App Bridge Kicker
概説

Ver.1.19.0

目次

１．はじめに 1

1.1 本書の役割 1
1.2 前提ドキュメント 1
1.3 App Bridge Kicker とは 1

２．タスク 2

2.1 タスクとステップ 2
2.2 タスク定義とタスクインスタンス 3
2.3 タスクインスタンスの生成 4
2.3.1 スケジューラによるインスタンス生成 4
2.3.2 ブラウザによるインスタンス生成 5
2.3.3 コマンドによるインスタンス生成 5
2.3.4 App Bridge Monitor によるインスタンス生成 5
2.4 ステップインスタンスの抽出条件 6
2.5 実行可能モジュール 7
2.6 ステータス 8
2.6.1 タスクステータス 8
2.6.2 ステップステータス 8
2.7 タスク定義のコピー 9

３．App Bridge Kicker Agent 10

3.1 Agent の種類 10
3.1.1 Windows Agent 10
3.1.2 Linux Agent 10
3.1.3 センタ Agent 10
3.2 ノード定義 11
3.3 ステップインスタンス実行 12
3.3.1 実行ノードによるステップインスタンス実行 12
3.3.2 実行ノードタグによるステップインスタンス実行 13
3.3.3 実行タグによるタスク実行サーバの冗長化 13
3.3.4 サーバ IDによる実行サーバの確認 14
3.3.5 リトライ 14
3.4 スケーリングサーバ 15
3.4.1 スケーリングサーバでのステップインスタンス実行 15
3.4.2 スケーリングタグによるステップインスタンス実行 16
3.5 スレッド制御 17
3.5.1 メインスレッドとワーカスレッド 17
3.5.2 ワーカスレッドの動作 18
3.5.3 キャンセル機能 18
3.5.4 キャンセル要求に伴うプロセス強制終了 18
3.6 App Bridge センタとの通信障害 19
3.7 Agent 終了処理 19
3.7.1 終了処理手順 19

3.7.2 ステップ終了処理の遅延実行 19
3.7.3 終了処理の長時間化 19
3.8 サーバシャットダウン要求 19
3.9 サーバシャットダウン要求の拒否 19
3.10 タスクインスタンスの保存期間 20
4．ライブラリ 21
4.1 ライブラリとは 21
4.2 ライブラリの種類 21
4.3 パブリックライブラリ 22
4.3.1 パブリックモジュール 22
4.3.2 センタ Agent のパブリックモジュール 22
4.3.3 利用者サーバのパブリックモジュール 22
4.3.4 提供されるパブリックモジュール 22
4.4 プライベートライブラリ 22
5．ステップ間データ転送 23
5.1 ステップ間データ転送とは 23
5.2 データ転送方式 23
5.3 データアップロード仕様 23
5.4 データダウンロード仕様 23
5.5 データ転送速度の目標値 23
5.6 データ転送関連パブリックモジュール 24
5.7 ステップ間データの保存期間 24
5.8 App Bridge Transporter 25
6．クラウドストレージ 26
6.1 クラウドストレージの用途 26
6.2 データの分割と暗号化 26
6.3 クラウドストレージへの通信 26
7．App Bridge センタからの通報 27
7.1 異常検知の通報 27
7.2 タイムアウト通報 27
7.2.1 Waiting タイムアウト 27
7.2.2 Running タイムアウト 27
7.3 通報先 27
8．課金計算 28
8.1 単価 28
8.2 プラン基本料 28
8.3 利用者サーバ費用 28
8.4 利用サーバステップ費用 28
8.4.1 課金区分 28
8.4.2 課金区分タイプ 1 28
8.4.3 課金区分タイプ 2 28
8.5 センタ Agent ステップ費用 29

8.5.1 課金区分 29
8.5.2 課金区分タイプ1 29
8.5.3 課金区分タイプ2 29
8.6 パブリックモジュール費用 29
8.6.1 課金区分 29
8.6.2 課金区分タイプ1 29
8.6.3 課金区分タイプ2 29
8.7 データアップロード費用 29
8.8 データダウンロード費用 29
8.9 モジュールダウンロード費用 29
8.10 課金区分の選択 30
8.10.1 想定している選択 30
8.10.2 通常利用の計算例 30
8.10.3 繰返し利用の課金計算 30

１．はじめに

1.1 本書の役割

本書は、App Bridge Kicker の概要について記載しています。

1.2 前提ドキュメント

本書は、以下のドキュメントを前提に記載されています。本書をご覧いただく前にご一読下さい。

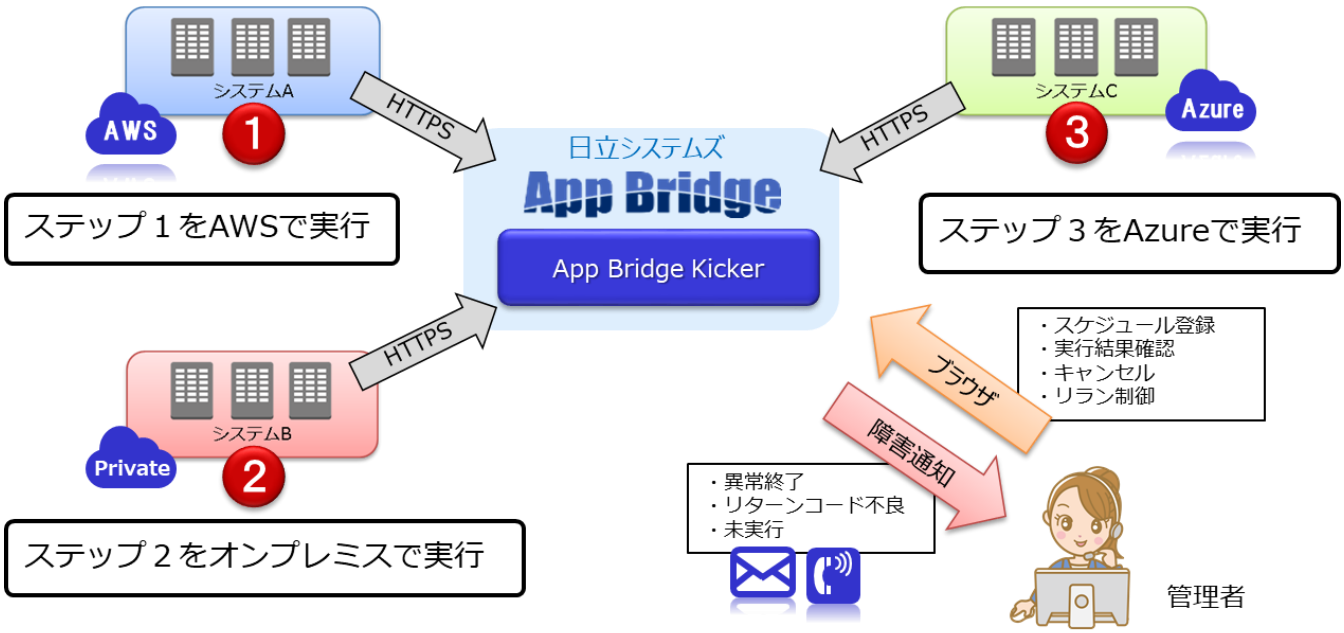
- App Bridge ファーストステップガイド
- App Bridge 管理者ガイド

1.3 App Bridge Kicker とは

App Bridge Kicker は、クラウド間をまたがるタスク制御（プログラム実行制御）を実現するミドルウェアサービスです。

利用者は、タスクを実行させたいサーバに、App Bridge Kicker Agent をインストールすることで、タスク制御を利用できます。App Bridge Kicker Agent は、App Bridge センタへの HTTPS 通信のみを使用しており、多くの環境でご利用いただけます。

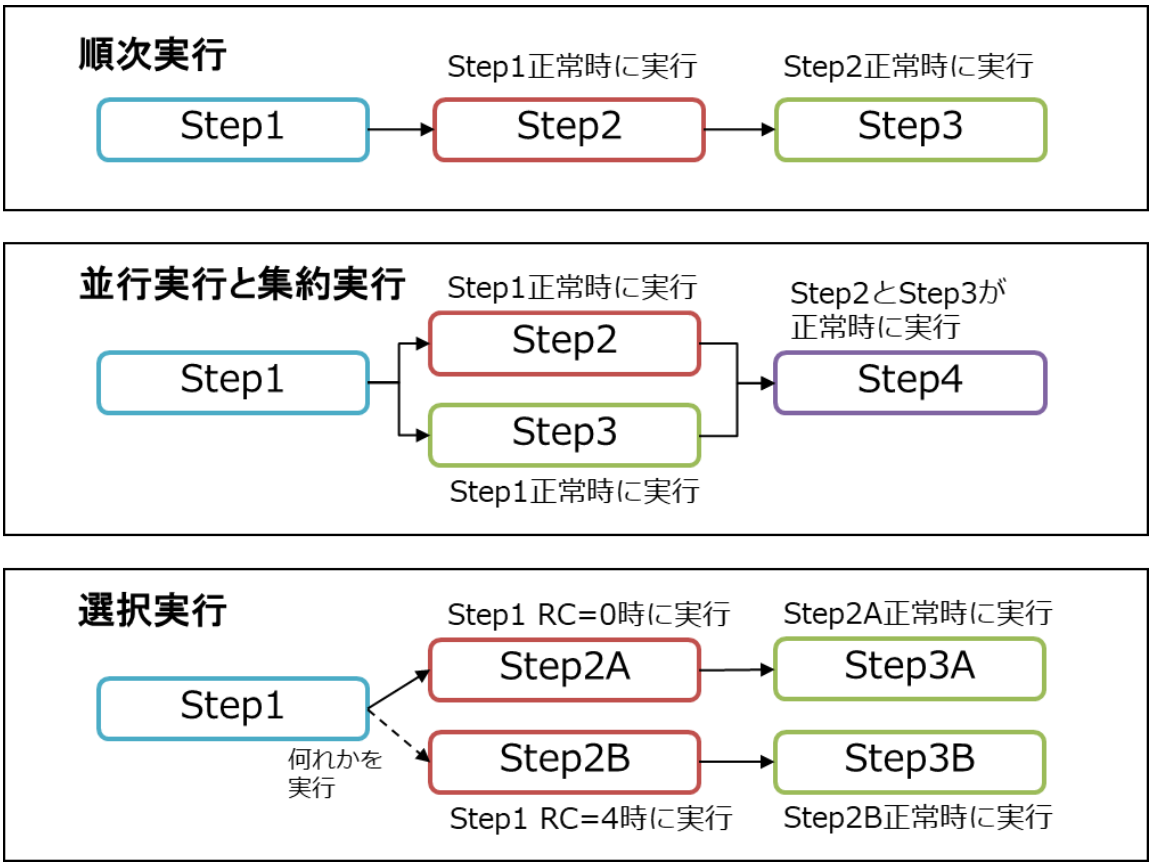
App Bridge Kicker の各種設定は、ブラウザから実施いただけます。タスク制御中に検出された障害は、App Bridge センタから電子メール、電話などで通知されます。



２．タスク

2.1 タスクとステップ

App Bridge Kicker は、「タスク」を実行単位とします。1 つのタスクには、複数のステップが定義できます。ステップに前後関係を設定することで、順次実行、並列実行、集約実行、選択実行などの実行順序制御が可能です。1 つのタスクに定義できる最大ステップ数は 128 です。

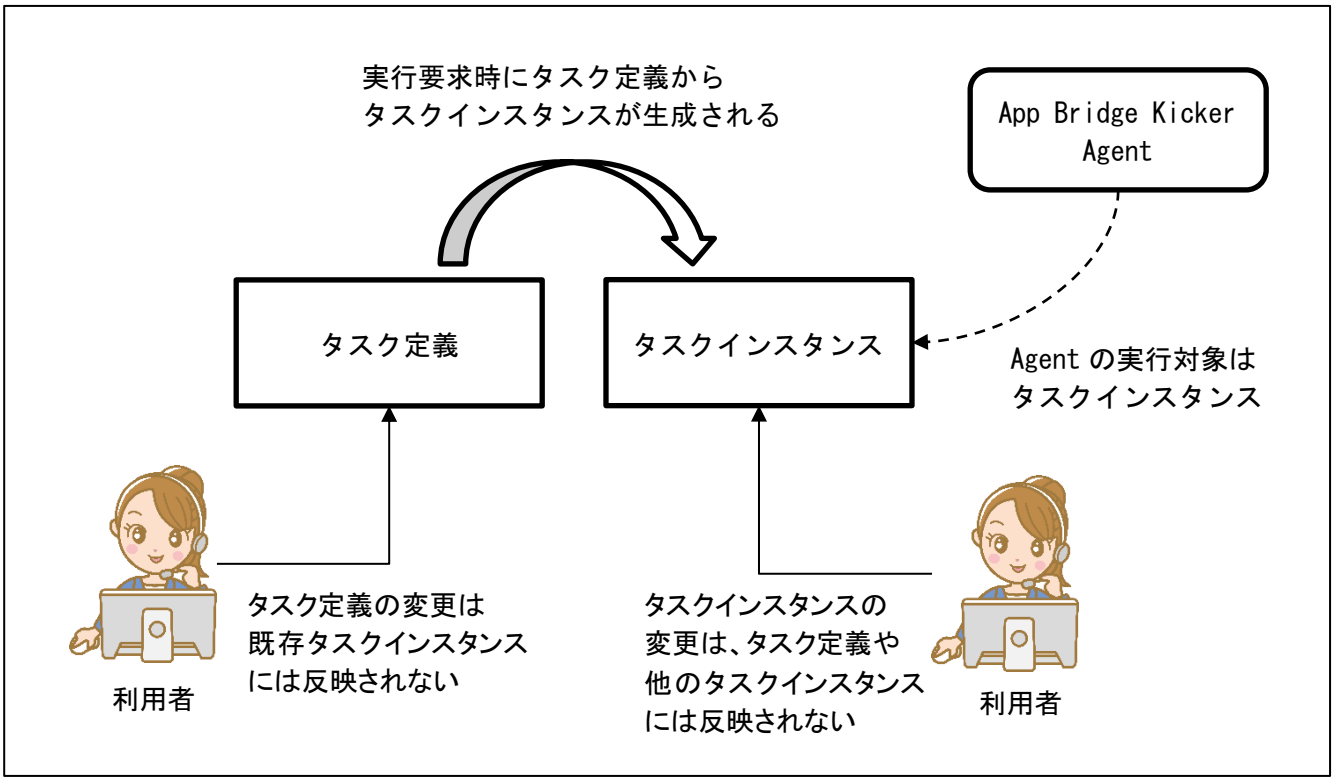


2.2 タスク定義とタスクインスタンス

利用者は、タスクを「タスク定義」として登録します。タスクの実行を要求した時点で、「タスク定義」から「タスクインスタンス」が生成されます。タスクインスタンスは、タスク定義のコピーです。

App Bridge Kicker Agent はタスクインスタンスを処理対象とします。1 つのタスク定義を 3 度実行した場合、3 つのタスクインスタンスが生成され、実行結果は各々のタスクインスタンスに格納されます。

タスク定義とタスクインスタンスは、独立した存在です。タスク定義を変更しても、既に生成されているタスクインスタンスの内容は変更されません。同様にタスクインスタンスを変更しても、タスク定義の内容は変更されません。



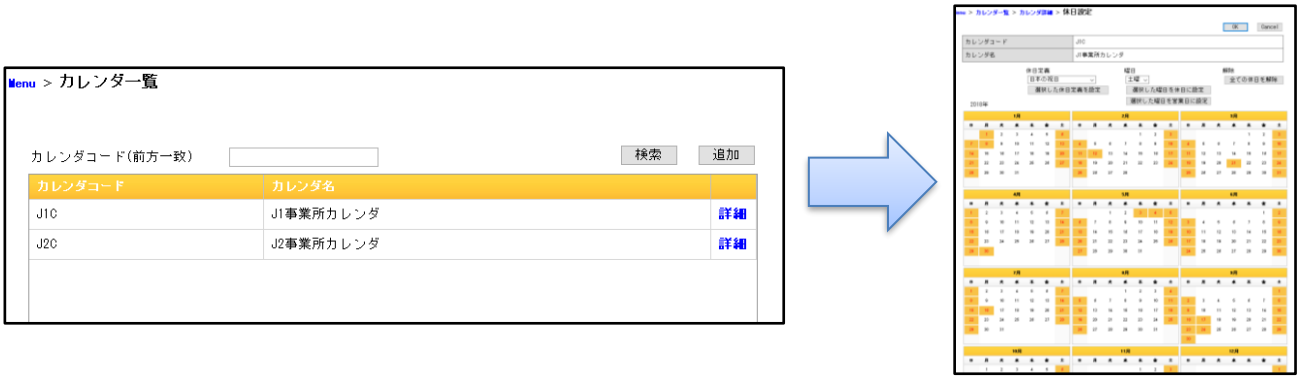
2.3 タスクインスタンスの生成

2.3.1 スケジューラによるインスタンス生成

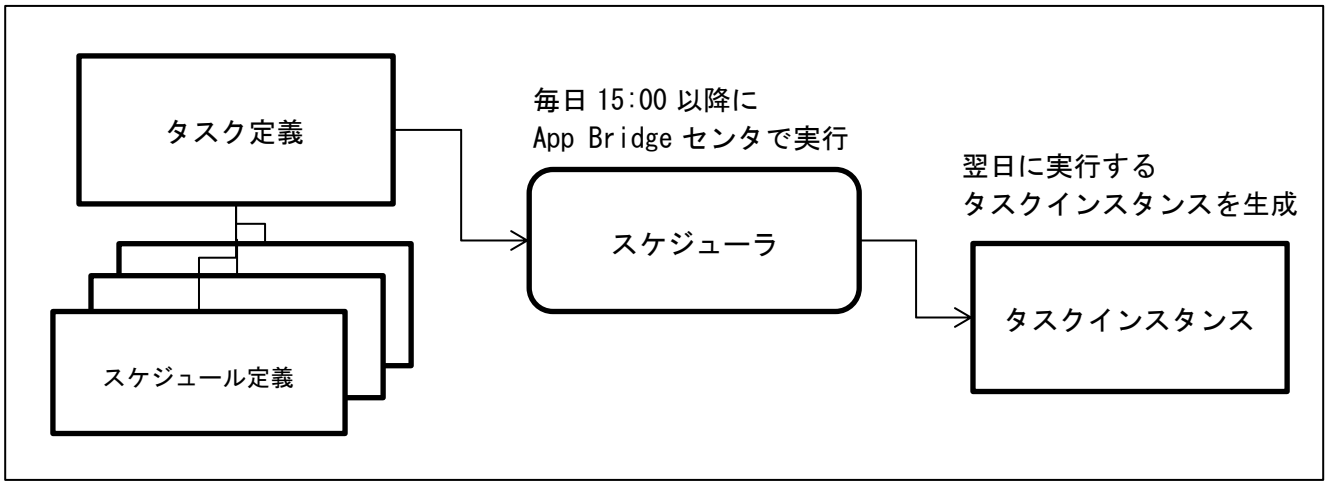
タスク定義には、1 つまたは複数のスケジュール定義を設定できます。スケジュール定義には、日次、週次、月次などの種類があり、カレンダに基づくスケジューリングを提供します。

スケジュールタイプ	実行区分	休日動作	備考
日次実行	毎日実行	・実行しない ・実行する	
週次実行	指定曜日に実行	・実行しない ・実行する ・前営業日に実行する ・翌営業日に実行する	第三日曜日など、週番号の指定が可能
月次実行	指定の暦日に実行	・実行しない ・実行する ・前営業日に実行する ・翌営業日に実行する	各月 1 日に実行などの指定が可能
営業日実行	指定の営業日に実行	ー	
休日実行	休日に実行	ー	

スケジュール定義には、カレンダを設定できます。スケジューラは、指定されたカレンダに基づき、実行日を決定します。



スケジューラは、毎日 15:00 以降に App Bridge センタで実行されます。利用者による再スケジュールも可能です。



2.3.2 ブラウザによるインスタンス生成

利用者は、App Bridge センタの Web サイトにアクセスし、ブラウザでタスクインスタンスを生成できます。インスタンス生成時に、スケジュール時刻、追加情報の設定も可能です。

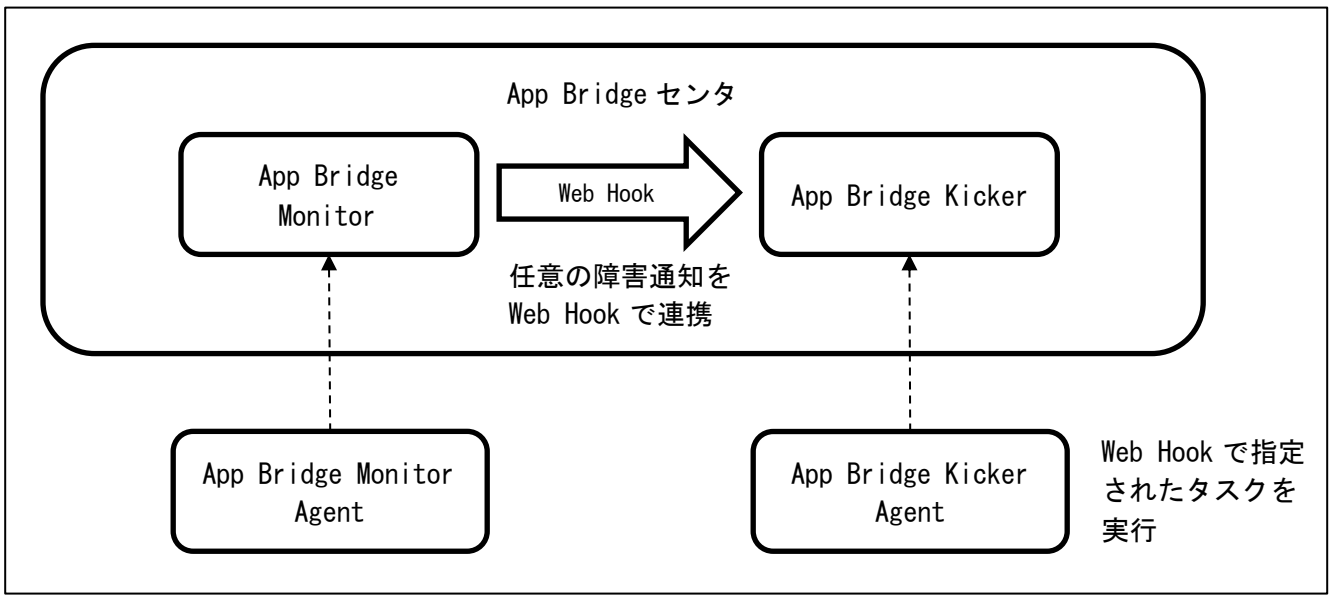
2.3.3 コマンドによるインスタンス生成

利用者は、App Bridge Kicker が提供するコマンド（LB4CreateInstance）を使用し、タスクインスタンスを生成できます。コマンドを使用することで、他のシステムから App Bridge Kicker に連携することが容易になります。

コマンドの詳細は、「App Bridge Kicker 開発者ガイド」を参照下さい。

2.3.4 App Bridge Monitor によるインスタンス生成

App Bridge Monitor の Web Hook 機能でタスクインスタンスを生成することができます。
これにより、特定の障害を検知したとき、任意のタスクを起動することができます。
App Bridge Monitor の設定は、App Bridge Monitor 通知サービスガイドを参照下さい。



タスクインスタンスの無限生成にご注意下さい

App Bridge Monitor で生成されたタスクインスタンスが異常終了した場合、通知メッセージが作成され、App Bridge Monitor で処理されます。この通知メッセージが同じタスクインスタンスを生成した場合、無限にタスクインスタンスの生成と実行が繰り返されます。

タスクインスタンスの無限生成を避けるため、App Bridge Monitor から生成されるタスクインスタンスを事前検証いただくともに、App Bridge Monitor 通知振分け条件においてもタスクインスタンスの無限生成を避けるよう条件を設定して下さい。

タスクインスタンスの無限生成が発生した場合、該当の App Bridge Monitor/Web Hook を無効とすることで事態を収束できます。本手順も事前に確認下さい。

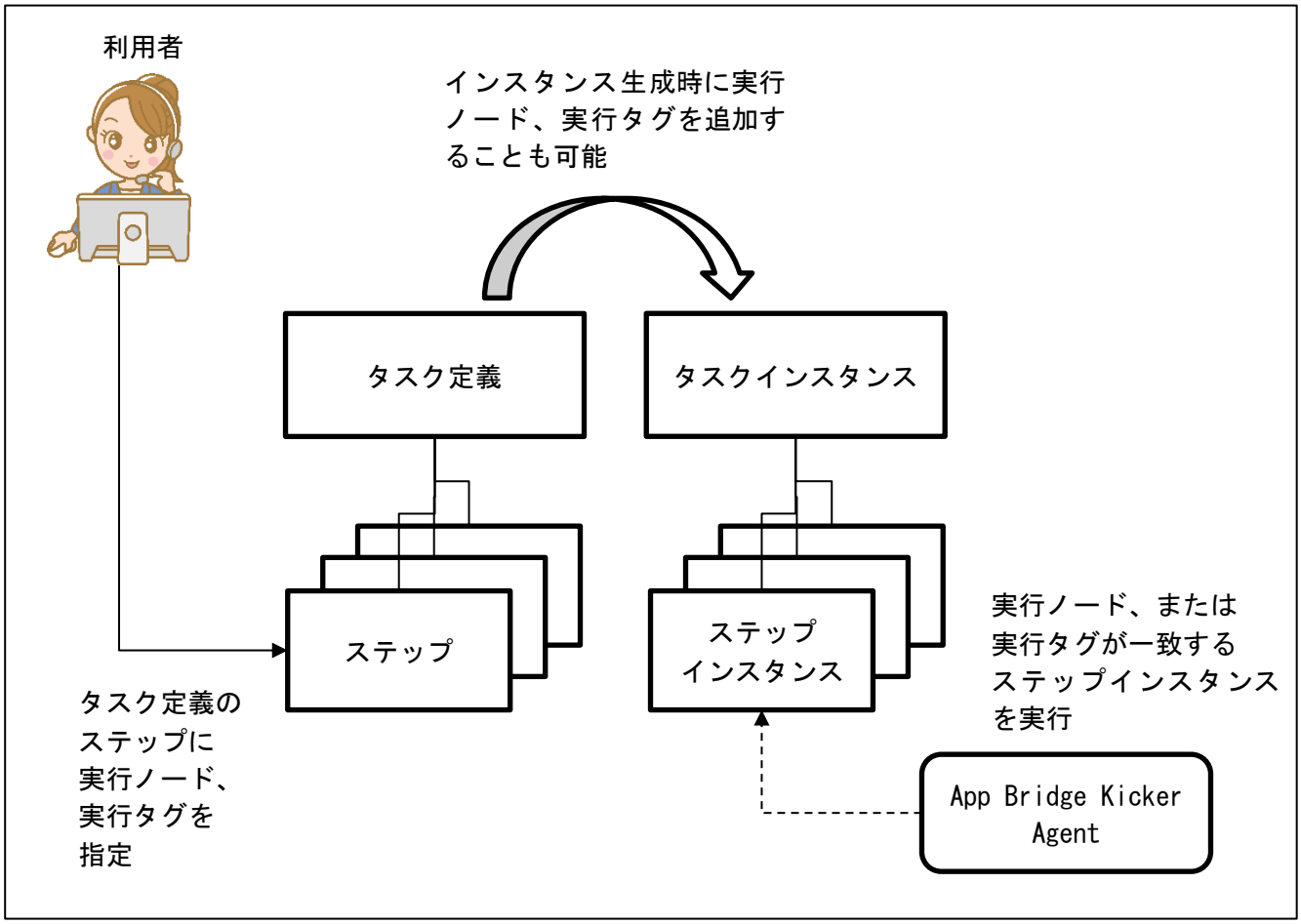
2.4 ステップインスタンスの抽出条件

ステップインスタンスは、タスク実行サーバの App Bridge Kicker Agent に抽出され、実行されます。ステップには「実行ノード」「実行タグ」が設定され、これらを基準に「タスク実行サーバ」が決定されます。「タスク実行サーバ」は、App Bridge Kicker Agent がインストールされた利用者サーバです。

「実行ノード」は「タスク実行サーバ」を一意に示す ID です。ステップインスタンスは、指定された「タスク実行サーバ」で実行されます。1 つのステップに複数の「実行ノード」を設定した場合、何れかの「タスク実行サーバ」でステップインスタンスが実行されます。

「実行タグ」は、ステップに付与されたノードタグ情報です。ステップインスタンスは、実行タグと同じ「ノードタグ」を持つ「タスク実行サーバ」で処理されます。

「実行ノード」、「実行タグ」はタスクインスタンス生成時に追加することができます。これにより利用者は、タスクインスタンス生成時点でタスク実行サーバを指定することができます。



2.5 実行可能モジュール

ステップインスタンスで実行できるモジュールは、以下の 4 種類です。

モジュールの種類	内容	備考
パブリックモジュール	後述のパブリックライブラリからダウンロードしたモジュールを実行します	
プライベートモジュール	後述のプライベートライブラリからダウンロードしたモジュールを実行します	
スクリプト	ステップ定義に記述されたスクリプトを実行します	※1、※2
ローカルファイル	タスク実行サーバに存在するファイルを実行します	※3

※1：App Bridge Kicker Windows Agent がサポートしているスクリプトは以下のとおりです
- バッチファイル
- PowerShell
- Windows Script Host（JScript、VBScript）

※2：App Bridge Kicker Linux Agent がサポートしているスクリプトは以下のとおりです
- Shell スクリプト

※3：Windows Agent で実行できるファイルの拡張子は、.bat、.exe、.com の 3 種類です

2.6 ステータス

2.6.1 タスクステータス

タスクインスタンスの実行状態は、タスクステータスとして管理されます。タスクステータスは以下のとおりです。

タスクインスタンス	内容	備考
Waiting	全てのステップインスタンスが実行待ちとなっている状態です	
Running	1 つ以上のステップインスタンスが実行中となっている状態です	
Complete	実行すべきステップインスタンスが全て正常に終了した状態です	
Abort	実行すべきステップインスタンスが全て終了し､1 つ以上の異常終了を検知した状態です	
Cancel	実行すべきステップインスタンスが全て終了し､1 つ以上のキャンセル終了を検知した状態です	異常終了よりキャンセル終了が優先します

2.6.2 ステップステータス

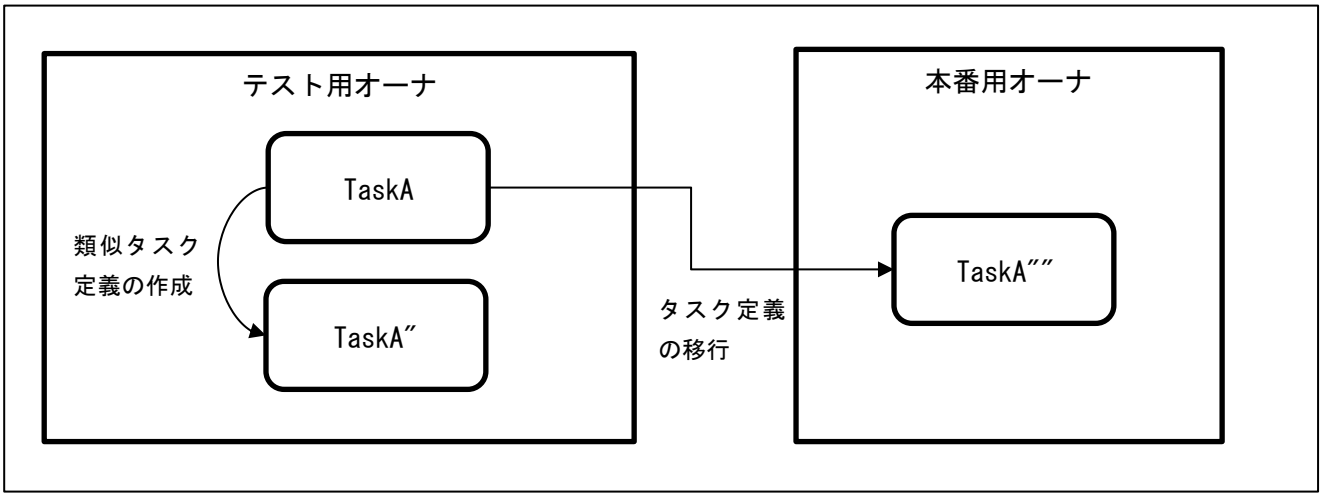
ステップインスタンスの実行状態は、ステップステータスとして管理されます。ステップインスタンスは以下のとおりです。

ステップインスタンス	内容	備考
Waiting For Time	起動予定時刻の到達を待っている状態です。既に起動予定時刻に到達している場合､Agent からの抽出を待っている状態です	
Waiting For Step	先行ステップの終了を待っている状態です	
Running	実行中の状態です	
Canceling	利用者からキャンセル要求を受け付け､Agent でのキャンセル実行を待っている状態です	
Complete	正常終了した状態です	
Abort	異常終了した状態です	
Canceled	キャンセル終了した状態です	
Disabled	先行ステップの異常終了などを受け、ステップの実行が抑止された状態です	

2.7 タスク定義のコピー

作成したタスク定義は、同じオーナ内、または他のアプリケーションオーナ（※1）にコピーすることができます。

これにより、類似タスク定義の作成が容易となり、用途別に準備したオーナ間のタスク定義移植が可能となります。



※1：App Bridge では、アプリケーションオーナの単位で各種設定、管理が実施されます。詳細は、「App Bridge ファーストステップガイド」を参照下さい。

３. App Bridge Kicker Agent

3.1 Agent の種類

3.1.1 Windows Agent

App Bridge Kicker は、マイクロソフト社の Windows サーバ向けのエージェント（以下 Windows Agent）を提供します。利用者は任意の Windows サーバに Windows Agent をインストールすることで、当該サーバでタスクを実行することができます。

App Bridge Kicker Windows Agent の対象サーバ、前提条件は、「Windows Agent サポートリスト」を参照下さい。

App Bridge Kicker Windows Agent のインストール方法は、「App Bridge Kicker Windows Agent インストールガイド」を参照下さい。

3.1.2 Linux Agent

App Bridge Kicker は、Linux サーバ向けのエージェント（以下 Linux Agent）を提供します。利用者は任意の Linux サーバに Linux Agent をインストールすることで、当該サーバでタスクを実行することができます。

App Bridge Kicker Linux Agent の対象サーバ、前提条件は、「Linux Agent サポートリスト」を参照下さい。

App Bridge Kicker Linux Agent のインストール方法は、「App Bridge Kicker Linux Agent インストールガイド」を参照下さい。

3.1.3 センタ Agent

App Bridge センタから供給する一部のモジュールは、App Bridge センタのセンタ Agent で実行することができます。
センタ Agent を使用する場合、利用者サーバに App Bridge Kicker Agent をインストールする必要はありません。

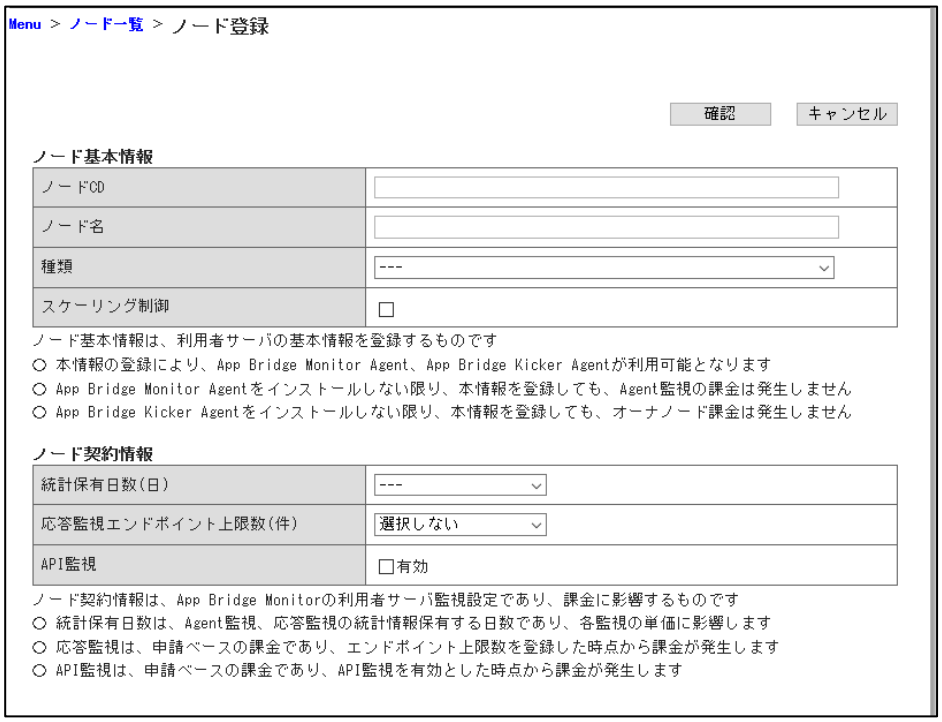
詳細は、後述の「パブリックモジュール」を参照下さい。

3.2 ノード定義

App Bridge Kicker Agent をインストールするためには、利用者サーバをノードとして App Bridge に登録する必要があります。

App Bridge にノードを登録することで、App Bridge Kicker Agent が利用できることに加え、App Bridge Monitor によるサーバ監視も可能となります。

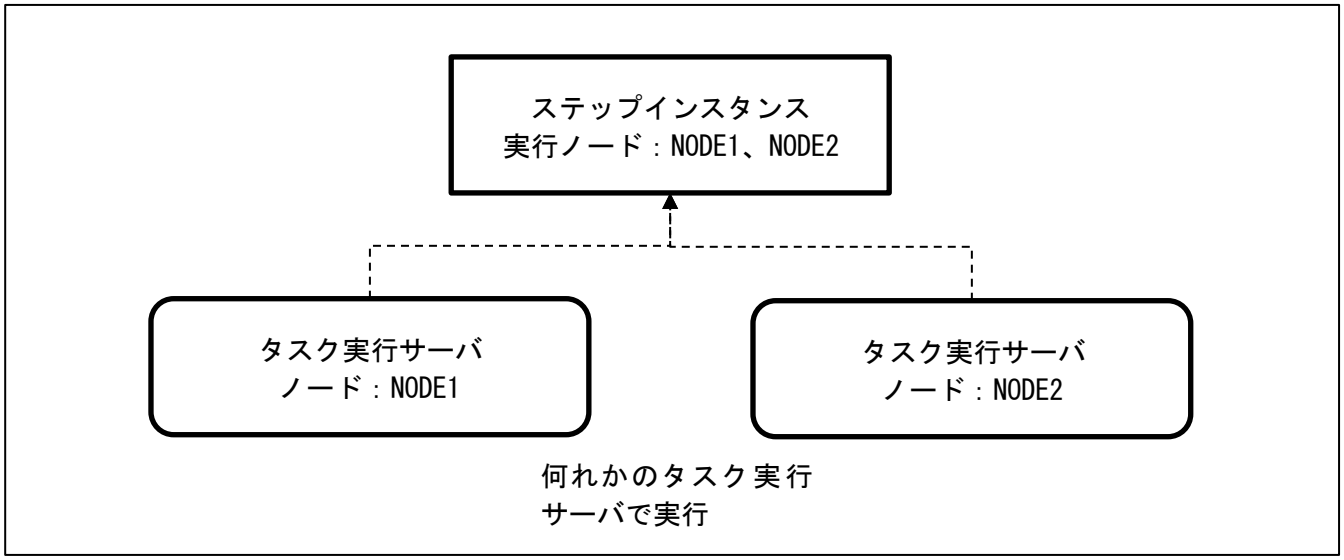
App Bridge ノード登録については、「App Bridge 管理者ガイド」を参照下さい。



3.3 ステップインスタンス実行

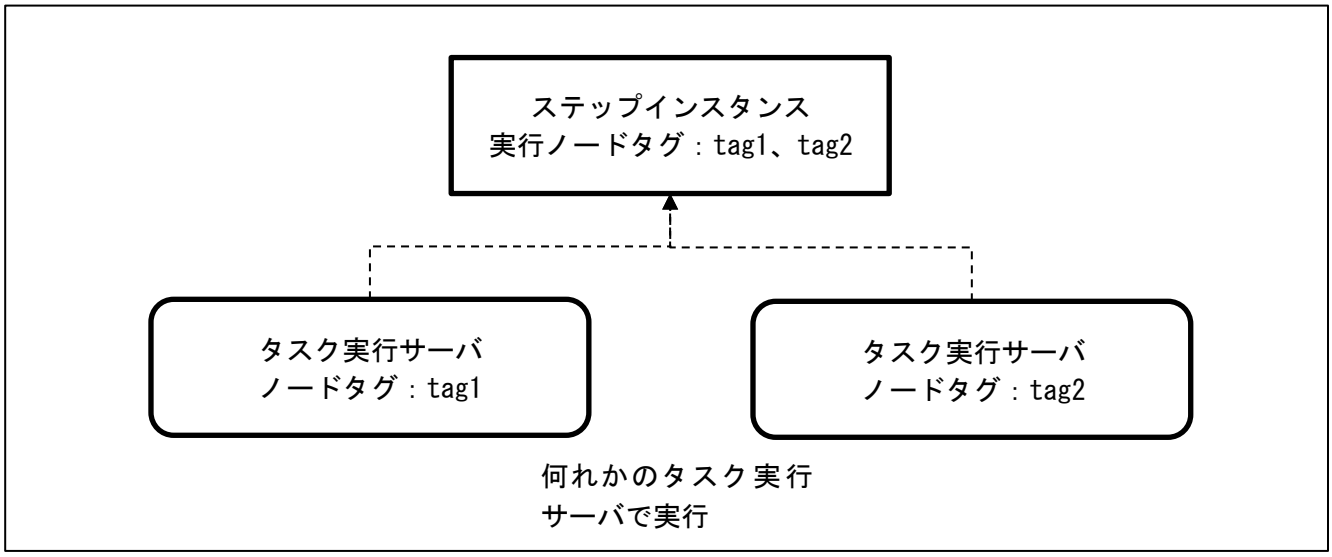
3.3.1 実行ノードによるステップインスタンス実行

App Bridge Kicker Agent は、自身と同じ「実行ノード」を持つステップインスタンスを実行します。ステップインスタンスに、複数の実行ノードが指定されていた場合、ステップインスタンスは、何れかのノード（タスク実行サーバ）で実行されます。



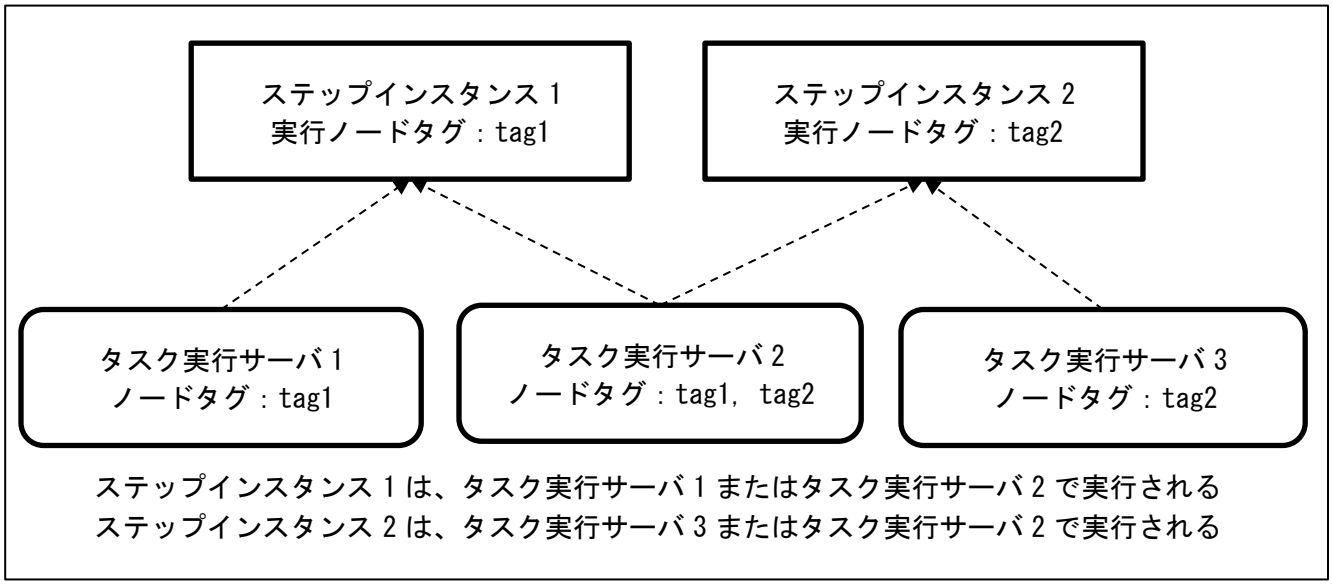
3.3.2 実行ノードタグによるステップインスタンス実行

App Bridge Kicker Agent は、自身に設定されたノードタグと同じ「実行タグ」を持つステップインスタンスを実行します。ステップインスタンスに、複数の実行タグが指定されていた場合、ステップインスタンスは、同じノードタグを持つ何れかのノード（タスク実行サーバ）で実行されます。



3.3.3 実行タグによるタスク実行サーバの冗長化

複数のタスク実行サーバに同じ実行タグを設定することで、タスク実行サーバを冗長化することができます。



3.3.4 サーバ ID による実行サーバの確認

App Bridge Kicker Agent をインストールしたサーバインスタンスには、当該サーバインスタンスを一意に示すための、サーバ ID が付与されます。

このサーバ ID は、ステップインスタンス実行履歴に「実行サーバ ID」として記録され、実行サーバを特定する際に利用でき、課金処理（実行サーバ数の確認）にも使用されます。

App Bridge Kicker Agent は起動時にサーバ ID を確認し、付与されていなければ生成します。一度生成されたサーバ ID は記録され、維持されますが、当該サーバが所属するオーナ、ノードを変更した場合に再付与されます。

また、App Bridge Kicker Agent は、起動時に下記条件を確認し、該当すればサーバ ID を再付与します。この動作は、App Bridge Kicker Agent 導入後のサーバイメージ作成に対応するものです。

Agent	サーバ ID 再付与条件	備考
App Bridge Kicker Windows Agent	前回起動時のサーバ SID と今回起動時のサーバ SID が一致しなかったとき	
App Bridge Kicker Linux Agent	前回起動時の IP アドレスと今回起動時の IP アドレスが全て一致しなかったとき	前回と今回の IP アドレスを比較し、1 つでも一致すれば再付与しない

サーバインスタンスに割り当てられたサーバ ID の確認方法は、「App Bridge Kicker Windows Agent インストールガイド」「App Bridge Kicker Linux Agent インストールガイド」を参照下さい。

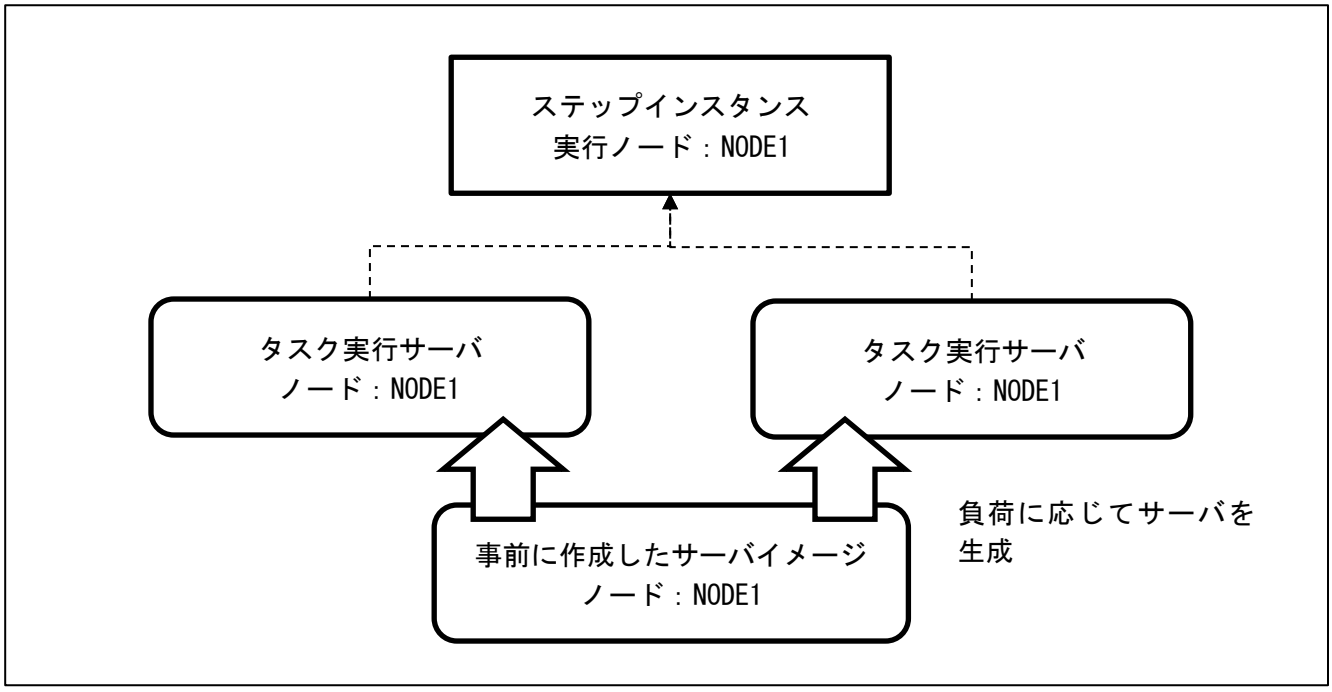
3.3.5 リトライ

ステップステータスが Abort、Cancel のステップインスタンスはリトライすることができます。リトライできる有効期限は、タスクインスタンスのステータス最終更新日時から 10 日間です。

3.4 スケーリングサーバ

3.4.1 スケーリングサーバでのステップインスタンス実行

スケーリング制御されたサーバ（※1）をタスク実行サーバとした場合、スケーリングされたサーバの何れかで同じ実行ノードを持つステップインスタンスを実行します。



スケーリング制御で、動的にサーバインスタンス数を増減させる場合、サーバインスタンス停止時の動作に注意が必要です。詳細は後述の「サーバシャットダウン時の注意事項」を参照下さい。

※1：スケーリング制御とは、1 つのサーバイメージから複数のサーバインスタンスを生成し、状況に応じてサーバインスタンスを増減するものです

3.4.2 スケーリングタグによるステップインスタンス実行

スケーリングタグは、スケーリングされたタスク実行サーバの中から、特定のサーバインスタンスを一意に示すものです。

スケーリングタグをステップ定義の実行タグ（または追加実行タグ）に指定することで、特定のサーバインスタンスでステップインスタンスを実行できます。

スケーリングタグは、対象サーバのオーナ、ノード ID、サーバのコンピュータ（ホスト名）をハッシュして作成されます。サーバインスタンスに割り当てられたスケーリングタグの確認方法は、「App Bridge Kicker Windows Agent インストールガイド」「App Bridge Kicker Linux Agent インストールガイド」を参照下さい。

スケーリングタグは、App Bridge Monitor との連携でも使用されます。スケーリングされたサーバでエラーが発生した場合、スケーリングタグを使用し、対象サーバでステップインスタンスを実行します。詳細は、「App Bridge Monitor 通知サービスガイド/App Bridge Kicker 連携」を参照下さい。

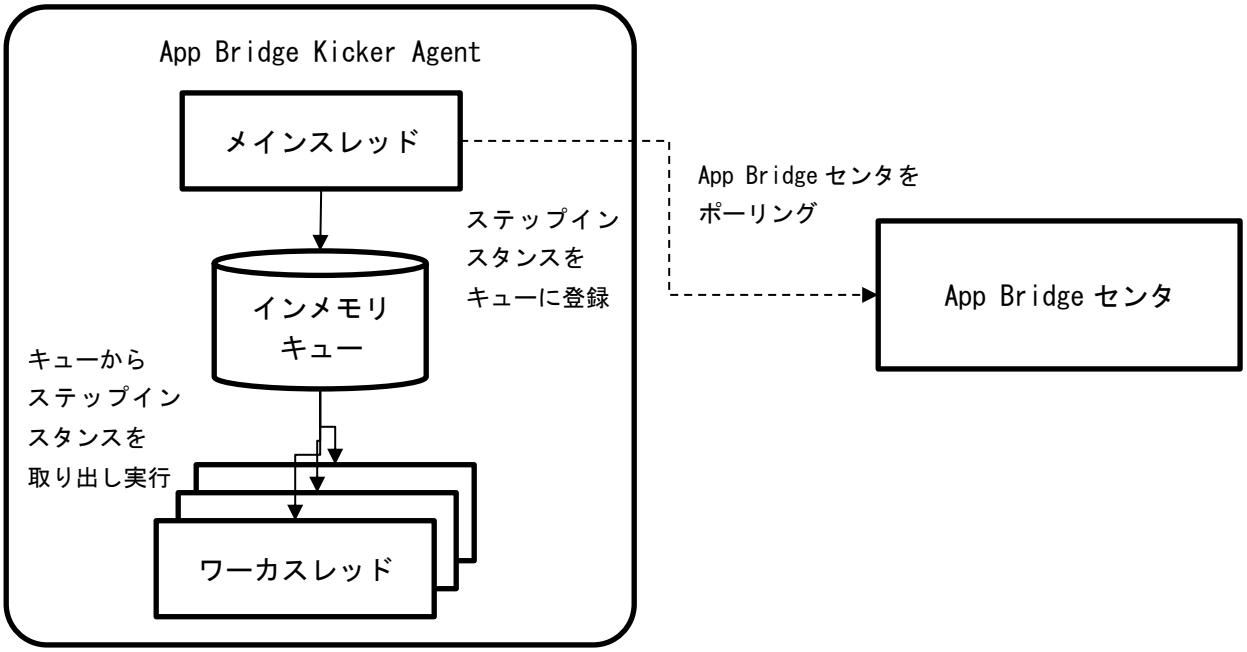
3.5 スレッド制御

3.5.1 メインスレッドとワーカスレッド

App Bridge Kicker Agent は、1 つのメインスレッドと複数のワーカスレッドを保有します。ワーカスレッド数は設定可能であり、初期値は CPU コア数×4 に設定されています。

メインスレッドは、App Bridge センタをポーリングし、対象ステップインスタンスを発見するとインメモリキューに登録します。App Bridge センタのポーリング間隔は動的に変化し、最長で 30 秒です。

ワーカスレッドは、インメモリキューをポーリングし、対象を発見するとステップインスタンスの処理を実行します。



3.5.2 ワーカスレッドの動作

ワーカスレッドは、以下の順序でステップインスタンスを処理します。

#	処理	内容	備考
1	ステップ開始宣言	ステップインスタンスの実行開始を App Bridge センタに通知します	ステップステータスが Running に変更されます
2	モジュールダウンロード	実行対象がパブリックモジュール、またはプライベートモジュールのとき、モジュールをダウンロードします	キャッシュに存在するとき、ダウンロードは実行されません
3	データダウンロード	データダウンロードが指示されている場合、先行ステップのアップロードデータをダウンロードします	先行ステップが複数あるとき、全てダウンロードします
4	プロセス起動	指定モジュールをプロセス起動します	
5	データアップロード	データアップロードが指示されている場合、出力データをアップロードします	異常終了の場合、アップロードされません
6	ステップ終了宣言	実行結果を App Bridge センタに通知します	ステップステータスが変更され、後続ステップのステップステータスも更新されます

3.5.3 キャンセル機能

ステップインスタンス実行中、ワーカスレッドは、キャンセル確認スレッドを起動します。キャンセル確認スレッドは 15 秒間隔で実行中ステップインスタンスのキャンセル要求有無を App Bridge センタに問合せます。ステップインスタンスのキャンセル要求は、キャンセル確認スレッドからワーカスレッドに伝わり、この時点でステップインスタンスの実行は中断されます。

3.5.4 キャンセル要求に伴うプロセス強制終了

プロセス起動中にキャンセル要求を検知した場合、App Bridge Kicker Agent は、実行モジュールのプロセスを強制終了します。実行モジュールが子プロセスを起動していた場合、当該子プロセスも強制終了されます。プロセスの強制終了は再帰的に実行され、関連する全プロセスの強制終了を試みます。

プロセス強制終了の結果は、App Bridge Kicker サイトで確認することができます。キャンセル処理を実施した場合、プロセス強制終了状況を確認するようお願いします。

3.6 App Bridge センタとの通信障害

何らかの要因で、App Bridge センタと通信できなくなった場合、App Bridge Kicker Agent はリトライを実施します。通信できない状態が続いた場合、App Bridge Kicker Agent は通信エラーを発生させます。

ステップ終了処理（データアップロードとステップ終了宣言）で通信エラーが発生した場合、一定のインターバルをおき、再処理されます。

3.7 Agent 終了処理

3.7.1 終了処理手順

App Bridge Kicker Agent は、終了が指示されると、ステップインスタンスの抽出を中止し、実行中ステップインスタンスの終了を待ちます。実行中ステップインスタンスが終了し、全ワーカスレッドの停止を確認すると、メインスレッドが終了し、App Bridge Kicker Agent は終了します。

3.7.2 ステップ終了処理の遅延実行

Agent 終了処理中に、ステップ終了処理（データアップロードとステップ終了宣言）で通信エラーが発生した場合、ステップ終了処理は再処理されず、Agent は終了します。未処理のステップ終了処理は、ストレージに記録され、Agent 起動後に再実行されます。

3.7.3 終了処理の長時間化

実行中プロセスが終了しない場合や、大きなデータを転送している場合など、Agent 終了処理は長時間化します。Agent 終了処理が長時間化した場合、タスクの実行状態を確認し、必要に応じ、実行中ステップインスタンスにキャンセルを要求して下さい。

3.8 サーバシャットダウン要求

シャットダウン要求が設定されているステップインスタンスが正常終了した場合、App Bridge Kicker Agent は、以下の処理を実施します。

シャットダウン要求	処理内容	備考
Shutdown	Agent 終了処理を実施し、OS にシャットダウンを要求する	
Restart	Agent 終了処理を実施し、OS に再起動を要求する	
Stop Agent	Agent 終了処理を実施する	Agent 停止のみ

3.9 サーバシャットダウン要求の拒否

App Bridge Kicker Agent の設定により、サーバシャットダウン要求を拒否することができます。これにより、誤ったサーバシャットダウンを防ぐことができます。

3.10 タスクインスタンスの保存期間

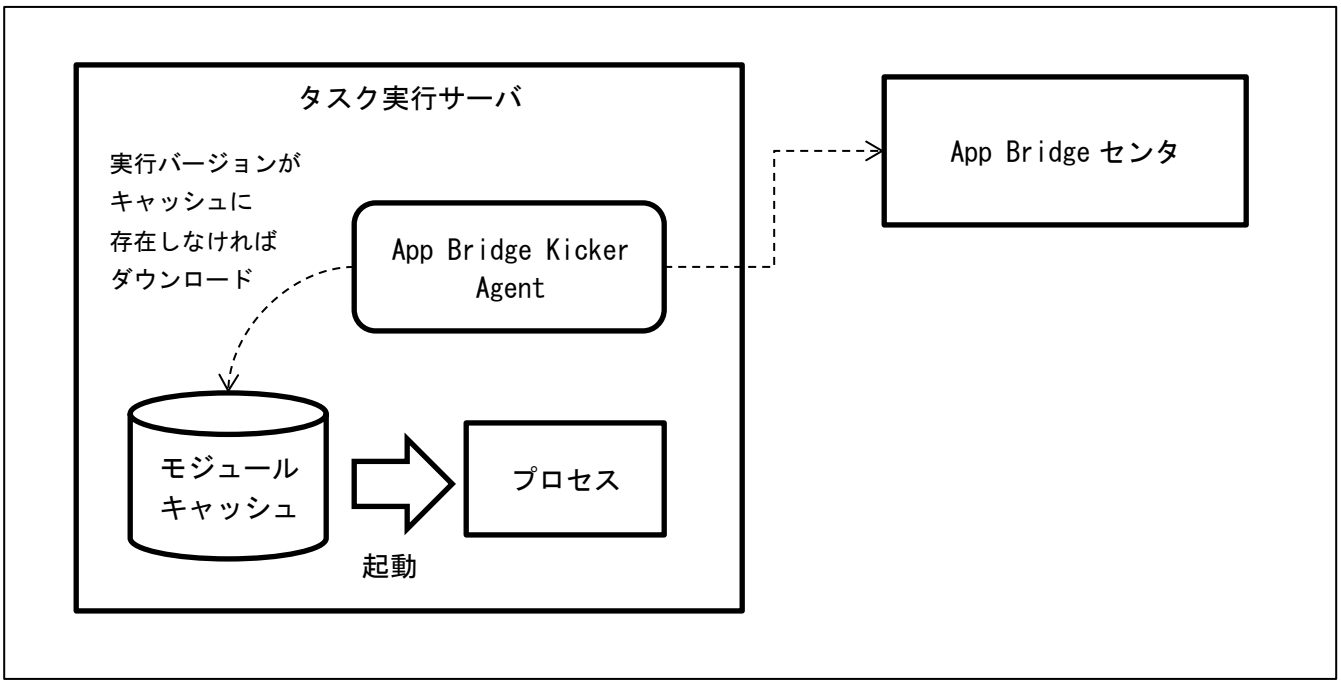
タスクインスタンスは、タスクインスタンスのステータス最終更新日時から 90 日間で削除されます。

４．ライブラリ

4.1 ライブラリとは

App Bridge Kicker のライブラリ機能は、予め、利用者がライブラリにアップロードしたモジュールを App Bridge Kicker Agent がダウンロードし、実行するものです。

App Bridge Kicker Agent は、ダウンロードモジュールを実行サーバ内にキャッシュし、バージョンが変更された場合のみダウンロードを実施します。この機能により、複数のタスク実行サーバに対するモジュール展開を容易に実施でき、モジュールのバージョン管理を確実とします。



4.2 ライブラリの種類

ライブラリには、パブリックライブラリ、プライベートライブラリの 2 つがあります。

パブリックライブラリは、全利用者が使用可能なパブリックモジュールを格納するものであり、App Bridge センタで管理されます。

プライベートライブラリは、オーナごとに準備されたライブラリであり、利用者が開発したプライベートモジュールが格納されます。プライベートライブラリは、オーナごとに最大 3GB の容量が確保されています。

ライブラリの種類	モジュールの開発、管理	使用可能範囲
パブリックライブラリ	App Bridge センタ	全オーナ
プライベートライブラリ	利用者	当該オーナのみ

4.3 パブリックライブラリ

4.3.1 パブリックモジュール

パブリックライブラリに格納されたモジュールをパブリックモジュールと呼びます。パブリックモジュールは、App Bridge センタから供給されるモジュールであり、全ての利用者が使用できます。

4.3.2 センタ Agent のパブリックモジュール

パブリックモジュールの中には、App Bridge センタのセンタ Agent で稼働するものがあります。

例えば､Amazon EC2 サーバを起動･停止するパブリックモジュールをセンタ Agent で稼働させることで、テスト環境の起動・停止をサーバレス（利用者サーバを使用せず）に実施できます。

4.3.3 利用者サーバのパブリックモジュール

パブリックモジュールの中には、利用者サーバの Windows Agent、Linux Agent で稼働するものがあります。

4.3.4 提供されるパブリックモジュール

提供されるパブリックモジュールは、以下の URL で公開されています。

http://kicker-doc.app-bridge.com/App-Bridge-Kicker-Public-Module-List.pdf

4.4 プライベートライブラリ

プライベートライブラリに格納されたモジュールをプライベートモジュールと呼んでいます。プライベートモジュールは、利用者が登録するモジュールであり、登録オーナ以外からは使用できません。

プライベートモジュールの登録方法は、「App Bridge Kicker 開発者ガイド」を参照下さい。

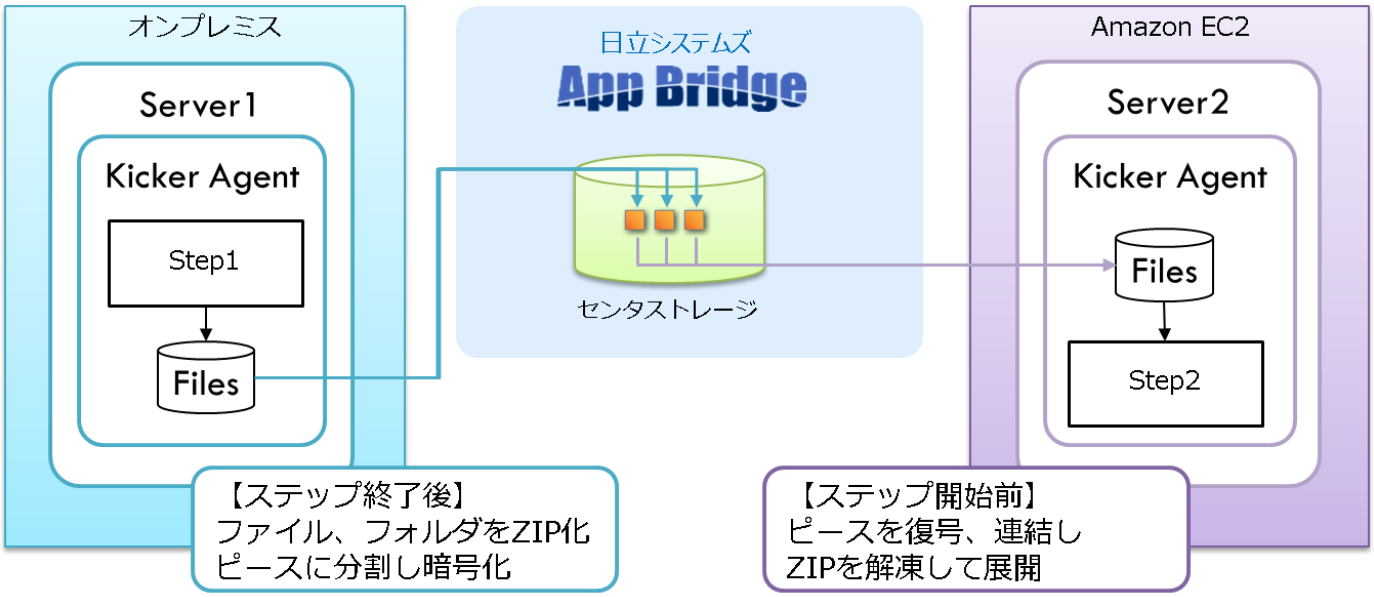
５．ステップ間データ転送

5.1 ステップ間データ転送とは

ステップ間データ転送とは、利用者が実行環境を意識することなく、ステップ間でデータを引き継ぐための機能です。この機能により、クラウドを跨ったステップでデータを引き継ぐことができます。

5.2 データ転送方式

転送されるデータ（ステップが出力したファイル、フォルダ）は、ZIP 圧縮され、ピースと呼ばれる断片に分割、暗号化され、並列に送受信されます。ピースは、ピースごとのキーで暗号化されており、単独で復号することは困難です。



5.3 データアップロード仕様

アップロードできる最大のデータ量は、圧縮後の ZIP ファイルサイズで 500GB までです。

5.4 データダウンロード仕様

データをダウンロードするステップは、隣接するアップロードステップのデータをダウンロードします。ダウンロード仕様の詳細は、「App Bridge Kicker 開発者ガイド」を参照下さい。

5.5 データ転送速度の目標値

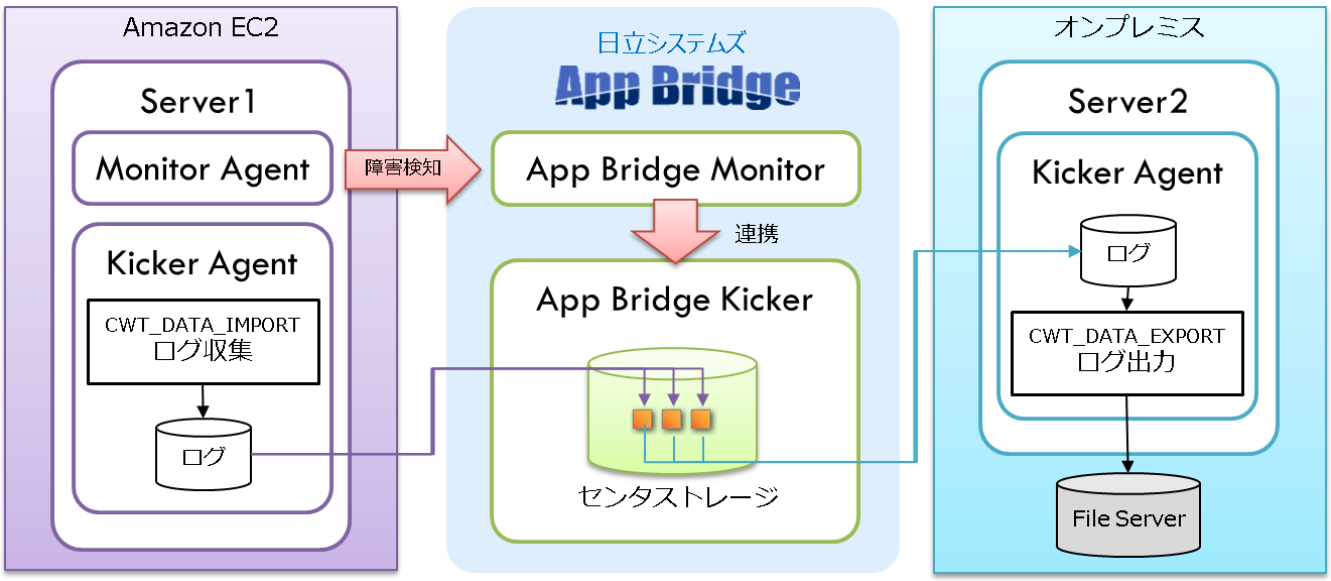
App Bridge Kicker Agent は、指定されたデータ転送速度の目標値に従って、並列転送数を設定します。データ転送速度目標値の初期値は 128Mbps です。

5.6 データ転送関連パブリックモジュール

データ転送用に以下のパブリックモジュールが提供されます。利用者は当該モジュールを使用することで、データの収集を容易に開始できます。

分類	パブリックモジュール	内容	備考
DATA_TRANSPORT	CWT_DATA_IMPORT	指定データの転送	
	CWT_DATA_EXPORT	転送データの出力	

パブリックモジュールを使用することで、データ転送を伴う処理を容易に実現できます。下図例では、App Bridge Monitor からの通知を受け、対象サーバでログを収集し、社内ファイルサーバに展開しています。



5.7 ステップ間データの保存期間

ステップ間データは、タスクインスタンス終了からリトライ有効期限（10 日間）超過後に順次削除されます。

5.8 App Bridge Transporter

ステップ間データ転送は、App Bridge Transporter（データ転送サービス）の簡易版として、実装されています。以下のケースでは、App Bridge Transporter のご利用をご検討下さい。

- 500GB を超えるデータ量
- データの長期保存
- データの世代管理
- より確実なデータ転送
  - 並列データ転送数の動的調整によるパフォーマンス
  - 中断されたポイントからのデータ転送再開

６．クラウドストレージ

6.1 クラウドストレージの用途

App Bridge Kicker は、以下のデータ保管にクラウドストレージを使用します。

データの種類	クラウドストレージに格納する内容	備考
ステップログ	Agent 出力するログ情報	100 万文字を超える部分は切り捨てられます
ステップ標準出力	実行モジュールが出力した標準出力	100 万文字を超える部分は切り捨てられます
ステップ標準エラー出力	実行モジュールが出力した標準エラー出力	100 万文字を超える部分は切り捨てられます
パブリックライブラリ	パブリックモジュール	
プライベートライブラリ	プライベートモジュール	
ステップ間転送データ	転送データ	

6.2 データの分割と暗号化

クラウドストレージを使用する場合、App Bridge Kicker は、データを一定サイズの断片（ピース）に分割し、ピースごとに異なるキーで暗号化します。この分割、暗号化は、App Bridge 内で実施され、復号に必要となる暗号キーとマップ情報は App Bridge センタに保管されます。

6.3 クラウドストレージへの通信

App Bridge は、クラウドストレージとして、Amazon S3 を使用しています。App Bridge Kicker Agent から当該クラウドストレージへの https アクセスが発生します。

７. App Bridge センタからの通報

7.1 異常検知の通報

App Bridge センタは、ステップインスタンスの異常（異常終了、キャンセル終了など）を検知したとき、利用者への通報を実施します。全ての通報は、「App Bridge 通知メッセージガイド/８. センタ発信の通知メッセージ」を参照下さい。

7.2 タイムアウト通報

App Bridge センタは、2 つのタイムアウトを通報します。

7.2.1 Waiting タイムアウト

ステータスが Waiting For Time のステップインスタンスが、起動予定時刻に到達し、指定時間を経過しても実行が開始されないとき、App Bridge センタから利用者に Waiting タイムアウトが通知されます。

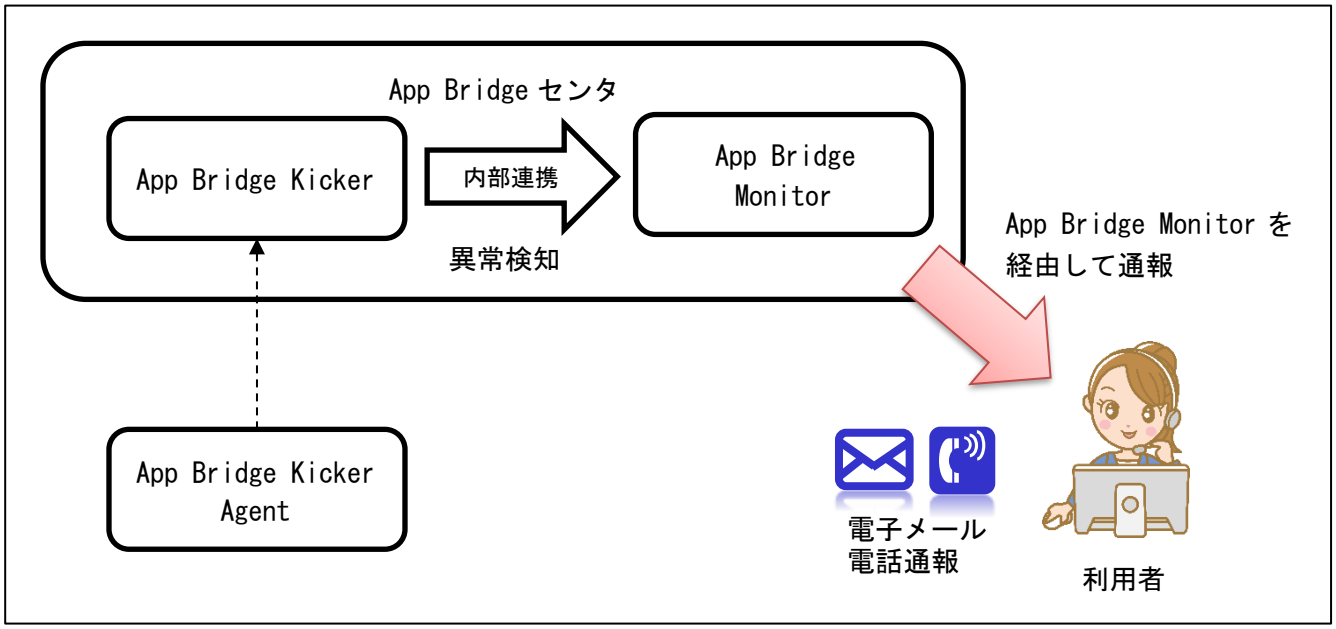
この通報が発生するとき、実行サーバまた App Bridge Kicker Agent の停止、App Bridge センタとの通信障害などが発生している可能性があります。

7.2.2 Running タイムアウト

ステータスが Running のステップインスタンスが、指定時間を経過しても終了しないとき、App Bridge センタから利用者に Running タイムアウトが通知されます。
この通報が発生するとき、想定時間を超過してもプログラムが終了しない、または App Bridge センタとの通信障害により実行結果を送信できないなどの状態が考えられます。

7.3 通報先

App Bridge センタからの通報には、App Bridge Monitor の通知先機能が使用されます。これにより、任意アドレスへの電子メール送信、任意番号への電話発信（有料）、Slack 連携などが可能となっています。



８．課金計算

8.1 単価

App Bridge Kicker は利用量に応じた従量課金であり、利用月ごとに、以下の単価で利用料を計算します。アプリケーションオーナを使用し、複数オーナを利用されている場合、全オーナの利用量を合算し、課金計算が実施されます。

月額費用		課金基準	プランA	プランB	プランC	プランD
プラン基本料		定額	0円	10,000円	50,000円	100,000円
利用者サーバ費用		ステップ実行した利用者サーバ数	500円/サーバ	サーバ20台まで0円 400円/超過サーバ	サーバ50台まで0円 250円/超過サーバ	サーバ100台まで0円 100円/超過サーバ
利用者サーバステップ費用	タスクの課金区分タイプ１	利用者サーバステップ実行回数	5円/回	500回まで0円 4円/超過回数	1,500回まで0円 2円/超過回数	10,000回まで0円 1円/超過回数
	タスクの課金区分タイプ２	利用者サーバステップスケジュール回数	10円/回	8円/回	4円/回	2円/回
センタAgentステップ費用	タスクの課金区分タイプ１	センタAgentステップ実行回数	10円/回	500回まで0円 8円/超過回数	1,500回まで0円 4円/超過回数	10,000回まで0円 2円/超過回数
	タスクの課金区分タイプ２	センタAgentステップスケジュール回数	40円/回	30円/回	20円/回	10円/回
パブリックモジュール費用	タスクの課金区分タイプ１	パブリックモジュールステップ実行回数	10円/回	500回まで0円 5円/超過回数	1,500回まで0円 2円/超過回数	10,000回まで0円 1円/超過回数
	タスクの課金区分タイプ２	パブリックモジュールステップスケジュール回数	20円/回	10円/回	4円/回	2円/回
データアップロード費用		データアップロードサイズ	30円/GB	50GBまで0円 25円/超過GB	250GBまで0円 20円/超過GB	500GBまで0円 10円/超過GB
データダウンロード費用		データダウンロードサイズ	60円/GB	50GBまで0円 50円/超過GB	250GBまで0円 40円/超過GB	500GBまで0円 30円/超過GB
モジュールダウンロード費用		モジュールダウンロードサイズ	60円/GB	10GBまで0円 50円/超過GB	50GBまで0円 40円/超過GB	100GBまで0円 30円/超過GB

8.2 プラン基本料

ご利用のオーナ単位で、プランを決定します。

アプリケーションオーナを使用し、複数オーナを利用されている場合、プランは課金オーナで設定いただけます。

8.3 利用者サーバ費用

当該月で、1 度でもステップ実行した利用者サーバ数に応じた費用です。センタ Agent は含まれません。センタ Agent のみを利用した月の利用者サーバ必要はゼロとなります。

8.4 利用サーバステップ費用

8.4.1 課金区分

利用サーバステップ費用は、課金区分に応じて計算されます。課金区分には、タイプ 1 とタイプ 2 があり、タスク定義ごとに設定します。課金計算は、課金区分ごとに実行結果を集計し、利用料を算出します。

8.4.2 課金区分タイプ 1

利用者サーバのステップ実行回数に応じた費用です。

8.4.3 課金区分タイプ 2

日次スケジューラ（または随時スケジューラ）が利用者サーバ向けステップ定義をスケジュールした回数に応じた費用です。詳細は後述の「課金区分の選択」を参照して下さい。

8.5 センタ Agent ステップ費用

8.5.1 課金区分

センタ Agent ステップ費用は、課金区分に応じて計算されます。課金区分には、タイプ 1 とタイプ 2 があり、タスク定義ごとに設定します。課金計算は、課金区分ごとに実行結果を集計し、利用料を算出します。

8.5.2 課金区分タイプ 1

センタ Agent のステップ実行回数に応じた費用です。

8.5.3 課金区分タイプ 2

日次スケジューラ（または随時スケジューラ）がセンタ Agent 向けステップ定義をスケジュールした回数に応じた費用です。詳細は後述の「課金区分の選択」を参照して下さい。

8.6 パブリックモジュール費用

8.6.1 課金区分

パブリックモジュール費用は、課金区分に応じて計算されます。課金区分には、タイプ 1 とタイプ 2 があり、タスク定義ごとに設定します。課金計算は、課金区分ごとに実行結果を集計し、利用料を算出します。

8.6.2 課金区分タイプ 1

パブリックモジュールのステップ実行回数に応じた費用です。

8.6.3 課金区分タイプ 2

日次スケジューラ（または随時スケジューラ）がパブリックモジュール指定のステップ定義をスケジュールした回数に応じた費用です。詳細は後述の「課金区分の選択」を参照して下さい。

8.7 データアップロード費用

アップロードしたデータサイズに応じた費用です。当該月のアップロードサイズを合計し、GB 未満を切り上げます。

8.8 データダウンロード費用

ダウンロードしたデータサイズに応じた費用です。当該月のダウンロードサイズを合計し、GB 未満を切り上げます。

8.9 モジュールダウンロード費用

利用者サーバにダウンロードしたモジュールサイズに応じた費用です。モジュールは Agent にキャッシュされ、再利用されます。モジュールに変更が生じるたびにダウンロードが実施されます。当該月のダウンロードサイズを合計し、GB 未満を切り上げます。

8.10 課金区分の選択

8.10.1 想定している選択

利用者サーバステップ費用、センタ Agent ステップ費用、パブリックモジュール費用は、タスク定義で定めた課金区分に従って、費用が算出されます。

課金区分タイプ 1 は、実行回数に従い、課金計算するものであり、通常のご利用を想定しています。
課金区分タイプ 2 は、スケジュール回数に従い、課金計算するものであり、繰返し実行のご利用を想定したものです。繰返し実行とは、スケジュール定義の実行時刻において、一定間隔実行（5 分～30 分）を設定したものです。

8.10.2 通常利用の計算例

プラン A の利用者サーバステップ費用を例に、通常利用のタイプ 1 とタイプ 2 を比較すると次のとおりとなります。何れもタスク内のステップは 5 ステップとします。

（１）通常の利用の課金計算

（１）タイプ 1 の場合
実行回数：1 回/日
1 回×30 日＝30 回/月
30 回×5 ステップ＝150 回
利用者サーバステップ費用：150 回×単価 5 円＝ 750 円

（２）タイプ 2 の場合
スケジュール回数：1 回/日
1 回×30 日＝30 回/月
30 回×5 ステップ＝150 回
利用者サーバステップ費用：150 回×単価 10 円＝ 1,500 円

8.10.3 繰返し利用の課金計算

プラン A の利用者サーバステップ費用を例に、繰返し利用のタイプ 1 とタイプ 2 を比較すると次のとおりとなります。何れもタスク内のステップは 5 ステップとします。

（１）タイプ 1 の場合
実行回数：時間当実行回数 12 回×24 時間＝288 回/日
288 回×30 日＝8,640 回/月
8,640 回×5 ステップ＝43,200 回
利用者サーバステップ費用：43,200 回×単価 5 円＝ 216,000 円

（２）タイプ 2 の場合
スケジュール回数：1 回×30 日＝30 回/月
30 回×5 ステップ＝150 回
利用者サーバステップ費用：150 回×単価 10 円＝ 1,500 円