

LifeKeeper for Linux

Tomcat Generic ARK スクリプト仕様書

9 版



目次

1. 概要	3
2. サポート動作環境	3
8.1. Tomcat Generic ARK のサポートポリシー	3
3. 前提条件	5
4. 基本動作	6
4.1. 起動処理(restore)	6
4.2. 停止処理(remove)	6
4.3. 監視処理(quickCheck)	6
4.4. 回復処理(recover)	6
5. 障害時の切り替え動作	7
5.1. Tomcat リソース障害時	7
5.2. ノード障害時	7
6. スクリプト仕様	8
6.1 監視スクリプト(quickCheck)	8
6.2 起動スクリプト(restore)	8
6.3 停止スクリプト(remove)	9
6.4 回復スクリプト(recover)	10
7. スクリプトの実行時間	11
7.1 restore、remove、recover	11
7.2 quickCheck	11
8. スクリプトのパラメータ	12
8.1. 必須パラメータの設定	12
8.2. /etc/default/Lifekeeper でパラメータを設定する場合	13
8.3. 各スクリプト (quickCheck、recover、remove、restore) でパラメータを設定する場合	13
8.4. カスタマイズ可能なパラメータ一覧	14
9. 免責事項	15

改版履歴

2013年 12月 19日	初版
2014年 1月 27日	2版
2014年 3月 6日	3版
2014年 7月 7日	4版
2015年 4月 1日	5版
2017年 5月 31日	6版
2017年 6月 12日	7版
2020年 11月 25日	8版
2020年 12月 24日	9版

1. 概要

本書は Tomcat サービス向けの Generic ARK スクリプトの動作詳細を解説します。あらゆる環境において正常に動作することを保証するものではありません。必要に応じて変更可能なパラメータを修正し、ご利用ください。以降、本文書内でスクリプトと記載している箇所は、Tomcat サービス向けの Generic ARK スクリプトを指すものとします。

2. サポート動作環境

2.1 Tomcat Generic ARK のサポートポリシー

Tomcat Generic ARK のサポートポリシーは以下の通りです。

[サポート対象 OS]

LifeKeeper のサポートに準じます。

[サポート対象 Tomcat]

- Tomcat5
- Tomcat6
- Tomcat7

[留意点]

- 保護対象とする Tomcat のメジャーバージョンは 1 種類です。複数のメジャーバージョンを扱うことはできません。
- 本 Generic ARK は、OS が提供している Tomcat の RPM パッケージを使用して設定した Tomcat 環境で動作検証を行っています。Tomcat のソースファイルからコンパイルした環境は動作確認していません。
- 動作検証が行われていない方法でインストールされた Tomcat 環境において、Tomcat に関連した問題が発生した場合は、OS が提供している RPM からインストールする Tomcat に変更することをお願いする場合があります。予めご了承ください。
- Amazon EC2 環境で本 Generic ARK を使用することができます。
なお、Elastic IP を利用している場合は、保護対象のアプリケーションをすべての IP に対して Listen する設定が必要です。
- LifeKeeper Single Server Protection for Linux で、本 Generic ARK を使用することができます。

[参考情報]

Tomcat Generic ARK の各スクリプトは、以下の環境で動作検証を行っております。

- RHEL5.8(Kernel-2.6.18-308.el5) + Tomcat(5-5.5.23-0.el5) + LifeKeeper for Linux v8.2
- RHEL5.10(Kernel-2.6.18-371.el5) + Tomcat(5-5.5.23-0jpp.40.el5) + LifeKeeper for Linux v8.2.1
- RHEL6.4(Kernel-2.6.32-358.el6) + Tomcat(6-6.0.24-49.el6) + LifeKeeper for Linux v8.2
- RHEL6.5(Kernel-2.6.32-431.el6) + Tomcat(6-6.0.24-62.el6) + LifeKeeper for Linux v8.2.1
- RHEL 7 (Kernel-3.10.0-123.el7) + Tomcat (7.0.42-4.el7) + LifeKeeper for Linux v8.4.0
- RHEL7.3 (Kernel-3.10.0-514.el7) + Tomcat (7.0.69-10.el7) + LifeKeeper for Linux v9.1.2

3. 前提条件

Tomcat リソースを作成する前に、以下の前提条件を満たすことが必要です。

- 各ノードの OS が同一のリリースであること
- 各ノードにおいて pkill コマンド、ss コマンド、col コマンド、及び curl コマンドが利用できること
- 各ノードに同一のバージョンの Tomcat がインストールされていること
- 各ノード上の Tomcat の設定が同一であること
- 各ノードにおいて、OS 起動時の Tomcat の自動起動が無効にされていること
- Tomcat リソース導入前に、各ノードにて手動で Tomcat の起動、停止が行えること
スクリプトの中で停止と起動は OS の service コマンドを使用しています。service コマンドを使用して保護対象とする Tomcat の停止や起動が正常に行えることを確認してください。
- クラスタシステムで Tomcat のポートを Listen するために、IP アドレスが確保されていること
(初期値ではループバックアドレス 127.0.0.1 を使用します。IP リソースによる切り替え可能な仮想 IP アドレスを指定する場合は、依存関係を作成し、IP リソースを Tomcat リソースの下位に配置する必要があります。)
- Tomcat の動作に必要なポートが開放されていること
(初期値では TCP の 8080 ポートで接続を待ち受けます。)
- 各ノードに同一の LifeKeeper for Linux(ライセンスの含む)がインストールされており、コミュニケーションパスの設定等、リソース作成前の基本設定が完了していること。
- /etc/default/LifeKeeper に本スクリプトに必須のパラメータが設定してあること
必須パラメータについては、「8 スクリプトのパラメータ」を参照してください。

4. 基本動作

Tomcat リソースでは 4 つの動作を行います。

4.1. 起動処理(restore)

Tomcat リソースを ISP(サービス稼動状態)にするために、Tomcat サービスの起動処理を行います。この動作は LifeKeeper の GUI クライアントまたは CUI で Tomcat リソースの起動処理を実行したときが該当します。起動処理に失敗した場合は、Tomcat リソースは OSF(サービス障害状態)になります。Tomcat リソースの上位に依存関係を持つリソースが存在する場合は、そのリソースは起動されません。スイッチオーバーやフェイルオーバーの切り替えにおいても同様の処理が行われます。この動作の詳細は「**6.スクリプト仕様**」を参照してください。

4.2. 停止処理(remove)

Tomcat リソースを OSU(サービス停止状態)にするために、Tomcat サービスの停止処理を行います。この動作は LifeKeeper の GUI クライアントまたは CUI で Tomcat リソースの停止処理を実行したときが該当します。停止処理に失敗した場合は、Tomcat リソースは OSF(サービス障害状態)になります。Tomcat リソースの下位に依存関係を持つリソースが存在する場合は、そのリソースは停止されません。スイッチオーバーやフェイルオーバーの切り替えにおいても同様の処理が行われます。

LifeKeeper for Linux v7.1 以降はフェイルオーバーの発生時に、リソースの停止に失敗する場合、稼動系ノードの強制再起動し、フェイルオーバーを継続させます。この動作の詳細は「**6.スクリプト仕様**」を参照してください。

4.3. 監視処理(quickCheck)

Tomcat リソースが ISP のときに、Tomcat のサービスの生死判定を行います。監視処理でサービスの障害を検出すると回復処理に状態遷移します。この動作の詳細は「**6.スクリプト仕様**」を参照ください。

4.4. 回復処理(recover)

監視処理で障害を検出した場合に限り実行します。監視処理で障害を検出したノード上で Tomcat サービスの再起動を行います。回復処理に失敗した場合、待機系へのフェイルオーバーを開始します。この動作の詳細は「**6.スクリプト仕様**」を参照してください。フェイルオーバー動作の詳細は「**5.障害時の切り替え動作**」を参照してください。

5. 障害時の切り替え動作

以下の場合にリソースの切り替えの実行を目的に、各スクリプトを作成してあります。Tomcat リソースには起動スクリプト(restore)、停止スクリプト(remove)、監視スクリプト(quickCheck)、回復スクリプト(recover) がそれぞれ指定されるものとします。

5.1. Tomcat リソース障害時

Tomcat リソースの障害を検知した場合は、以下のように動作します。

- (1).Tomcat リソースの quickCheck 処理で障害を検知します。
- (2).Tomcat リソースの recover 処理が実行され、回復処理が行われます。
 - 回復処理に成功した場合、通常のサービス状態に戻ります。
 - 回復処理に失敗した場合、待機系ノードにフェイルオーバーします。

5.2. ノード障害時

コミュニケーションパスが全て DEAD 状態になった場合は、待機系ノードでフェイルオーバーを開始します。このフェイルオーバーでは待機系ノードで各リソースを順次、起動処理を行います。この時の Tomcat リソース起動時の動作は起動処理を行うこととなります。詳細は「**0.起動処理(restore)**」を参照してください。

6. スクリプト仕様

各スクリプトの仕様を以下に説明します。

説明の各所で実行コマンドラインや関連ファイル名 / パスを例示します。

具体的には実際のスクリプトを参照してください。

6.1 監視スクリプト(quickCheck)

Tomcat リソース状況を確認するために、下記監視項目をチェックします。

1 つでも失敗した場合に Tomcat が停止状態であると判定します。それ以外の場合を稼働状態と判定します。

(1).実行ユーザ権限で稼働中の tomcat の java プロセスの存在を確認

```
# pgrep -u tomcat java
```

(2).ポート番号を Listen していることを確認

```
# ss -ln sport = :8080
```

(3).コネクションの確認

```
# curl --connect-timeout 1 --retry 3 --retry-delay 1 --head  
http://127.0.0.1:8080
```

6.2 起動スクリプト(restore)

以下の手順で Tomcat を起動します。

(1).Tomcat の稼働状況の確認

Tomcat の稼働状態を確認します。

詳細は「6.1 監視スクリプト(quickCheck)」を参照してください。

Tomcat が稼働状態であると判定した場合、起動スクリプトを正常終了します。

停止状態だった場合、次の処理に進みます。

(2).ファイルの削除

(1)の確認結果にしたがって、Tomcat が起動状態ではないと判断した場合は、Tomcat のロックファイルと Tomcat の PID ファイルを削除します。

例) ロックファイルと PID ファイル

ロックファイル /var/lock/subsys/tomcat

PID ファイル /var/run/tomcat.pid

(3).Tomcat の起動コマンドを実行します。

```
# /sbin/service tomcat start
```

(4).遅延処理

sleep による 3 秒の間隔をあけます。

(5).Tomcat の稼働状況の確認

Tomcat の稼働状態を再確認します。

再び、上記の (1) を実施します。

Tomcat が稼働状態であると判定した場合、起動スクリプトを正常終了します。

6.3 停止スクリプト(remove)

以下の手順で Tomcat の停止を試行します。

(1).Tomcat の稼働状況の確認

Tomcat の停止状態を確認します。

詳細は「**6.1 監視スクリプト(quickCheck)**」を参照してください。

Tomcat が停止状態であると判定した場合、停止スクリプトを正常終了します。

(2).Tomcat の停止コマンドを実行します。

```
# /sbin/service tomcat stop
```

(3).遅延処理

sleep による 3 秒の間隔をあけます。

(4).Tomcat の稼働状況の再確認

Tomcat の停止状態を確認します。

再び、上記の (1) を実施します。

Tomcat が停止状態であると判定した場合、起動スクリプトを正常終了します。

(5).強制停止処理

(4)の確認結果にしたがって、Tomcat が停止状態ではないと判断した場合、強制停止処理を実行します。

```
#pkill -9 -u <Tomcatの実行ユーザ> <Tomcatのプロセス>
```

(6).ファイルの削除

Tomcat のロックファイルと Tomcat の PID ファイルを削除します。

例) ロックファイルと PID ファイル

ロックファイル /var/lock/subsys/tomcat

PID ファイル /var/run/tomcat.pid

(7).遅延処理

sleep による 3 秒の間隔をあけます。

(6).Tomcat の稼働状況の再確認

Tomcat の停止状態を確認します。

再び、上記の (1) を実施します。

Tomcat が停止状態であると判定した場合、停止スクリプトを正常終了します。

6.4 回復スクリプト(recover)

Tomcat の停止処理および起動処理を実行し、Tomcat のサービスを回復します。回復スクリプトは、内部的に停止スクリプトと起動スクリプトの処理を呼び出すことで、Tomcat の再起動処理を実現します。

(1).Tomcat の稼働状況の確認

Tomcat の稼働状態を確認します。

詳細は「**6.1 監視スクリプト(quickCheck)**」を参照してください。

Tomcat が稼働状態であると判定した場合、起動スクリプトを正常終了します。

(2).停止処理

停止スクリプトの処理を呼び出します。

詳細は「6.3 停止スクリプト(remove)」を参照してください。

(3).強制停止処理

強制停止処理を実行します。

```
#pkill -9 -u <Tomcat の実行ユーザ> <Tomcat のプロセス>
```

(4).起動処理

起動スクリプトの処理を呼び出します。

詳細は「**6.2 起動スクリプト(restore)**」を参照してください。

(5).遅延処理

sleep による 3 秒の間隔をあけます。

(6).Tomcat の稼働状況の再確認

Tomcat の稼働状態を再確認します。

再び、上記の (1) を実施します。

Tomcat が稼働状態であると判定した場合、回復スクリプトを正常終了します。

失敗と判定した場合、待機ノードへフェイルオーバを開始します。

7. スクリプトの実行時間

スクリプトの実行時間およびタイムアウトは下記の通りです。

7.1 restore、remove、recover

スクリプトの実行時間はスクリプト内部で使用されるコマンドや Tomcat の各コマンドの実行処理時間に依存します。導入するシステムの CPU、Memory、I/O 等のシステムリソースの使用状況により終了時間に遅延が発生する場合があります。

これらのスクリプトでは、タイムアウトの初期値が 22 秒で設定されています。スクリプトの処理に 22 秒以上の時間を要した場合は、タイムアウトを意味する以下のメッセージをログに出力し、処理が終了します。

Script was hung. Forcibly terminating.

タイムアウト値を調整するには、次の要領でファイル/etc/default/LifeKeeper に TIMEOUT パラメータを追加してください。

例) Tomcat_TIMEOUT=<秒数>

パラメータ名内の「Tomcat」部分は、リソース階層を生成するときに指定するタグ名です。ただし、英数字以外の記号は“_”に置き換えてください。

それぞれのスクリプトで異なるタイムアウトの設定はできません。TIMEOUT パラメータを設定すると、全てのスクリプトでタイムアウトが設定されますので注意してください。

7.2 quickCheck

quickCheck スクリプトも、restore、remove、recover と同じ TIMEOUT パラメータでタイムアウトが設定できます。ただし、quickCheck のタイムアウトは、設定値より 10 秒小さい値になります。

quickCheck スクリプトのタイムアウトは障害とは見なさないため、タイムアウト後にローカルリカバリやフェイルオーバの動作に移行しません。

8. スクリプトのパラメータ

本 Generic ARK のスクリプトが使用するパラメータの設定やカスタマイズができます。注意点とあわせて、カスタマイズ対象となるパラメータを以下に説明します。

スクリプトのパラメータを設定するために次の場合があります。

1) 必須パラメータを設定する場合

本スクリプトを使用して Tomcat リソースを作成するために、ファイル /etc/default/LifeKeeper にパラメータ「TOMCAT_APP_VERSION」を設定してください。詳細については次項 8.1 を参照してください。

2) Tomcat の利用要件に応じてパラメータを変更・調整する場合

サービスで使用するポート番号や、起動や停止処理のタイムアウト値を調整する場合等、以下のファイルでパラメータの設定や変更ができます。

- A) /etc/default/LifeKeeper
- B) スクリプト:quickCheck、recover、remove、restore

上記のそれぞれを利用する場合や留意点については、次項 7.8.2、7.8.3 を参照してください。

8.1. 必須パラメータの設定

リソースを作成する前に、以下のパラメータを必ず設定してください。

パラメータ	初期値
TOMCAT_APP_VERSION	(値無し)

本 Generic ARK を機能させる基本設定です。保護対象の Tomcat サービス名を定義します。

設定値：	Tomcat のサービス名
設定先ファイル：	/etc/default/LifeKeeper
設定書式例：	TOMCAT_APP_VERSION=tomcat * 「tomcat」 保護対象の tomcat サービス名

注意事項

本パラメータの設定に関連して次の点に注意点してください。

- 1) デフォルト設定では、/etc/default/LifeKeeper にこのパラメータが未設定です。このパラメータが未設定の場合、リソースを作成できず、Tomcat リソースを使用することができません。
- 2) 設定できる tomcat のサービス名の数はひとつです。複数のサービス名を扱うことはできません。
- 3) パラメータの設定は両ノードで行ってください。

8.2. /etc/default/Lifekeeper でパラメータを設定する場合

このファイルでパラメータを設定することで、各スクリプト内で定義してある同名のパラメータの値を一括に変更できます。利用できるパラメータについては「**8.4 カスタマイズ可能なパラメータ**」を参照してください。

注意事項

次の点に注意してください。

- 1) このファイル内で定義 / 設定するパラメータは、quickCheck、recover、remove、restore の各スクリプトに対して一括で有効になります。
- 2) Tomcat スクリプトのパラメータ名(後述)が同ファイル内に定義済みのパラメータと重複する場合(※)にはファイル /etc/default/LifeKeeper でパラメータを変更できません。変更方法については、次の「**0 各スクリプト (quickCheck、recover、remove、restore)でパラメータを設定する場合**」を参照してください。
- 3) 複数の Tomcat リソースを作成し、それぞれのリソースについてそれぞれ異なるパラメータを利用する場合には、次の「**0 各スクリプト (quickCheck、recover、remove、restore)でパラメータを設定する場合**」を参照してください。
- 4) パラメータを設定する際には、両ノードで設定してください。

※補足

パラメータの重複について、可能性のある発生条件として、別の Generic ARK リソースがファイル/etc/default/LifeKeeper 内でパラメータを定義する場合があります。尚、製品として提供する LifeKeeper の Core や ARK が参照するパラメータと重複することはありません。

8.3. 各スクリプト (quickCheck、 recover、 remove、 restore) でパラメータを設定する場合

各スクリプト内でパラメータを設定することで、他の LifeKeeper 関連モジュールに影響を与えると無く、本 Generic ARK 用パラメータ値を定義できます。利用できるパラメータについては「**8.4 カスタマイズ可能なパラメータ**」を参照してください。

注意事項

次の点に注意ください。

- 1) パラメータは基本的に各スクリプト同一の内容で、定義 / 設定してください。
- 2) /etc/default/LifeKeeper にも同一のパラメータが設定されている場合には、そちらが優先します。
- 3) リソース作成後にパラメータを変更する場合には、
/opt/LifeKeeper/subsys/gen/resources/app/actions/以下に取り込まれた各スクリプトを変更する必要があります。
変更の方法は Generic ARK のスクリプト内容を変更する場合と同様です。変更する場合にはリソースを再作成する、あるいは、以下の URL にあるユーザーサイトの情報を参照してください。

[Linux][Windows] Generic ARK で使用中のスクリプトを更新する手順を教えてください。」

<http://lk.sios.com/?p=680>

8.4. カスタマイズ可能なパラメーター一覧

カスタマイズ可能なパラメータは「**8.2 /etc/default/LifeKeeper を使用する場**合」と、「**8.3 各スクリプトでパラメータを設定する場合**」の両方で共通です。

パラメータ	初期値	説明
APP_IPADDR	127.0.0.1	Tomcat サービス用 IP アドレス
APP_PORT	8080	Tomcat サービス用ポート
APP_USER	tomcat	Tomcat の実行ユーザ名
APP_PROGRAM	Java	Tomcat の実行プロセス名
APP_CONNECT_TIMEOUT	1	Tomcat に対する接続時のタイムアウト(秒)
APP_CONNECT_RETRY	3	Tomcat に対する再接続回数
APP_CONNECT_RETRY_DELAY	1	Tomcat に対する再接続間隔(秒)
APP_START_DELAY	3	起動時の遅延処理(秒数)
APP_STOP_DELAY	4	停止時の遅延処理(秒数)

9. 免責事項

- 本書に記載された情報は予告なしに変更、削除される場合があります。最新のものをご確認ください。
- 本書に記載された情報は、全て慎重に作成され、記載されていますが、本書をもって、その妥当性や正確性についていかなる種類の保証もするものではありません。
- 本書に含まれた誤りに起因して、本書の利用者に生じた損害については、サイオステクノロジー株式会社は一切の責任を負うものではありません。
- 第三者による本書の記載事項の変更、削除、ホームページ及び本書等に対する不正なアクセス、その他第三者の行為により本書の利用者に応じた一切の損害について、サイオステクノロジー株式会社は一切の責任を負うものではありません。
- システム障害などの原因によりメールフォームからのお問い合わせが届かず、または延着する場合がありますので、あらかじめご了承ください。お問い合わせの不着及び延着に関し、サイオステクノロジー株式会社は一切の責任を負うものではありません。

【著作権】

本書に記載されているコンテンツ(情報・資料・画像等種類を問わず)に関する知的財産権は、サイオステクノロジー株式会社に帰属します。その全部、一部を問わず、サイオステクノロジー株式会社の許可なく本書を複製、転用、公衆への送信、販売、翻案その他の二次利用をすることはいずれも禁止されます。またコンテンツの改変、削除についても一切認められません。

本書では、製品名、ロゴなど、他社が保有する商標もしくは登録商標を使用しています。

サイオステクノロジー株式会社

住所：〒106-0047

東京都港区南麻布 2 丁目 12-3 サイオスビル

URL：<http://www.sios.com>