Userで使用できます。

利用できるその他の認証方式:

アプリトークン(サーバー認証)

お客様のアプリケーションでのコンテンツのプレビューをスコープとしたAPI機能を提供します。これにより、Box内のユーザーを作成したり管理したりすることはできません。続行する前に、新しいBox ViewでBoxのドキュメントをお読みになり、このAPI機能がお客様のアプリケーション要件に一致することを確認してください。

標準OAuth 2.0 (ユーザー認証)

アプリがユーザーのコンテンツにアクセスすることを認証するには、ユーザー名とパ

■公開キーの追加について

アプリケーションの追加後、「構成」から公開キーの追加が出来ます。JobCenterからJWT認証を行う場合にはRSAキーペアを生成して公開鍵をBoxに登録する必要があります。

「公開キーを追加」のボタンを押下してください。公開キーを入力するウィンドウが表示されます。

🔧 マイアプリ

JC_EXECUTE

⚙️ 一般

🔗 構成

🔗 Webhook

リファレンス

🔗 SDK

📄 APIドキュメント

🔗 サポート

構成

アプリでBox APIの使用を開始するための認証と権限を構成します。これらの設定の手順については、[入門ガイド](#)をご覧ください。

公開キーの追加と管理

アプリのJWT要求に署名して認証するためのRSAキーペアを生成するか、独自の公開キーをアップロードします。[独自のRSAキーペアを生成する方法を参照します。](#)

注: Boxに秘密キーは保存されないの
で、生成ボタンを使用している場合
は、ダウンロードしたファイルを実
際に保存してください。

公開キーを追加

公開/秘密キーペアを生成

表示された「新しい公開キーを追加」のウィンドウに作成したRSAキーペアの公開鍵を登録します。

新しい公開キーを追加

公開キー

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwjZ0sbsh0JRwHfJr4f1t
Zbus3bS46meAUvvcev4Gfi1K4QzHH3+yz/f6ykRfrJtcCOPRA3Brd6FDgPGrYMLY
pIX85aRdc4vndJ7g7jSd8JvuZSghs2h2s2lFLHCU0CETjnGxZwW6XVKn+QV1nu2H
TDP6sxx5U0DGCQA+ln7Bj6g8Hnx8Yjg0RUMbeb/QYC4jIfrJvR8kUkMRqF9+Hj6h
TY/A2K3TsXW0ISwIXBckukjq0Vi7DwH2n34NLDIsEr8u5K1t4tf1jkZskvIB48MU
Mq8sa7lnBfD01lDarX3XF6AJXRGngUahwN4q6WZTEfJSeoy4bjhfbEmf4Ci104j
9wIDAQAB
-----END PUBLIC KEY-----
```

キャンセル

確認して保存

登録すると公開キーIDが発行されます。この公開キーIDをJobCenterのBox連携部品のパラメータで利用します。

マイアプリ

JC_EXECUTE

一般

構成

Webhook

リファレンス

SDK

APIドキュメント

サポート

構成

アプリでBox APIの使用を開始するための認証と権限を構成します。これらの設定の手順については、[入門ガイド](#)をご覧ください。

公開キーの追加と管理

公開キー-1 ID: s3jng6og

削除

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwjZ0sbsh0JRwHfJr4f1t
Zbus3bS46meAUvvcev4Gfi1K4QzHH3+yz/f6ykRfrJtcCOPRA3Brd6FDgPGrYMLY
pIX85aRdc4vndJ7g7jSd8JvuZSghs2h2s2lFLHCU0CETjnGxZwW6XVKn+QV1nu2H
TDP6sxx5U0DGCQA+ln7Bj6g8Hnx8Yjg0RUMbeb/QYC4jIfrJvR8kUkMRqF9+Hj6h
TY/A2K3TsXW0ISwIXBckukjq0Vi7DwH2n34NLDIsEr8u5K1t4tf1jkZskvIB48MU
Mq8sa7lnBfD01lDarX3XF6AJXRGngUahwN4q6WZTEfJSeoy4bjhfbEmf4Ci104j
9wIDAQAB
-----END PUBLIC KEY-----
```

公開キーを追加

公開/秘密キーペアを生成

アプリのJWT要求に署名して認証するためのRSAキーペアを生成するか、独自の公開キーをアップロードします。**独自のRSAキーペアを生成する方法を参照**します。

注: Boxに秘密キーは保存されないの

で、生成ボタンを使用している場合

は、ダウンロードしたファイルを確認

に保存してください。

(ご参考) RSAキーペアを作るコマンドの一例

RSAキーペアはOpensslコマンド等で生成可能です。作成した実際のコンソール画面の一例を記載します。

119

```
[root@localhost ~]# openssl genrsa -out private.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
[root@localhost ~]# openssl rsa -in private.pem -pubout -out public.out
writing RSA key
[root@localhost ~]# cat public.out
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwVzkYwaYwSvX/JydbwLr
FSWIS+I6uTPxKWeuIXS9N/2KKA85RtQNL0ZIOzCVv/e5EiEq0vp3sWIo1LI2elxJ
HR6/zxEjGQgmd5GpHJHo16dLxxIbXHzpLrCap+IJ+ivstIychKqVwDpTfpg1jr1S
ExvKhZqN+WaH3AeUB2awSnQ+t80bsce8zbhRWEVN5KxJB+wwsqNg9zzBW5bBpvH
atNvTpiu5JSTUakUpWBr38XG7D3cLlLoLQgSrQhspvfuWU6zAGmz9bB6h3zpRuJL
x4qGQCx9B9EU2QV+gha imor zeD0+2UaJ1v9dGq8XH9bQ3oA65YW01kmKtp5KEwFh
JQIDAQAB
-----END PUBLIC KEY-----
[root@localhost ~]#
```

opensslコマンドで作成したpublic.out（公開鍵）の出力内容をBox側に登録してください。また、JWT認証の際にはprivate.pem（秘密鍵）が必要となりますのでご注意ください。



Box側でRSA鍵を作成することも可能ですが、その際にダウンロードする秘密鍵にはパスフレーズが設定されています。Box連携部品はパスフレーズの秘密鍵を利用することはできませんので、別途OpenSSLコマンドで「パスフレーズを解除」を行ってからご利用ください。

■作成したBoxアプリの設定

作成したアプリの「構成」から必要な設定を行います。以下の表に従い、必要な設定を行ってください。

項目	設定内容	設定詳細
アプリケーションアクセス	Enterpriseを指定	既存ユーザーへのアクセスが必要となるため、Enterpriseを指定してください。
アプリケーションスコープ	利用したい機能にチェックボックス有効化	利用するAPIの対象となるスコープを選択してください。（アップロードには書き込み権限が必要となります）
高度な機能	両方とも有効化	ユーザーとしてAPIを実行できるよう、両方とも有効化を行ってください。

■アプリの承認

管理コンソールから作成したカスタムアプリケーションの承認を行います。管理コンソールの「Enterprise設定」を開きます。

「Enterprise設定」の上のバーメニューから「アプリ」を押下し、表示された項目の「カスタムアプリケーション」－「新しいアプリケーションを承認」を押下します。（画面の赤枠を①、②、③の順番で押下してください）

管理コンソール

- インサイト
- ユーザーとグループ
- コンテンツ
- レポート
- Relay
- ガバナンス
- プラットフォーム
- アカウントと請求
- Enterprise設定** ①
- マイアカウントに戻る

ユーザーの検索

カスタム設定 セキュリティ コンテンツと共有 通知 ユーザー設定 **アプリ** モバイル アプリ使用の管

アプリ

アプリトークン: ☐ アプリトークンを使用するには管理者による承認が必要 ①

保存

カスタムアプリケーション 1/25個のApp Userライセンスを使用中

名前	App User	アクセス	処理
	0	アプリを無効にしました	...
	0	アプリを無効にしました	...
	1	アプリを無効にしました	...
	0	すべてのユーザー	...

③ ③

新しいアプリケーションを承認

アプリケーションの個別管理

© 2019 Box Inc. 1/147

ヒント: 会社固有のアクセス権でカスタムアプリケーションへのアクセスを許可できます。アプリ認証とApp Userを使用したカスタムアプリケーションの作成の詳細については、[開発者向けWebサイト](#)を参照してください。

注: サードパーティのサービスに接続する場合、そのサードパーティサービスの使用に関しては、その会社の該当する利用規約およびプライバシーポリシーが適用されます。

ヒント: ユーザーが使えるアプリケーションの追加/削除ができます。アプリケーションを公開するときは、アプリケーションの特定のコンポーネントをユーザーに公開するかどうかも設定できます。

アプリの承認というダイアログが出てきたら、作成したアプリのAPIキーが求められます。このAPIキーは作成したBoxアプリのクライアントIDなので、アプリ詳細画面からクライアントIDの文字列をコピーしてアプリの認証を行ってください。

7.2. 部品共通のカスタムパラメータ

Box連携部品で共通となるカスタムパラメータの説明をします。

以下に記載するパラメータは全部品で共通的に設定可能です。

「[7.3 BOX_APIトークン取得部品](#)」ではこれらのパラメータの設定が必須となります。

それ以外の部品の場合、「box_apitoken」が未指定の場合に設定が必須となります。

7.2.1. カスタムパラメータ設定

必須	変数	説明	設定例
○	privkey_filepath	Boxに登録したRSA公開鍵と対になるRSA秘密鍵のファイルパスを指定してください	C:\tmp\boxpem.key
○	head_kid	BoxアプリでRSA公開鍵を登録後に表示される公開キーIDを指定してください	abcde……(文字列を設定)
○	payload_iss	サービスのクライアントIDを指定してください	abcde……(文字列を設定)
○	payload_sub	対象となるエンタープライズIDかユーザーIDを指定してください	12345678……(数値のIDを設定)
○	payload_box_sub_type	Boxのユーザー種別(enterprise or user)を指定してください	enterprise
○	client_id	BoxアプリのクライアントIDを指定してください	abcde……(文字列を設定)
○	client_secret	Boxアプリのクライアントシークレットを指定してください	abcde……(文字列を設定)

7.3. BOX_APIトークン取得部品

7.3.1. 部品情報

■格納パス
/Root/System/Tools/Box

■定義名
box_gettoken

■表示アイコン



7.3.2. 部品説明

Box APIを実行するためのAPIトークンを取得します。

APIトークンの取得に成功すると拡張カスタムパラメータ"dbkey"で指定された暗号化DBのキーにAPIトークンを格納します。

後続のジョブは暗号化DBからAPIトークンを参照して利用することが出来ます。例えば、後続のジョブでHTTP接続部品の「Delete」部品を配置して「ファイルの削除」を行うなど様々なBOXのAPIを利用することが可能になります。

7.3.3. スクリプト

本部品にはスクリプトはありません。

7.3.4. カスタムパラメータ設定

必須	変数	説明	設定例
○	dbkey	APIトークンを暗号化して格納する暗号化DBのキーを指定してください	testkey

7.3.5. 実行結果

■APIトークンの取得に成功した場合

APIトークンの取得に成功すると終了値0となり、暗号化DBの"dbkey"で指定したキーにAPIトークンを暗号化して格納します。

■必須パラメータが不足した場合

必須パラメータが不足すると終了値1となり、エラーメッセージが標準エラーに出力されます。

■APIトークンの取得に失敗した場合

APIトークンの取得に失敗すると終了値2となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■暗号化DBの書き込みに失敗した場合

暗号化DBへの更新に失敗すると終了値3となり、エラーメッセージが標準エラーに出力されます。

7.3.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。

メッセージ	内容
get_APItoken request error.	トークン取得のAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
set Access_token error.	暗号化DBへの更新が失敗した場合に出力されます。

7.3.7. 注意事項

- 暗号化DBに格納した情報は同一マシンからしか参照することが出来ません。この為、後続部品で暗号化DBに格納されたAPIトークンを利用する場合には同一マシンのカスタムキューに投入して参照するようにしてください。
- 発行されたトークンには有効時間があります。トークンの有効期限切れを意識してジョブネットワークを構築してください。有効期限が切れた場合の復旧として再度トークンの取得を行う実装などを検討ください。
- 拡張カスタムパラメータ「dbkey」のキーに格納されるトークンは重複が無いようご注意ください。同一ユーザーで同じキー名にトークンを格納すると、後から更新したトークンに上書きされます。（JNW毎、ユーザー毎でキー名を分けてお使いください）
- ファイルパスを指定するパラメータにおいて、ネットワークファイルパスは指定できませんのでご注意ください。
- Boxへの通信が出来ない環境から本部品を使いAPIトークンの取得は出来ません。Boxへ接続ができることを確認できた環境でご利用ください。
- 「privkey_filepath」で指定したRSA秘密鍵はパスフレーズが設定されていると利用することができません。パスフレーズ解除を行ったRSA秘密鍵を指定してください。

7.4. BOXアップロード部品

7.4.1. 部品情報

■格納パス

/Root/System/Tools/Box

■定義名

box_upload

■表示アイコン



7.4.2. 部品説明

指定したファイルをBoxに対してアップロードします。

アップロードを行う方法は3種類提供しています。

■フォルダID指定

box_parent_folderidで指定したフォルダに対してファイルのアップロードを行います。IDがすでに分かっている場合などに利用可能です。

■フォルダID+サブフォルダ指定

box_parent_folderidで指定したフォルダのサブフォルダbox_sub_foldernameに対してファイルのアップロードを行います。対象のフォルダIDがわかっていて、ファイル名で各フォルダに振り分けたい場合などに利用可能です。

■Boxフォルダ指定

box_full_folderpathで指定したBoxのフォルダパスに対してファイルのアップロードを行います。他の2つの方法と比較し、パスからフォルダIDの全検索を行うためアップロードが完了するまでに時間がかかる場合があります。

7.4.3. スクリプト

本部品にはスクリプトはありません。

7.4.4. カスタムパラメータ設定

必須	変数	説明	設定例
	box_apitoken	APIトークンの文字列を直接指定してください。本指定がある場合、共通項目で指定したJWT認証パラメータは利用されません。	abcde...(文字列を直接指定)
○	box_uploadfilename	アップロード後のファイル名を指定してください。(ローカルファイル名と異なっても問題ありません)	sample.pdf
○	local_folderpath	アップロードをするファイルがあるフォルダのパスを指定してください。	C:\tmp\samplefolder
○	local_filename	local_folderpathフォルダ内に存在するアップロードしたいファイル名を指定してください。	uploadfile.pdf

必須	変数	説明	設定例
	box_parent_folderid	ファイルをアップロードする先のBoxのフォルダIDを指定してください。	0
	box_sub_foldername	box_parent_folderidで指定した配下のフォルダにファイルを置きたい場合にフォルダ名を指定してください。	sample_folder
	box_full_folderpath	ファイルをアップロードする先のBoxのフォルダパスを指定してください。box_folderidの指定より、本パラメータでの指定が優先されます	/sample_folder/ folder_A

7.4.5. 実行結果

■アップロードに成功した場合

APIトークンの取得に成功すると終了値0となり、標準出力にAPIトークンの文字列が展開された形でスクリプトの内容が出力されます。

■必須パラメータが不足した場合

必須パラメータが不足すると終了値1となり、エラーメッセージが標準エラーに出力されます。

■APIトークンの取得に失敗した場合

APIトークンの取得に失敗すると終了値2となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■フォルダID取得APIに失敗した場合

アップロード先のフォルダIDを取得するAPIの実行に失敗すると終了値3となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■指定したBoxのフォルダが存在しなかった場合

box_full_folderpathやbox_sub_foldernameで指定したフォルダ名が存在しなかった場合、終了値4となりエラーメッセージが標準エラーに出力されます。正しいフォルダ名を確認してパラメータを設定してください。

■アップロードAPIに失敗した場合

アップロードAPIの実行に失敗すると終了値5となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

7.4.6. 主要メッセージ

メッセージ	内容
upload success.	アップロードに成功した場合に出力されます。
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
get_APIToken request error.	トークン取得のAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
get_Folders request error.	フォルダ情報取得APIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
Error:Can't Find folderid from [box_full_folderpath].	box_full_folderpathで指定した名前のフォルダが存在していない場合に出力されます。

メッセージ	内容
	box_full_folderpathで指定したパスに誤りが無いか、確認してください。
Error:Can't Find folderid from [box_sub_foldername].	box_sub_foldernameで指定した名前のフォルダが存在していない場合に出力されます。 box_sub_foldernameで指定した名前に誤りが無いか、確認してください。またbox_parent_folderidで指定した親フォルダの配下にあるかも確認してください。
upload request error.	アップロードAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。

7.4.7. 注意事項

- ファイルパスを指定するパラメータにおいて、ネットワークファイルパスは指定できませんのでご注意ください。
- 既に指定されたパスに同一ファイルが存在するとエラーとなります（Boxの仕様となります）
- アップロード可能なファイルサイズの上限は50MBとなります。

7.5. BOXダウンロード部品

7.5.1. 部品情報

■格納パス
/Root/System/Tools/Box

■定義名
box_download

■表示アイコン


7.5.2. 部品説明

指定したファイルをBoxからダウンロードします。

ダウンロードするファイルを指定する方法は2種類提供しています。

■ファイルID指定

download_fileidで指定したファイルにたいしてダウンロードを行います。IDがすでに分かっている場合などに利用可能です。

■Boxパス指定

box_download_folderpathで指定したパスの中にあるbox_download_filenameで指定したファイルに対してダウンロードを行います。パスからフォルダIDの全検索を行うためID指定よりダウンロードが完了するまでに時間がかかります。

7.5.3. スクリプト

本部品にはスクリプトはありません。

7.5.4. カスタムパラメータ設定

必須	変数	説明	設定例
	box_apitoken	APIトークンの文字列を直接指定してください。本指定がある場合、共通項目で指定したJWT認証パラメータは利用されません。	abcde...(文字列を直接指定)
○	local_folderpath	ダウンロード後に格納するローカルのフォルダパスを指定してください。	C:\tmp
	local_filename	ダウンロードしたファイルにつけるファイル名を指定してください。未指定の場合にはBox側で設定されている名前を取得して設定します。	download_output.pdf
	box_download_fileid	ダウンロードするBoxのファイルIDを指定してください。box_download_folderpathの指定より本パラメータが優先されます。	0123… (数値)
	box_download_folderpath	ダウンロードするファイルが配置されているフォルダのパスを記載してください。	/sample_folder/folder_A
	box_download_filename	box_download_folderpathで指定したフォルダ内に存在するダウンロードしたいファイル名を指定してください	download.pdf

7.5.5. 実行結果

■ダウンロードに成功した場合

APIトークンの取得に成功すると終了値0となり、指定したlocal_folderpathのフォルダにダウンロードしたファイルが生成されます。

■必須パラメータが不足した場合

必須パラメータが不足すると終了値1となり、エラーメッセージが標準エラーに出力されます。

■APIトークンの取得に失敗した場合

APIトークンの取得に失敗すると終了値2となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■フォルダID取得APIに失敗した場合

アップロード先のフォルダIDを取得するAPIの実行に失敗すると終了値3となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■指定したBoxのフォルダが存在しなかった場合

box_download_folderpathで指定したフォルダ名やbox_download_filenameで指定したダウンロードしたいファイル名が存在しなかった場合、終了値4となりエラーメッセージが標準エラーに出力されます。正しいフォルダ名を確認してパラメータを設定してください。

■ダウンロード対象のファイル名取得に失敗した場合

box_download_folderpathのみ指定を行いbox_download_filenameを指定しなかった場合、Box側のファイル名情報を取得するAPIを実行します。

このファイル名取得のAPI実行に失敗した場合、終了値5となりエラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■ダウンロードAPIに失敗した場合

ダウンロードAPIの実行に失敗すると終了値6となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■ダウンロードしたファイルの保存に失敗した場合

ダウンロードAPIの実行完了後、ローカルファイルの保存に失敗すると終了値7となり、エラーメッセージが標準エラーに出力されます。ローカルストレージの空き容量が残っているかなど、ローカルストレージの確認を行ってください。

■リダイレクト先からのダウンロードに失敗した場合

ダウンロードAPIの実行に通知されるリダイレクトアドレスからファイルのデータを読み込む際にエラーが発生すると終了値8となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

7.5.6. 主要メッセージ

メッセージ	内容
box_download success.	ダウンロードに成功した場合に出力されます。
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
get_APItoken request error.	トークン取得のAPIが失敗した場合に出力されます。

メッセージ	内容
	box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
get_Folders request error.	フォルダ情報取得APIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
Error:Can't Find folderid from [box_download_folderpath].	box_download_folderpathで指定した名前のフォルダが存在していない場合に出力されます。 box_download_folderpathで指定したパスに誤りが無いか、確認してください。
Error:Can't Find folderid from [box_download_filename].	box_download_filenameで指定した名前のフォルダが存在していない場合に出力されます。 box_download_filenameで指定した名前に誤りが無いか、確認してください。
get_FileName request error.	ファイル情報取得APIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
box_download request error.	ダウンロードAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
fileopen error. download_filename. check	ダウンロードしたファイルの保存に失敗した場合に出力されます。後続に原因のメッセージが出力されます。 ローカルストレージの空き容量が残っているかなど、ローカルストレージの確認を行ってください。
redirect_download request error.	リダイレクトアドレスからファイルのデータを読み込みに失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。

7.5.7. 注意事項

- ファイルパスを指定するパラメータにおいて、ネットワークファイルパスは指定できませんのでご注意ください。
- BoxのファイルIDは随時付与されますので、同一ファイル名、同一フォルダにアップロードしてもBoxのIDは異なりますのでご注意ください。

7.6. コンテンツID取得部品

7.6.1. 部品情報

■格納パス
/Root/System/Tools/Box

■定義名
box_getContentID

■表示アイコン


7.6.2. 部品説明

指定したBoxのパスからコンテンツIDを標準出力にフォーマット出力します。BoxのAPIを実行する際にはフォルダIDかファイルIDが必要になりますが、Boxのフォルダパス、ファイルパスからIDを取得するAPIは存在していません。

本部品はその課題を解決する部品として、フォルダIDやファイルIDをスクリプトで出力し、他のアプリケーションや環境変数にいれて後続のジョブで利用することが出来ます。

2種類の方法でコンテンツIDを取得できます。

■フォルダパス指定

box_folderpathで指定したフォルダのIDを出力します。

■フォルダパス指定+ファイル名指定

box_folderpathで指定したパスの中にあるbox_filenameで指定したファイルのIDを出力します。

7.6.3. スクリプト

実行結果を出力するテンプレートをMustache言語形式で記述します。

BoxのコンテンツIDを参照するには{{{boxContentID}}}を記述します。この{{{boxContentID}}}}の部分は指定したコンテンツIDに置き換わり表示されます。

各部品にはサンプルのテンプレートを記述していますので、参考にしてください。

7.6.4. カスタムパラメータ設定

必須	変数	説明	設定例
	box_apitoken	APIトークンの文字列を直接指定してください。本指定がある場合、共通項目で指定したJWT認証パラメータは利用されません。	abcde...(文字列を直接指定)
○	box_folderpath	IDを取得したいフォルダパスを指定してください。box_filename未指定時にはフォルダIDが、指定時にはbox_filenameで指定したファイルIDがフォーマット出力されます。	/sample_folder/ folder_A
	box_filename	box_folderpathで指定したフォルダ内に存在するファイル名を指定してください。	download.pdf

7.6.5. 実行結果

■ID取得に成功した場合

ID取得に成功すると終了値0となり、標準出力に指定したコンテンツのIDが出力されます。

■必須パラメータが不足した場合

必須パラメータが不足すると終了値1となり、エラーメッセージが標準エラーに出力されます。

■APIトークンの取得に失敗した場合

APIトークンの取得に失敗すると終了値2となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■フォルダID取得APIに失敗した場合

アップロード先のフォルダIDを取得するAPIの実行に失敗すると終了値3となり、エラーメッセージが標準エラーに出力されます。このエラーメッセージにはBoxからのレスポンスの内容も出力されます。

■指定したBoxコンテンツが存在しなかった場合

box_folderpathで指定したフォルダ名やbox_filenameで指定したファイル名が存在しなかった場合、終了値4となりエラーメッセージが標準エラーに出力されます。

7.6.6. 主要メッセージ

メッセージ	内容
parameter [XXX] is required.	必須パラメータが指定されていない場合に出力されます。
get_APItoken request error.	トークン取得のAPIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
get_Folders request error.	フォルダ情報取得APIが失敗した場合に出力されます。 box return status code:の後ろにBoxから返されたメッセージが記録されますので参照してください。
Error:Can't Find folderid from [box_folderpath].	box_folderpathで指定した名前のフォルダが存在していない場合に出力されます。 box_folderpathで指定したパスに誤りが無いか、確認してください。
Error:Can't Find fileid from [box_filename].	box_filenameで指定した名前のファイル名が存在していない場合に出力されます。 box_filenameで指定した名前に誤りが無いか、確認してください。

7.6.7. 注意事項

■ファイルパスを指定するパラメータにおいて、ネットワークファイルパスは指定できませんのでご注意ください。

■参照するBoxフォルダパスの階層の数だけBoxのAPIを実行しますので、Boxの階層が深い場合にはAPIの実行回数が多くなります。

Box APIの実行回数は契約されているライセンスの種類により制限がある場合がありますのでご注意ください。ライセンスの詳細などはご契約されているBoxのサポートまでお問い合わせください。

