



**SIOS Protection Suite for Linux  
Amazon EC2 Cross Region  
v9.2.1**

**クイックスタートガイド**

**2017年 12月**

本書およびその内容は SIOS Technology Corp. (旧称 SteelEye® Technology, Inc.) の所有物であり、許可なく使用および複製は禁止されています。SIOS Technology Corp. は本書の内容に関していかなる保証も行いません。また、事前の通知なく本書を改訂し、本書に記載された製品に変更を加える権利を保有しています。SIOS Technology Corp. は、新しい技術、コンポーネント、およびソフトウェアが利用可能になるのに合わせて製品を改善することを方針としています。そのため、SIOS Technology Corp. は事前の通知なく仕様を変更する権利を保留します。

LifeKeeper、SteelEye、および SteelEye DataKeeper は SIOS Technology Corp. の登録商標です。

本書で使用されるその他のブランド名および製品名は、識別のみを目的として使用されており、各社の商標が含まれています。

出版物の品質を維持するために、弊社は本書の正確性、明瞭性、構成、および価値に関するお客様のご意見を歓迎いたします。

以下の宛先に電子メールを送信してください。

[ip@us.sios.com](mailto:ip@us.sios.com)

Copyright © 2017

By SIOS Technology Corp.

San Mateo, CA U.S.A.

All rights reserved

# 目次

---

|  |           |
|--|-----------|
| <b>Chapter 1: 本資料の目的</b> .....                   | <b>1</b>  |
| <b>Chapter 2: 構成の概要</b> .....                    | <b>2</b>  |
| サービス用 インスタンス .....                               | 4         |
| VPN 用 インスタンス .....                               | 4         |
| 本環境の保護対象 サービスとサービスへの接続 .....                     | 4         |
| <b>Chapter 3: 利用のための必要要件</b> .....               | <b>6</b>  |
| Amazon Web Service (AWS) 環境上の要件 .....            | 6         |
| EC2 インスタンス作成時の要件 .....                           | 6         |
| 全インスタンス共通 .....                                  | 6         |
| VPN 用 インスタンス .....                               | 8         |
| サービス用 インスタンス .....                               | 8         |
| 利用ディストリビューション、ソフトウェアの要件 .....                    | 8         |
| <b>Chapter 4: 構築手順</b> .....                     | <b>10</b> |
| <b>Chapter 5: 関連する LifeKeeper リソースについて</b> ..... | <b>25</b> |
| Openswan リソース .....                              | 25        |
| 動作概要 .....                                       | 25        |
| Openswanリソースの監視とリカバリ動作 .....                     | 25        |
| Openswanリソースのチューニング項目 .....                      | 26        |
| Route53 リソース .....                               | 26        |
| 動作概要 .....                                       | 26        |
| Route53 リソースの監視とリカバリ動作 .....                     | 27        |
| Route53 リソースのチューニング項目 .....                      | 27        |
| EC2 リソース .....                                   | 28        |
| 動作概要 .....                                       | 28        |
| EC2 リソースの監視とリカバリ動作 .....                         | 28        |

---

|   |           |
|---|-----------|
| EC2 リソースの切り替えが発生した際の route table の更新の動作 ..... | 28        |
| IP リソース .....                                 | 29        |
| 動作概要 .....                                    | 29        |
| <b>Chapter 6: 本構成における設定および運用上の留意点 .....</b>   | <b>30</b> |
| Quorum/Witness Server の利用を検討してください .....      | 30        |
| Route53リソース起動にともなうレコードの更新に時間がかかる場合があります ..... | 30        |
| <b>Chapter 7: 既知の問題とトラブルシューティング .....</b>     | <b>31</b> |

## Chapter 1: 本資料の目的

LifeKeeper for Linux v8.3 にて、Amazon EC2 Cross Region 構成がサポートされました。これにより、異なる2つの Region 間 (Cross Region) で HA クラスターを構成し、Amazon EC2 上のローカル IP アドレスを使用して提供されるバックエンドサービスの切り替えが可能になります。本資料は、LifeKeeper for Linux v8.3 で Amazon EC2 Cross Region 構成を利用するための要件や基本操作を解説するものです。

なお、本資料は LifeKeeper や Amazon Web Service (以下 AWS) の基本的な設定や操作、技術的な詳細情報を解説するものではありません。本構成の前提となる LifeKeeper や AWS に関する用語・操作・技術情報等につきましては、関連のマニュアルやユーザーサイト等であらかじめご確認ください。

**注記** : 「Amazon Web Services」、「Powered by Amazon Web Services」のロゴ、「AWS」、「Amazon EC2」、「EC2」、「Amazon Elastic Compute Cloud」、「Amazon Virtual Private Cloud」、「Amazon Route 53」および「Amazon VPC」は、米国その他の国における Amazon.com, Inc. またはその関連会社の商標です。

## Chapter 2: 構成の概要

本構成は、Amazon EC2 環境において、異なる2つのRegion間でHA クラスタを構成し、ローカルIPアドレスを使用して提供されるバックエンドサービスの冗長化構成を取るため利用することができます。この構成を使用することによって、異なるRegionに配置されたインスタンス上で動作するサービス切り替えを実現することができ、より幅広い障害に対する事業継続ソリューションを提供できるようになりました。

具体的な構成例は以下のようなものとなります。

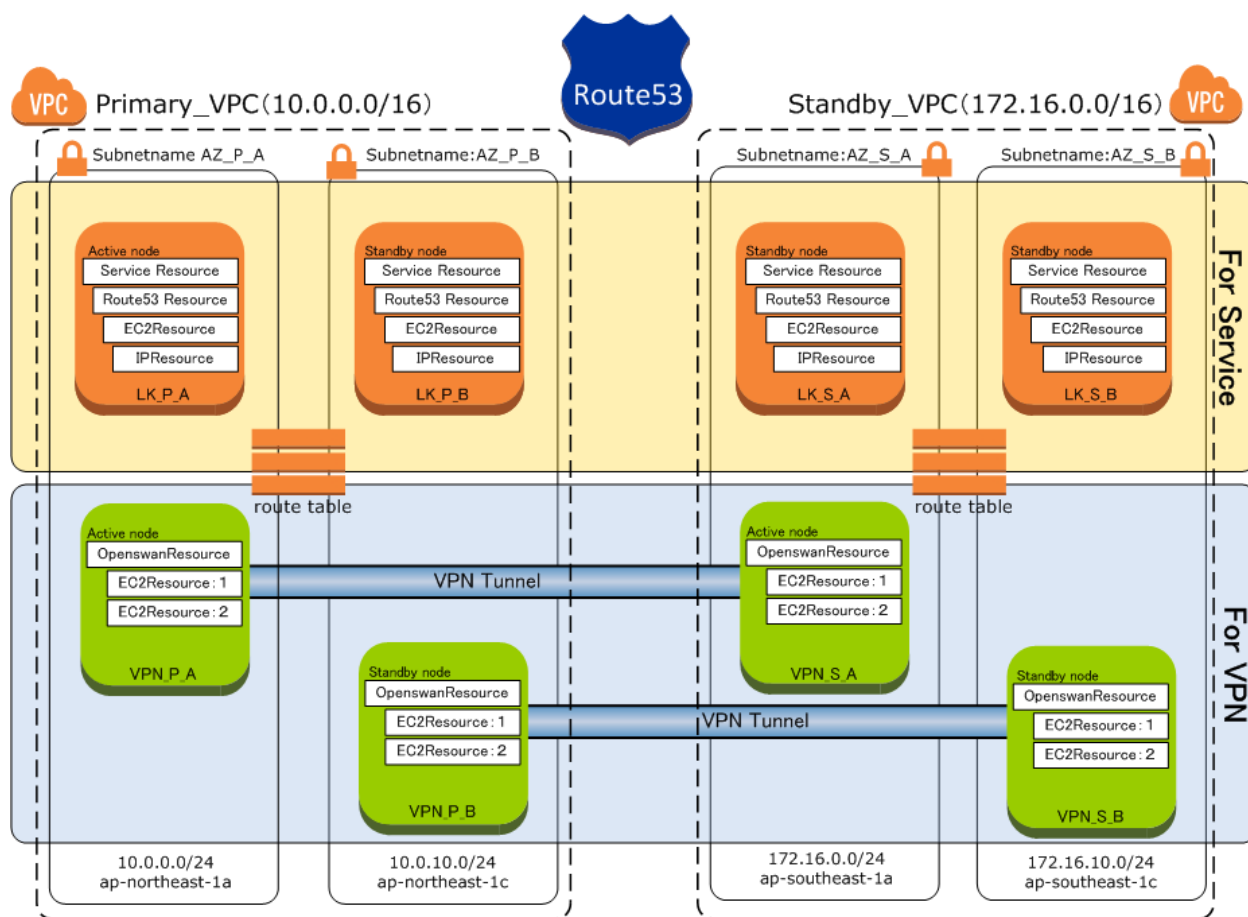


図 1 Amazon EC2 Cross Region 構成例

図 1 で使用している各名称について以下にまとめます。

| 上図にある名称                       | 説明  |
|-------------------------------|---|
| Region                        | 地理的に離れた場所にホスティングされている AWS の領域を表します。このドキュメントでは、優先的にサービスを提供するインスタンスが配置されている Region を「Primary Region」としています。また、Primary Region のフェイルオーバー先となる Region を「Standby Region」としています。  |
| VPC (Virtual Private Cloud)   | 同一 Region 内に作成できる論理的に独立したネットワーク領域を表します。   |
| Region AZ (Availability Zone) | 同一 Region にある独立したロケーションを表します。このドキュメントでは、Primary VPC 内で優先的にサービスを提供するインスタンスが配置されている AZ を「AZ_P_A」、スタンバイとなるインスタンスが配置されている AZ と「AZ_P_B」としています。また、Standby VPC 内で優先的にサービスを提供するインスタンスが配置されている AZ を「AZ_S_A」、スタンバイとなるインスタンスが配置されている AZ を「AZ_S_B」としています。 |
| Route Table                   | VPC 作成時に定義される routing table です。AZ 間での通信や Region 間での通信を行うための経路を定義しています。  |
| Route 53                      | Amazon Route 53 サービスによって提供される DNS です。   |
| Openswan                      | Openswan は、Linux で利用できるオープンソースのカーネルレベルの IPsec の実装です。詳細につきましては、 <a href="#">Openswan の公式サイト</a> などをご確認ください。  |
| サービス用インスタンス                   | 本構成における保護対象サービスがインストールされ、HA クラスタが構成される EC2 インスタンスです。このドキュメントではサービス用インスタンスと呼ぶこととします。   |
| VPN 用インスタンス                   | 異なる Region 上にある VPC との接続を可能にするために、Openswan を使用した VPN を構築するための EC2 インスタンスです。この Openswan をインストールして使用するサーバを、このドキュメントでは VPN 用インスタンスと呼ぶこととします。   |
| Route53 リソース                  | Route53 Recovery Kit (以下 Route53 RK) を使用して作成される LifeKeeper リソースです。Region 間でのサービス切り替えが発生した場合に、クライアントからの一意な URL でアクセスを可能にするために仮想 IP アドレスとそれに関連した DNS レコードの更新を行います。  |
| Openswan リソース                 | Openswan Recovery Kit(以下 Openswan RK) を使用して作成される LifeKeeper のリソースです。Openswan のプロセス起動状態や IPsec-VPN セッション状態を監視します。  |
| EC2 リソース                      | Openswan RK でリソース作成する際、また、Route53 リソース作成前に作成する LifeKeeper リソースです。AZ 間、または Region 間で切り替えが行われた場合、通信が継続できるよう関連する Route table の書き換えを行います。   |
| IP リソース                       | IP Recovery Kit (以下 IP RK) を使用して作成される LifeKeeper リソースです。切り替え可能な仮想 IP アドレスを作成しそれを保護します。  |

**注記:** Region 間の通信においては Internet Gateway(IGW)を経由しますが、図の記述は省略されています。

**注記:** これら AWS に関連する用語やその詳細につきましては、AWS のドキュメントや LifeKeeper のオンラインマニュアル等をご確認ください。また、本構成のために使用する LifeKeeper の関連リソースの詳細については別章 [関連する LifeKeeper リソースについて](#)に記載しますので、そちらをご参照ください。

次に、本構成で使用している各 LifeKeeper リソースの役割の概要を以下に述べます。

## サービス用 インスタンス

EC2 Cross Region 構成では、図 1 のようにサービス提供用 HA クラスタを構成するノードとして Primary Region 側の VPC に 2 ノード、Standby Region 側の VPC に 2 ノードの計 4 ノードを使用します。図 1 では Primary Region の LK\_P\_A インスタンスがプライマリノード(通常時サービスを優先的に提供するノード)、残りの LK\_P\_B、Standby Region 側の LK\_S\_A、LK\_S\_B は全てバックアップノードとなります。この時のリソースのフェイルオーバー時の動作としては、同 Region の AZ 間でフェイルオーバーが行われる場合には、自動的にフェイルオーバーが行われます。しかし、異なる Region へのフェイルオーバーについては、自動的に行わず管理者の判断によって手動で行われるように LifeKeeper の動作をチューニングして使用します。

なお、現時点での EC2 Cross Region 構成で保護できるサービスは Oracle12c と PostgreSQL 8.3、8.4、9.0、9.1、9.2、9.3、9.5 (他の Edition についてはサポート外) となります。Oracle または PostgreSQL の保護には LifeKeeper のオプションである Oracle Recovery Kit または PostgreSQL Recovery Kit を使用します。また Oracle Recovery Kit または PostgreSQL Recovery Kit を使用する場合、クラスターノード間で切り替え可能な共有ファイルシステムが必要となります。本構成では、共有ファイルシステムとして「DataKeeper for Linux」を使用します。

## VPN 用 インスタンス

Region を挟んで LifeKeeper で HA クラスタを構成するためには、Region 間での通信が可能である必要があります。そのため本構成では、それぞれのサーバが配置されている同一 AZ 内に Openswan を使用した VPN サーバを使用します。図 1 では Primary Region の VPN\_P\_A、VPN\_P\_B と、Standby Region の VPN\_S\_A、VPN\_S\_B がこれにあたります。これらの VPN 用インスタンスは、対向 Region のプライマリノード同士とバックアップノード同士で VPN Tunnel を張っています。対向 Region への接続は、どちらか一方の VPN Tunnel を経由して行われます。この時 VPN 接続の冗長性を確保するため、LifeKeeper for Linux v8.3 リリースとともに開発された Openswan Recovery Kit を使用します。

## 本環境の保護対象サービスとサービスへの接続

サービス提供用インスタンスでは、保護するアプリケーションやサービスを起動し、LifeKeeper で保護をして使用することになります。そのため、HA クラスタ間での切り替えが行われた場合でも一意の接続先となるアドレスを確保するため LifeKeeper では IP リソースを使用します。なお、本構成で利用できる IP アドレスは、パブリック IP アドレスではなく、プライベート IP アドレスである点に注意してください。

IP リソースは、ノード間での切り替えができる仮想 IP アドレスの作成と保護を行います。本構成での IP リソースは、Region ごとに異なる仮想 IP アドレスのリソースを作成することができます。よって、異なる Region の切り替え先で IP リソースを起動した場合、その仮想 IP アドレスは切り替え前と異なる仮想 IP アドレスとなります。そのため、リソースが異なる Region へ切り替えられた時、クライアントが同一の接続先を指定してアプリケーションへのアクセスを継続できるように考慮する必要があります。本構成では、この動作を LifeKeeper for Linux v8.3 リリースとともに開発された Route53 Recovery Kit(Route53リソース)が担います。

Route53 Recovery Kit は、Region 間でのサービスノードの切り替えが発生した場合、サービスが起動する Region のネットワークセグメントの IP アドレスに対応するよう、Amazon Route 53 に対して DNS レコードの更新要求を行います。クライアントが保護対象に接続する際には、DNS に問い合わせることによって IP アドレスが変化しても、それを意識することなく単一の名前を使用して接続することができます。

ここまでの切り替え時の動作に加えて、サービス提供用インスタンスや VPN 用インスタンス上で動作しているリソースの切り替えが行われた場合には、同時に AZ 間や Region 間での通信経路を更新する必要があります。この経路の更新を、各リソースと依存関係が作成される EC2 リソースが行っています。

このように、EC2 Cross Region 構成では、AZ 間や Region 間での切り替えが発生した場合の、経路の切り替えや、名前解決、Region 間接続の制御と保護を組み合わせることによって、Region 間でのサービスの冗長化を可能にしています。

## Chapter 3: 利用のための必要要件

LifeKeeper for Linux v9.2.1 Amazon EC2 Cross Region Support を利用するためには、環境を準備する段階で満たすべきいくつかの要件があります。以下に Amazon Web Service の環境と、その上に作成するインスタンスに関する要件をまとめます。

### Amazon Web Service (AWS) 環境上の要件

サービスを提供するための基盤となる環境を AWS 上に作成します。Amazon EC2 Cross Region を利用するための要件は以下の通りです。必要な設定等については、構築手順に記載しておりますので、そちらをご確認ください。

- 異なる Region それぞれに VPC を 1 つ計 2 つの VPC を設定する必要があります。2 つ以上の VPC 間の HA クラスタ構成についてはサポートしていません。なお、1 つの VPC 内にある Availability Zone (AZ) 間での HA クラスタ構成については、これまで提供しておりました「Recovery Kit for EC2」をご利用いただくことができます。
- 各 Region の AZ に 2 つのサブネットが必要です。よって、全体では 4 つの subnet が必要となります。同 Region 内にあるクラスターノードの冗長性を担保するために、プライマリノードとバックアップノードはそれぞれ異なる AZ 上に配置するようにします。
- AWS の管理者権限が必要です。また、自分の AWS アクセスキー ID と秘密アクセスキーも取得する必要があります。
- 各 AZ に LK インスタンス 1 つと VPN 用インスタンス 1 つの計 2 つの EC2 インスタンスが必要となります。よって、全体では 8 台の EC2 インスタンスが必要です。
- Amazon Route 53 にドメイン名を登録しサービスを利用できるようにする必要があります。これは Route53 リソース作成時に必要となります。
- サービスを利用するクライアントは、Route53 リソースで保護されるホスト名を名前解決できることを確認する必要があります。

### EC2 インスタンス作成時の要件

本ソリューションを利用するため、計 8 つの EC2 インスタンスを作成することになります。その際、各用途に応じたインストール要件等を考慮してインスタンスを作成する必要があります。その作成時の要件のポイントは以下の通りです。実際に構築にあたっての必要な設定項目等については、構築手順に記載していますのでそちらをご確認ください。

#### 全インスタンス共通

これは、VPN 用、サービス用全インスタンスに共通する要件となります。こちらの内容をまず確認した上で、それぞれの用途別の要件を確認してください。

- 全てのインスタンスは LifeKeeper for Linux のインストール要件を満たす必要があります。

本構成のために用意する EC2 インスタンス上には LifeKeeper をインストールする必要があります。

す。そのため、作成するインスタンスは、OS や NIC の数など LifeKeeper のインストール要件を満たす必要があります。本構成で利用できるディストリビューションや保護対象アプリケーションの情報につきましては、[利用ディストリビューション、ソフトウェアの要件](#)をご確認ください。

LifeKeeper を利用する上で考慮すべき OS の設定、ネットワーク、ディスク構成等の情報につきましては以下のインストレーションガイドのプランニング環境や環境のセットアップの内容をご確認ください。

[SIOS Protection Suite インストレーションガイド](#) (ネットワークやOSの設定等)

また、サービス用インスタンスに保護対象サービスをインストールする場合、クラスタ間で共有するファイルシステムとして DataKeeper for Linux を使用します。DataKeeper の利用に関する要件についても、合わせて確認してください。

[DataKeeper for Linux 構成上の要件](#)

**注記** : Lifekeeper を利用する要件の中に、通信のために使用する必要があるポートがいくつかありますので、インスタンスの設定の **[NETWORK & SECURITY]** にある **[Security Groups]** のルールを設定を確認するようにしてください。

- 同 Region 内で HA クラスタを構成するインスタンスは、プライマリ用インスタンスとスタンバイ用インスタンスがそれぞれ異なる AZ で起動するように構成する必要があります。
- Amazon EC2 API Tools をあらかじめ任意の場所にインストールする必要があります。

これは関連の LifeKeeper リソースが動作するために必要となります。Amazon EC2 API Tools は以下の URL よりダウンロードしてください。また、インストール方法につきましては、同 API のマニュアルをご確認ください。

[Amazon EC2 API Tools](#)

- 各 HA クラスタ用インスタンスの ENI の、[Change Source/Dest Check] の設定を <Disabled> に設定を変更する必要があります。

設定変更画面は、以下の EC2 インスタンスの管理画面の **[Actions]** ボタンの **[Change Source/Dest Check]** を選択すると起動することができます。

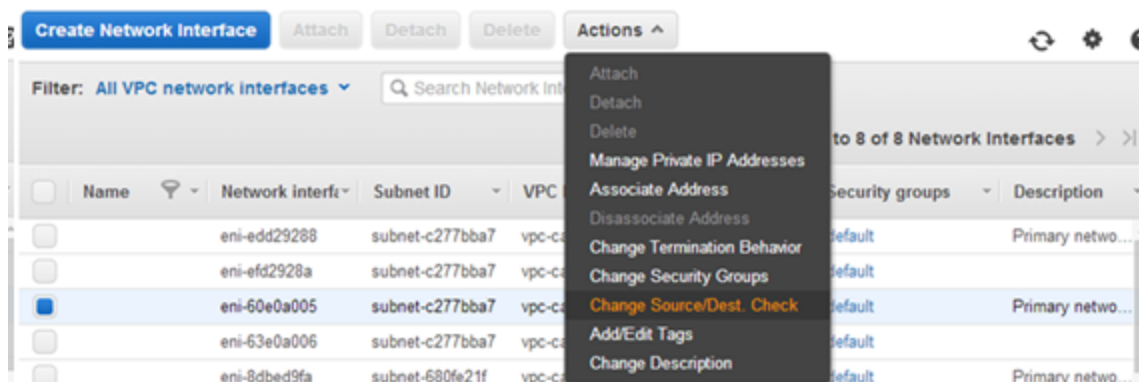


図 1 Change Source/Dest Check

## VPN 用 インスタンス

- VPN 用 インスタンスに Openswan をインストールする必要があります。そのため、LifeKeeper のインストール要件に加えて Openswan のインストール要件を満たす必要があります。

Openswan をインストールし、VPN セッションを張る必要があります。そのため、LifeKeeper をインストールする要件に加えて Openswan を利用するための要件を満たしている必要があります。

## サービス用 インスタンス

- 切り替え対象ソフトウェアとして Oracle を使用する場合は、LifeKeeper のインストール要件に加えて Oracle のインストール要件を満たす必要があります。

サービス用インスタンスには、Oracle 12c をインストールして保護対象サービスとします。よって、サービス用インスタンスは Oracle のインストール要件を満たす必要があります。Oracle のインストール要件につきましては、Oracle 社が提供する情報をご確認ください。

- 保護対象ソフトウェアが使用する共有ファイルシステム専用のローカルディスク、またはパーティション領域を1つ以上作成してください。

保護対象アプリケーションに Oracle を使用する場合、LifeKeeper のオプション製品である「Oracle Recovery Kit(Oracle RK)」を使用して Oracle リソースを作成します。

Oracle リソースを作成する要件として、データベース、アーカイブファイル、ログファイル、制御ファイル等は LifeKeeper によって保護する共有ファイルシステム上へ配置する必要があります。その共有ファイルシステムは、LifeKeeper for Linux のオプション製品である DataKeeper を使用してレプリケーションされるディスクを使用します。よって、各サービス用インスタンスには必ず、Oracle 専用のレプリケーション用のディスク、またはパーティション領域を1つ以上作成しておく必要があります。

用意するディスク構成やサイズ等については、データベースの規模や構成に依存します。Oracle のファイル配置の要件については以下の Oracle ARK の設定要件の関連ファイルの配置例を確認して、お客様の要件に適したものを選択しディスク構成やそのサイズを決定してください。

### [Oracle 特有の設定上の考慮事項](#)

#### [設定例](#)

## 利用ディストリビューション、ソフトウェアの要件

本バージョンでは、以下のソフトウェアの利用をサポートします。

| カテゴリ                 | OSおよびソフトウェア名   |
|----------------------|--|
| 各インスタンス用 ディストリビューション | Red Hat Enterprise Linux v6.4/6.5/7.0 の 64bit<br>CentOS v6.4/6.5/7.0 の 64bit |
| VPNインスタンス保護対象ソフトウェア  | 上記ディストリビューション付属の Openswan rpm または Libreswan rpm                              |

| カテゴリ                  | OSおよびソフトウェア名   |
|-----------------------|--|
| サービス用インスタンス保護対象ソフトウェア | Oracle 12c<br>PostgreSQL 8.3、8.4、9.0、9.1、9.2、9.3、9.5 (他の Edition についてはサポート外) |

これらの保護対象ソフトウェアや対応ディストリビューションは、随時適用範囲を広げていきます。VPNインスタンス保護対象ソフトウェアにLibreswan rpmをご使用の場合は、本資料の「Openswan」を「Libreswan」に読み替えてください。



図1の設定例では、Primary RegionにPrimary\_VPC、Standby RegionにStandby\_VPCという名前のVPCを作成しています。また、それぞれのCIDR BlockはPrimary\_VPCが10.0.0.0/16、Standby\_VPCが172.16.0.0/16としています。

## 2. VPCに2つのsubnetを作成し、AZに配置します

手順1で作成したそれぞれのRegionのVPCに2つのsubnetを作成してください。その際、指定するAZがsubnetごとに別々になるように設定してください。subnetの作成や関する情報はAmazon Web Serviceの情報をご確認ください。図1の例では、Primary RegionのPrimary\_VPCに10.0.0.0/24 (AZはap-northeast-1a)と10.0.10.0/24 (AZはap-notheast-1c)を作成しています。また、Standby RegionのStandby\_VPCには172.16.0.0/24 (AZはap-southeast-1a)と172.16.10.0/24 (AZはap-southeast-1c)を作成しています。

## 3. VPN用インスタンス1つ、サービス用インスタンスを1つ(計8つ)を作成します

各インスタンスの必要要件をもとにEC2インスタンスを作成します。インスタンスの必要要件については、前述の[インスタンス作成時の要件](#)のページにある情報を参照してください。

## 4. Amazon Route 53のhosted zoneにDomain nameを定義してください

後に作成するRoute53リソースはこのhosted zoneに仮想IPアドレスと関連付ける仮想ホスト名をAレコードとして追記や、切り替えに応じたレコードの更新などを行います。

## 5. VPCのRoute Tablesにルーティング情報を追加します

VPC管理画面のRouting tableの画面を開いてください。画面下部の**[Routes]**というタブをひらくと、Routing tableが表示されます。下記はPrimary\_VPCのデフォルトでのRouting tableの例です。

| Destination | Target    | Status | Propagated |
|-------------|-----------|--------|------------|
| 10.0.0.0/16 | Local     | Active | No         |
| 0.0.0.0/0   | igw-xxxxx | Active | No         |

このデフォルトのRoute tableに、以下の条件に基づき、Routingの情報を設定する必要があります。

- このVPCの領域で起動する仮想IPアドレス
- 対向側で起動する仮想IPアドレスが属するsubnet
- 対向RegionにあるVPCのsubnet

具体的には以下のように設定を変更します。以下の例はPrimary\_VPCのRoute table例です。

### Primary\_VPC Route table

| Destination | Target |
|-------------|--------|
| 10.0.0.0/16 | local  |

| Destination   | Target  |
|---------------|---|
| 0.0.0.0/0     | igw-xxxxxxx   |
| 10.1.0.10/32  | eni-zzzzzzzz サービス用のインスタンスの仮想IPを付ける予定のENI (構成例ではLK_P_AのENI)  |
| 10.2.0.0/16   | eni-yyyyyyyy VPNインスタンスのプライマリノードのENI (構成例ではVPN_P_AのENI)      |
| 172.16.0.0/16 | eni-yyyyyyyy 上記と同じVPNインスタンスのプライマリノードのENI (構成例ではVPN_P_AのENI) |

上記の赤文字で書かれた部分が、追加で設定すべき情報です。必要な設定を追加して保存してください。また、この Route table の設定は対向 Region でも必要となります。対向 Region の設定例は以下のとおりです。

#### Standby \_VPC Route table

| Destination   | Target  |
|---------------|---|
| 172.16.0.0/16 | local   |
| 0.0.0.0/0     | igw-xxxxxxx   |
| 10.2.0.10/32  | eni-aaaaaaaa サービス用のインスタンスの仮想IPを付ける予定のENI (構成例ではLK_S_AのENI)  |
| 10.1.0.0/16   | eni-bbbbbbbb VPNインスタンスのプライマリノードのENI (構成例ではVPN_S_AのENI)      |
| 10.0.0.0/16   | eni-bbbbbbbb 上記と同じVPNインスタンスのプライマリノードのENI (構成例ではVPN_S_AのENI) |

#### 6. VPN 用インスタンスに Openswan をインストールして、対向 Region にある VPN 用インスタンスと VPN セッションを張ります

Openswan リソースを設定する前に、Openswan がインストールされ VPN セッションが確立されている必要があります。

Openswanは使用するディストリビューションに付属するrpmパッケージを使用します。通常のインストールコマンド ( rpm または yum ) でパッケージを追加してください。VPN用インスタンス全てに Openswanをインストールしたら、必要な設定を行います。

Openswan の設定ファイルである ipsec.conf の設定例を以下に記します。

```

version 2.0
config setup
    protostack=netkey
    nat_traversal=yes
    virtual_private=
    oe=off
conn vpn1
    type=tunnel
    authby=secret
    left=%defaultroute
    leftid=54.249.2.50
    #VPN_P_A のパブリック IP アドレス
    leftnexthop=%defaultroute
    leftsubnets={10.0.0.0/16,10.1.0.0/16}
    #Primary_VPC の subnet と仮想 IP アドレスの subnet
    leftsourceip=10.0.0.5
    #VPN_P_A のローカルアドレス
    right=54.254.156.254
    #VPN_S_A のパブリック IP アドレス
    rightsubnets={172.16.0.0/16,10.2.0.0/16}
    #Standby_VPC の subnet と仮想 IP アドレスの subnet
    rightsourceip=172.16.0.5
    pfs=yes
    auto=start

```

上記設定ファイルの、left,leftid,leftsubnets,leftsourceip および right,rightid,rightsubnets,rightsourceip に設定された値を Openswan リソースの処理の中で参照しています。そのため、Openswan リソースを作成するための設定要件として以下のポイントに留意する必要があります。

- left,leftid のどちらかに VPN サーバのパブリック IP アドレスを記述しなければなりません。
- leftsubnets には VPN サーバのローカルSubnet を記述しなければなりません。

- leftsourceip には VPN サーバのローカル IP アドレスを記述しなければなりません。right 側も同様です。
- left,right のどちらにローカルサイド、リモートサイドを記述しても問題ありません。

これらの設定を使用してVPN\_P\_AとVPN\_S\_A、VPN\_P\_BとVPN\_S\_B間でVPNセッションを張って通信できることを確認してください。ここで起動したVPNセッションを張った状態のままにしてください。リソース作成時は、2本のVPN接続が可能な状態で作成する必要があります。

## 7. 各 EC2 インスタンスに LifeKeeper をインストールする

インストール手順については、LifeKeeper オンラインマニュアルのインストールガイドをご確認ください。

### [SIOS Protection Suite インストレーションガイド](#)

上記の基本的な内容に加え、本構成における留意点を以下に記載します。

- Openswan ARK、Route53 ARK、EC2 ARK は別途 RPM コマンドでインストールする必要があります。

インストール時に使用した CD イメージのマウントパスの直下にある *Amazon* ディレクトリの中にあります。rpmコマンドでインストールすることを忘れないようにしてください。

- LifeKeeper GUI を表示するため SSH X フォワーディングを使用してください。

LifeKeeper の起動が完了すると、LifeKeeper GUI を起動します。EC2 環境で SSH X フォワーディングを使用して管理用端末上に LifeKeeper GUI を表示させるようにします。X フォワーディングの利用法は、一般的な情報をご参照いただくか、X フォワーディングを使用した LifeKeeper GUI の表示方法は以下の URL に関連情報がありますので、こちらを参照してください。

### [ファイアウォール経由での LifeKeeper GUI の実行](#)

## 8. 各インスタンスの LifeKeeper に必要なチューニング設定を行う

本構成を利用するために必要な LifeKeeper のチューニング項目をあらかじめ設定します。設定項目は VPN 用インスタンスとサービス用インスタンスで異なっていますので、それぞれの環境別に確認するようにしてください。

- 全 VPN 用インスタンスに必要な設定
  - `/etc/default/LifeKeeper` ファイルにある LKCHECKINTERVAL の設定値をデフォルト 120 から 260 秒に変更してください。設定を反映するために LifeKeeper を再起動してください。
- 全 サービス用インスタンスに必要な設定
  - `/etc/default/LifeKeeper` ファイルにある LKCHECKINTERVAL の設定値をデフォルト 120 から 360 秒に変更してください。設定を反映するために LifeKeeper を再起動してください。

- IP リソースのブロードキャスト Ping を使用した監視を無効にするようにするため、`/etc/default/LifeKeeper` ファイルの `NOBCASTPING` の設定をデフォルト値の 0 から 1 に設定変更してください。
- `/etc/default/LifeKeeper` ファイルの `CONFIRMSODEF` の設定をデフォルトの 0 から 1 に変更して保存してください。
- Region 間の自動フェイルオーバーが行われないようにするためサーバープロパティの[Set Confirm Failover On]を設定する必要があります。設定例を以下の表に示します。(※ホスト名は図1設定例の構成に対応しており、各ノード上でサーバープロパティを開いて設定した例を表していません。)

## LK\_P\_A

| ホスト名   | Set Confirm Failover On | Set Block Resource Failover On |
|--------|-------------------------|--------------------------------|
| LK_P_A | (チェックなし)                | (チェックなし)                       |
| LK_P_B | (チェックなし)                | (チェックなし)                       |
| LK_S_A | ✓                       | (チェックなし)                       |
| LK_S_B | ✓                       | (チェックなし)                       |

## LK\_P\_B

| ホスト名   | Set Confirm Failover On | Set Block Resource Failover On |
|--------|-------------------------|--------------------------------|
| LK_P_B | (チェックなし)                | (チェックなし)                       |
| LK_P_A | (チェックなし)                | (チェックなし)                       |
| LK_S_A | ✓                       | (チェックなし)                       |
| LK_S_B | ✓                       | (チェックなし)                       |

## LK\_S\_A

| ホスト名   | Set Confirm Failover On | Set Block Resource Failover On |
|--------|-------------------------|--------------------------------|
| LK_S_A | (チェックなし)                | (チェックなし)                       |
| LK_P_A | ✓                       | (チェックなし)                       |
| LK_P_B | ✓                       | (チェックなし)                       |
| LK_S_B | (チェックなし)                | (チェックなし)                       |

## LK\_S\_B

| ホスト名   | Set Confirm Failover On | Set Block Resource Failover On |
|--------|-------------------------|--------------------------------|
| LK_S_B | (チェックなし)                | (チェックなし)                       |

| ホスト名   | Set Confirm Failover On | Set Block Resource Failover On |
|--------|-------------------------|--------------------------------|
| LK_S_A | (チェックなし)                | (チェックなし)                       |
| LK_P_A | ✓                       | (チェックなし)                       |
| LK_P_B | ✓                       | (チェックなし)                       |

GUIで設定した内容をコマンドで確認する場合、`flg_list`コマンドを使用できます。例のように設定した後、各ノード`flg_list`コマンドを実行した時、以下のように`confirmso!`フラグが作成されていることを確認できます。

| ホスト名   | Confirmso! フラグ                       |
|--------|--------------------------------------|
| LK_P_A | confirmso!LK_S_A<br>confirmso!LK_S_B |
| LK_P_B | confirmso!LK_S_A<br>confirmso!LK_S_B |
| LK_S_A | confirmso!LK_P_A<br>confirmso!LK_P_B |
| LK_S_B | confirmso!LK_P_A<br>confirmso!LK_P_B |

#### 9. VPN用 インスタンスにコミュニケーションパスを設定します

VPN 用インスタンスは、同 Region 内にあるインスタンス同士で HA クラスタを構成します。異なる Region を挟んでコミュニケーションパスを設定する必要はありませんので、この点に注意してください。具体的には、VPN\_P\_A インスタンスとVPN\_P\_B インスタンス間でコミュニケーションパスを設定し、VPN\_S\_A インスタンスとVPN\_S\_B インスタンス間でコミュニケーションパスを設定します。

コミュニケーションパスの設定手順は、以下のマニュアルで確認してください。

##### [コミュニケーションパスの作成](#)

#### 10. 各 Region のVPN インスタンスに Openswan リソースを作成および拡張します

プライマリノードとなるサーバの LifeKeeper GUI を起動して、リソース作成ウィザードを起動してください。

リソース作成ウィザードが起動したら、ウィザードに従って値を入力してください。画面を進めることによってリソースを作成することができます。ウィザードの入力項目と値は以下の通りです。

| 項目 | 設定内容 |
|----|------|
| 作成 |      |

| 項目                    | 設定内容   |
|-----------------------|--|
| Select Recovery Kits  | 保護対象となるアプリケーションを保護するために使用する Recovery Kit を選択します。   |
| Switchback Type       | 自動フェイルバックの有無   |
| Server                | リソースのプライマリサーバーとなるノードを選択します。例では PrimaryRegion にある VPN__P_A を選択しています。  |
| IPSec connection name | IPSec connection name を選択します。値は Openswan の設定ファイルから取得しています。   |
| EC2 Home              | EC2_HOME ディレクトリパスを選択するか、入力してください。この EC2_HOME は、EC2 API Tools のパスです。注記: デフォルト値は /opt/aws です。検証する対象の <\$EC2_HOME>/bin/ec2-describe-addresses が存在するかどうかを確認してください。 |
| EC2 URL               | 実際の EC2_URL を選択するか、入力してください。この EC2_URL は、Amazon EC2 Web サービスエンドポイントの URL です。   |
| AWS Access Key        | AWS のアクセスキーを入力してください。この AWS アクセスキーは、AWS Tools がユーザの識別に使用するアクセスキー ID です。アクセスキーとセキュリティキーの組み合わせを使用して、AWS サービス API に対する REST またはクエリプロトコルリクエストがセキュリティで保護されます。       |
| AWS Secret Key        | AWS セキュリティキーを入力してください。この AWS セキュリティキーは秘密キーです。アクセスキーとセキュリティキーの組み合わせを使用して、AWS サービス API に対する REST またはクエリプロトコルリクエストがセキュリティで保護されます。                                 |
| Openswan Tag          | リソースタグ名  |
| <b>拡張</b>             |  |
| Target Server         | 拡張先のサーバ名を指定  |
| Switchback Type       | 自動フェイルバックの有無   |
| Template Priority     | リソースの拡張元となるノードのプライオリティ値  |
| Target Priority       | リソースの拡張先、バックアップノードのプライオリティ値  |
| Openswan Tag          | リソースタグ名  |

作成から拡張までリソース作成ウィザードが完了すると、以下のようなリソース階層が作成されます。

Openswan リソースの子リソースとして EC2 リソースが自動的に作成されます。

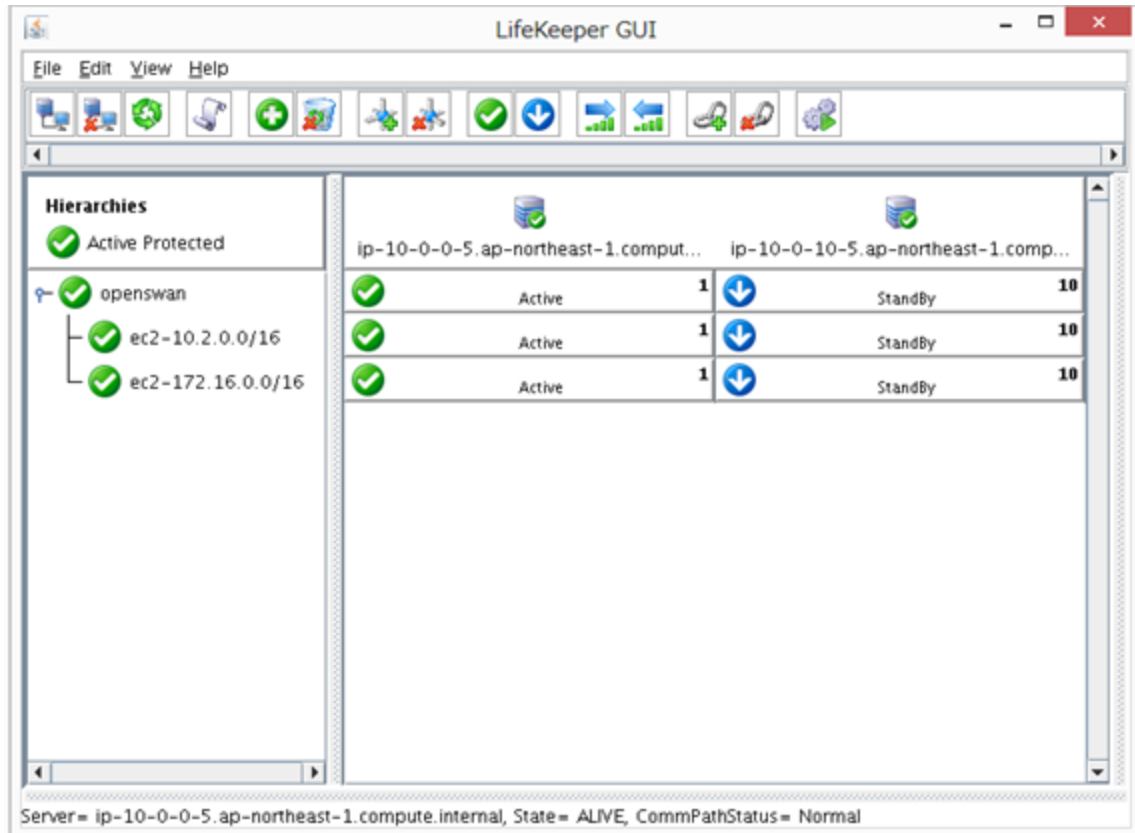


図 2 Openswan リソース階層

11. 10.の操作をもう一方の Region にある VPN 用 インスタンスでも実施し、同様の VPN リソースを作成してください。

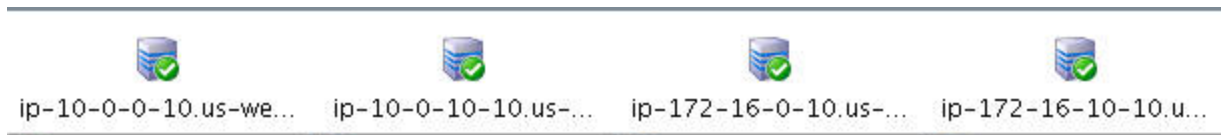
12. VPN インスタンスのスイッチオーバーテストを実施します。

作成した VPN リソースを、LifeKeeper GUI を使用して手動で切り替えられることを確認してください。確認後リソースをプライマリノードへ戻してから次の手順を実施してください。

13. 全 Region のサービス用インスタンスにコミュニケーションパスを設定します

サービス用インスタンスは Region を挟んで 4 つ作成されています。これら全てについてコミュニケーションパスを設定し 1 つのクラスターグループとして構成します。コミュニケーションパスを設定する操作方法は手順 9. と同様です。この操作を 4 ノード全てが接続されるようにしてください。結果

として、LifeKeeper GUI には以下のように4ノードが表示される状態となります。



#### 14. サービス用インスタンスに仮想 IP アドレスを作成し、全 AZ にある各インスタンスに拡張します

サービス用インスタンスのプライマリノードとなるサーバの LifeKeeper GUI を表示し、IP リソースの作成を行ってください。IP リソースの作成に関する基本的な操作方法につきましては、以下の IP Recovery Kit のオンラインマニュアルを確認してください。

##### [IP リソース階層の作成](#)

##### [リソース階層の拡張](#)

上記の IP リソース作成時の基本操作に加えて、以下の本構成上での注意点を合わせて確認し作成を行ってください。

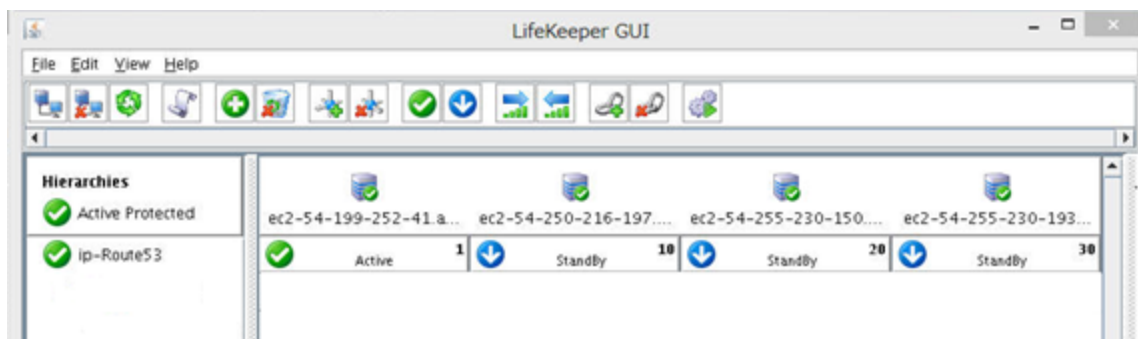
- 指定する仮想 IP アドレスは VPC 内の割り当て済みの CIDR の範囲外にする必要があります。
- リソース作成時 IP タグ名はデフォルトから変更するようにしてください。

本構成の IP リソースは 1 つの IP リソースで 2 つの仮想 IP アドレスを管理しています。よって、管理上の混乱を避けるためデフォルトのタグ名から「ip-route53」などのタグ名に変更してください。

- 異なる Region のインスタンスへ拡張を行う時、IP アドレスはそのセグメントで利用できる仮想 IP アドレスを指定します。

通常 IP リソースを作成する場合には、拡張元の仮想 IP アドレスをもとに拡張を行いますが、異なる Region に拡張する際のウィザードでは任意の IP アドレスを入力できるようになっていますので、拡張先のネットワークセグメントに合わせた仮想 IP アドレスを指定するようにしてください。

IP リソースが作成された時点の LifeKeeper GUI の表示は以下ようになります。



### 図 3 IP リソース作成時点の例

#### 15. サービス用インスタンスに EC2 リソースを作成します

サービス用インスタンスの EC2 リソースは、あらかじめ Recovery Kit for EC2 のバックエンドシナリオをもとに別途作成しておく必要があります。

作成ウィザードの内容等につきましては、以下の Recovery Kit EC2 のマニュアルを確認してください。

[リソース階層の作成](#)

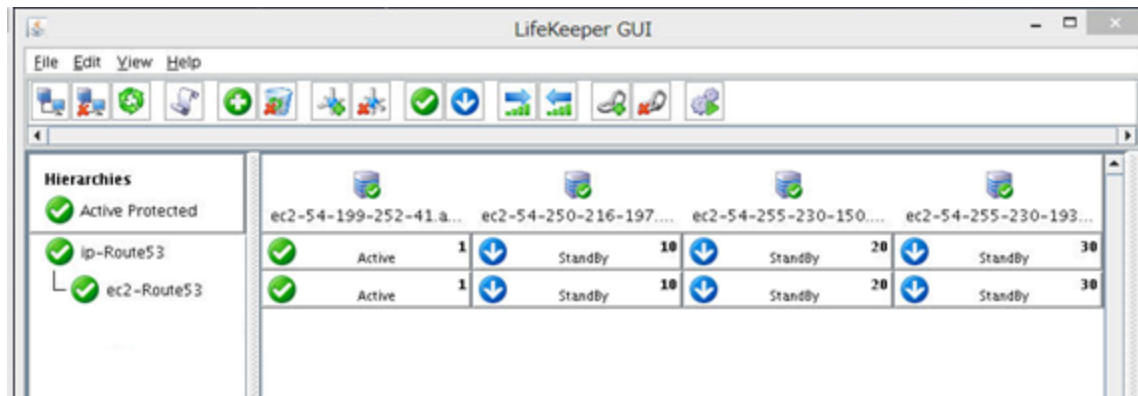
[リソース階層の拡張](#)

上記の EC2 リソース作成時の基本操作に加えて、以下の本構成上での注意点を合わせて確認し作成を行ってください。

- リソース作成ウィザードの EC2 Resource type で「ルートテーブルシナリオ」を選択してください。
- リソース作成時 EC2 リソースタグ名はデフォルトから変更するようにしてください。

例えば、IP リソースを「ip-route53」とした場合は、「ec2-route53」などとします。これも、IP リソース作成時と同様にデフォルトのタグ名をそのまま使用すると管理上混乱を来す可能性が考えられるためです。

EC2 リソースが作成された時点の GUI 表示は以下ようになります。



### 図 4 EC2 リソース作成時点のリソース階層例

#### 16. サービス用インスタンスに Route53 リソースを作成します

プライマリノードとなるサーバの LifeKeeper GUI を起動して、リソース作成ウィザードを起動してください。リソース作成ウィザードが起動したら、ウィザードに従って値を入力し画面を進めることによってリソースを作成することができます。ウィザードの入力項目は以下の通りです。

| 項目                                | 設定内容   |
|-----------------------------------|--|
| 作成                                |  |
| Select Recovery Kits              | 保護対象となるアプリケーションを保護するために使用する Recovery Kit を選択します。   |
| Switchback Type                   | 自動フェイルバックの有無   |
| Server                            | リソースのプライマリサーバーとなるノードを選択します。例では Primary Region にある VPN_P_A を選択しています。  |
| AWS Access Key                    | Route 53 に接続するためのアクセスキーを入力   |
| AWS Secret Key                    | Route 53 に接続するためのシークレットキーを入力   |
| Domain name (Route53 hosted zone) | Route 53 に登録されている Domain Name が表示されますので、このサービスで使用するものを選択します。 <b>注記</b> ：この時点で Route 53 にドメインネームが作成されている必要があります。これは前述の AWS 環境作成上の要件に記載されています。 |
| Host Name (Not FQDN)              | Route 53 の A レコードで使用するホスト名を入力します。  |
| IP resource                       | DNS A レコードに登録する IP アドレスを、リストから選択します。リストに表示される IP アドレスは、IP リカバリキットにより保護されている仮想 IP アドレスです。   |
| Route53 Resource Tag              | リソースの名前  |
| 拡張                                |  |
| Target Server                     | 拡張先のサーバ名を指定  |
| Switchback Type                   | 自動フェイルバックの有無   |
| Template Priority                 | リソースの拡張元となるノードのプライオリティ値  |
| Target Priority                   | リソースの拡張先、バックアップノードのプライオリティ値  |
| Route53 Tag                       | リソースタグ名  |

リソース拡張は4ノード目まで行ってください。

作成から拡張までリソース作成ウィザードが完成すると、以下のようなリソース階層が完成します。

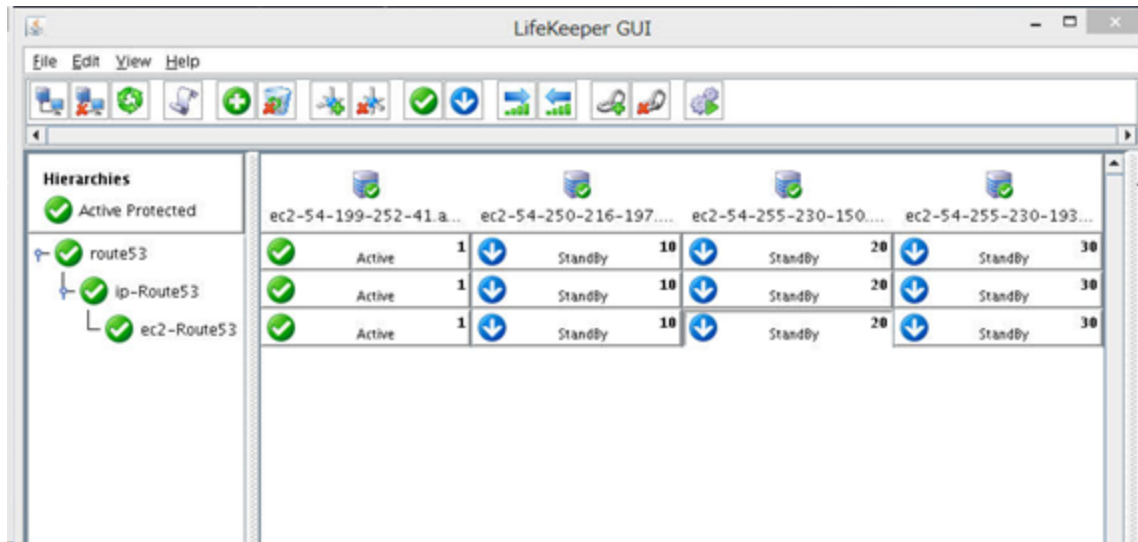


図 5 Route53 リソース完成時のリソース階層

17. Oracle ARK のインストール要件に従って Oracle のインストールを行い、Oracle リソースが作成できるよう設定を行ってください

Oracle Recovery Kit のインストール要件につきましては、以下の Oracle Recovery Kit 管理ガイドで確認してください。

[Oracle Recovery Kit 管理ガイド](#)

Oracle は4ノードへのインストールが必要となります。それぞれ同一の設定となるようにしてください。また、その際 DataKeeper のミラーリング対象となるディスクへの配置をするようにしてください。

また、リスナーを保護する場合には、Listen アドレスに Route53 リソースで設定したホスト名を指定してください。

設定が完了したらプライマリとなるサーバ以外の Oracle インスタンスは停止させ、レプリケーション対象ディスクをアンマウントしてください。

18. レプリケーション対象とするディスクを使用して DataKeeper リソースを作成します

サービス用ノード間で共有するデータ領域を Oracle とともに切り替えて使用できるように、DataKeeper リソースを作成します。DataKeeper リソースの基本的な作成方法につきましては、以下のマニュアルを確認してください。

[DataKeeper リソース階層の作成](#)

[リソース階層の拡張](#)

今回の手順上、DataKeeper リソースを作成する際に、リソース作成ウィザードの中の **[Hierarchy Type]** の選択の際に **[Replicate Existing File System]** を使用してください。これは、既にレプリケーション対象のディスクに Oracle 関連のファイルが保存されているためです。よって、DataKeeper リソースを作成する前に、プライマリノードで対象ディスクをマウントポイントにマウントしてからリソース作成の操作を行ってください。

## 19. Oracle リソースを作成します

Oracle Recovery Kit を使用してリソース設定を行います。リソース作成ウィザードの内容につきましては以下のオンラインマニュアルの内容を確認してください。

[リソース階層の作成](#)[リソース階層の拡張](#)

リソース作成時全てのサービス用インスタンスへ拡張してください。

ここまでの時点でサービス用インスタンスの LifeKeeper GUI のリソースは以下のようになっています。

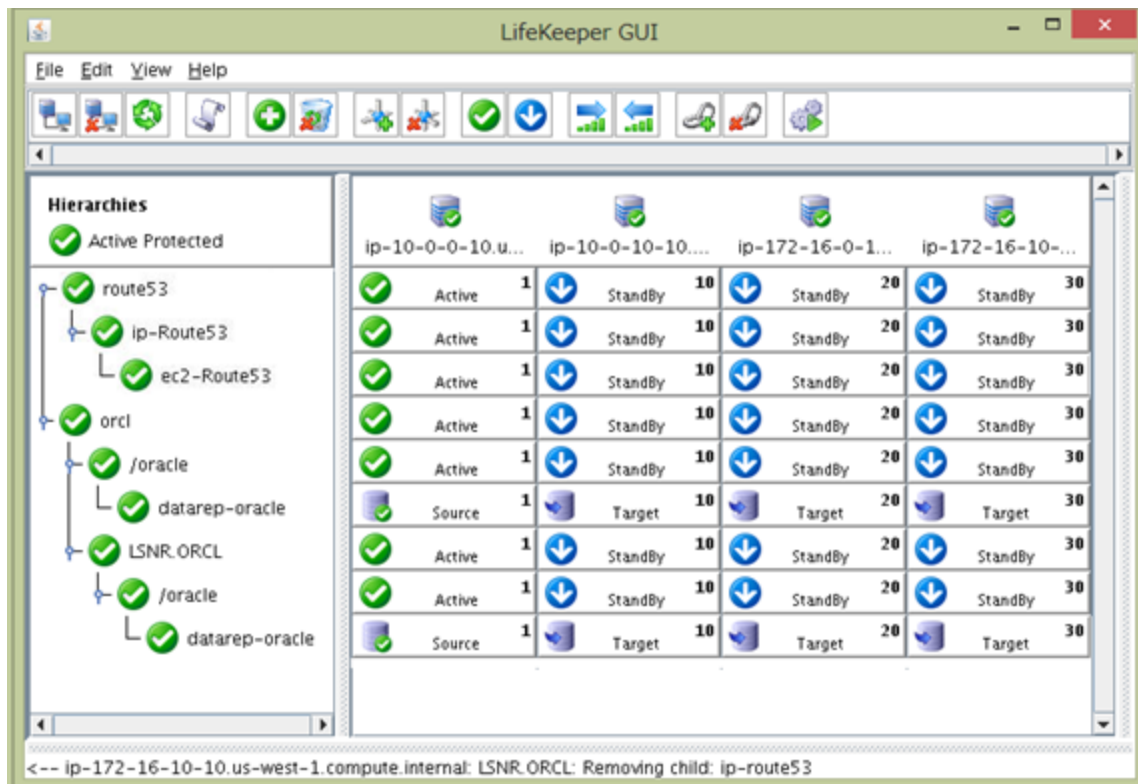


図 6 Oracleリソース作成時点

## 20. Route53 リソース階層と Oracle リソース階層の依存関係を作成します

Oracle のリソース階層より先に Route53 リソース階層が先に起動するように依存関係を手動で構成する必要があります。例えば以下の図のような階層を作成します。

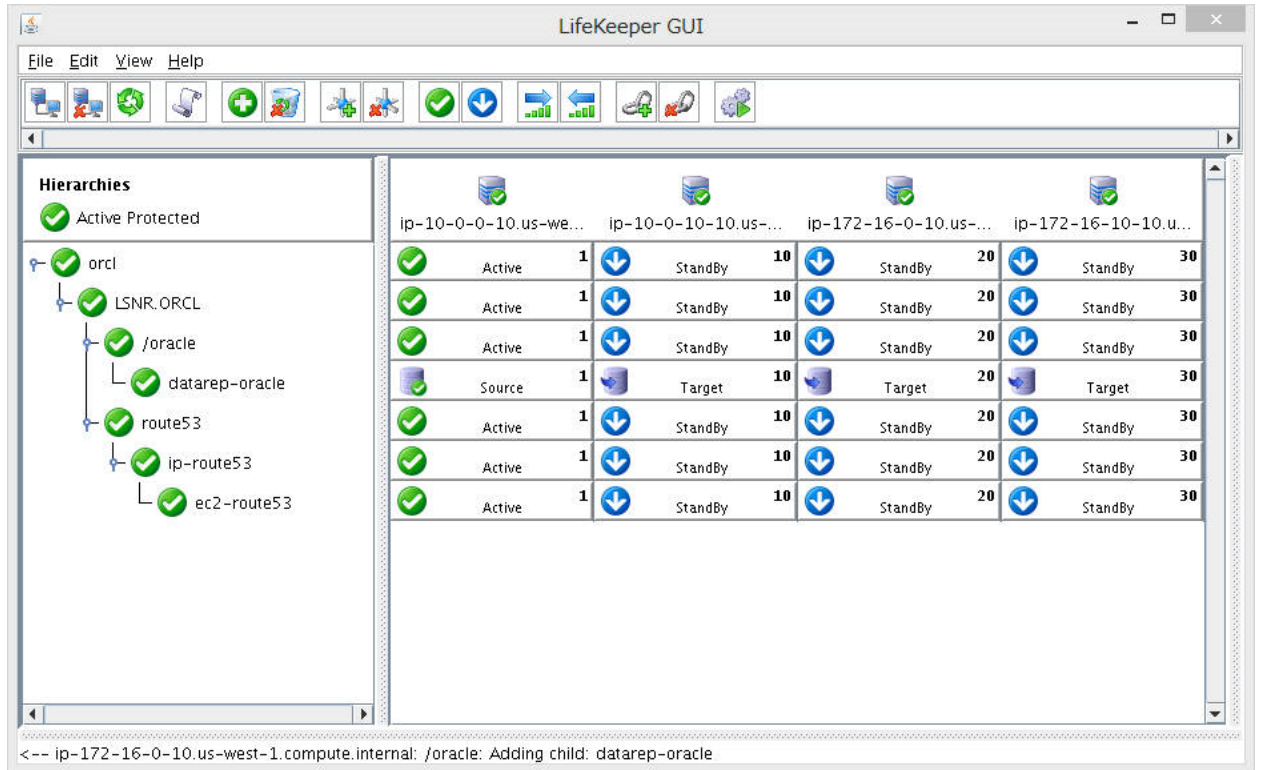


図 7 Route53リソース階層とOracleリソース階層の依存関係例

この例では LSNR.ORCL リソースの子リソースに Route53 リソース階層を設定しています。

このように Oracle 関連のリソースよりも先に Route53 リソース階層が起動されるように依存関係を構成してください。

手順は以上です。

導入作業の最後に、各サービスを提供する上で必要な動作確認等を実施してください。切り替えなどを行う場合には、DataKeeper リソースのステータスが Source と Target になっていることを確認してください。

## Chapter 5: 関連する LifeKeeper リソースについて

構成の概要で触れたとおり、本構成には 4 つの LifeKeeper リソースが利用されています。

それぞれの機能と動作概要は以下の通りです。

[Openswan リソース](#)

[Route53 リソース](#)

[EC2 リソース](#)

[IP リソース](#)

### Openswan リソース

#### 動作概要

Region 間の接続を行うために必要な VPN 接続の監視を行うリソースです。

Openswan リソースは、アクティブ・スタンバイ両方の VPN トンネルの接続性について監視を行っています。対向 Region への接続には Openswan リソースのステータスが Active となっている VPN 用インスタンス側の VPN トンネルを経由します。この時の通信経路を決定しているのが、Route Table です。Openswan リソースの切り替えなどに応じて、必要な Route table の更新を EC2 リソースが行っています。

この時の、EC2 リソースの動作については別途 EC2 リソースの項目を参照してください。

#### Openswan リソースの監視とリカバリー動作

Openswan リソースは、システムにインストールされた Openswan を使用して、構築された対向 Region にあるインスタンスとの間に構成された 1 つの IPsec-VPN セッションを保護することができます。リソースの監視内容として、以下のような処理をアクティブノードとスタンバイノードの双方で実行しています。

1. Openswan daemon のプロセス状態をチェックします。(このチェックで NG だった場合以後のチェックを行わずその時点で障害と判定します。)
2. VPN トンネルを経由して対向 VPN server の local IP に ping を送信し、応答があるかチェックします。(最大応答時間はチューニング可能)
3. 対向 VPN server の public IP に ping を送信し、応答があるかチェックします。
4. ipsec コマンドを使用して、VPN トンネルが正常に張られているかチェックします。

上記いずれかのチェックで NG と判定された場合、リソース障害と判定しリソースのリカバリー動作 (ローカルリカバリーやフェイルオーバー)を行います。

アクティブノードまたはスタンバイノードのどちらか、あるいは両方において障害と判定した場合には、ローカルリカバリーを行います。リカバリーを実行した結果からフェイルオーバーを行う条件は以下となります。

- アクティブノード、スタンバイノードが共にリカバリーに成功した場合は、ローカルリカバリーを成功としてフェイルオーバーは行わない
- アクティブノードでリカバリーが成功して、スタンバイノードで失敗した場合はローカルリカバリー成功としてフェイルオーバーは行わない
- アクティブノードがリカバリーに失敗して、スタンバイノードで成功した場合はローカルリカバリーに失敗したもとして、リソースのフェイルオーバーを行う。
- アクティブノード スタンバイノードの両方でローカルリカバリーに失敗した場合には、ローカルリカバリーの処理としては成功した結果を返してフェイルオーバーを行わないようにします。そして、以後両ノード、またはどちらか一方のVPNトンネルが復旧できるまで監視とリカバリーが行われます。

ほぼ全てのLifeKeeper リソースが、上述のローカルリカバリーに失敗した場合にはリソースをバックアップノードへフェイルオーバーしますが、Openswan リソースは、アクティブノードとスタンバイノードの両方でVPNトンネルによる通信経路の保護を行い、フェイルオーバーによって通信を継続できる場合のみフェイルオーバーを行います。そのため、前述のようなフェイルオーバーの発生条件となっています。

## Openswanリソースのチューニング項目

状況に合わせて以下のチューニングを行うことができます。チューニングを行う場合には、両ノードの `/etc/default/LifeKeeper` ファイルに以下のパラメータを追記します。変更後保存すると、即時その値が有効となります。LifeKeeper や OS の再起動は必要ありません。

| パラメータ名                   | デフォルト値 | 説明                             |
|--------------------------|--------|--------------------------------|
| OPENSWAN_REMOVE_TIMEOUT  | 15 (秒) | remove 処理のタイムアウト時間             |
| OPENSWAN_PING_TIMEOUT    | 5 (秒)  | ping タイムアウト                    |
| OPENSWAN_CHECK_TRY_COUNT | 3 (回)  | プロセス起動後のVPNトンネル疎通確認の最大 TRY 回数。 |
| OPENSWAN_CHECK_INTERVAL  | 15 (秒) | プロセス起動後のVPNトンネル疎通確認のインターバル     |

## Route53 リソース

### 動作概要

リージョン間のスイッチオーバーが生じた後、サービスへの接続を継続して確保するためには、新しいリージョンで定義した仮想 IP アドレス情報に対応する Amazon Route 53 DNS 情報の更新が必要となります。この機能は、Route53 リソースで提供されています。スイッチオーバーが生じた場合、Route53 リソースは、新しいリージョ

ンの仮想 IP アドレスに対応する DNS A レコードを生成し、更新を行うために Amazon Route 53 サービスに信号を送信します。

## Route53 リソースの監視とリカバリ動作

Route53 リソースは作成時に登録した DNS A レコードの取得と仮想 IP アドレスとの関連づけの正常性を監視しています。リソースの監視内容として、以下のような処理を行っています。

1. Route 53 の A レコードで設定されているアドレスを API で取得します。取得に失敗した場合は、デフォルトで 2 秒の間隔を置いてから、情報の再取得を行います。3 回取得を試行して取得できなかった場合、ログに記録を残し監視処理を終了します。
2. Route53 リソースと依存関係がある IP リソースから IP アドレスを取得して、Route53 の A レコードで設定されているアドレスと IP リソースの IP アドレスを比較します。一致した場合、正常であると判断して処理を正常終了します。一致しなかった場合、異常であると判断しリカバリを開始します。

## Route53 リソースのチューニング項目

状況に合わせて以下のチューニングを行うことができます。チューニングを行う場合には、両ノードの `/etc/default/LifeKeeper` ファイルに以下のパラメータを追記します。変更後保存すると、即時その値が有効となります。LifeKeeper や OS の再起動は必要ありません。

| パラメータ名                        | デフォルト値 | 説明   |
|-------------------------------|--------|--|
| ROUTE53_TTL                   | 10 (秒) | TTL ( Time To Live ) (秒) の設定値。<br>*設定を反映させるためにはスイッチオーバーさせること |
| ROUTE53_RECORD_INTERVAL       | 2 (秒)  | A レコード更新時の Route 53 API 通信のインターバル                            |
| ROUTE53_RECORD_TRY_COUNT      | 3 (回)  | A レコード更新時の Route 53 API 通信のトライ回数                             |
| ROUTE53_CHANGEID_INTERVAL     | 20 (秒) | Route 53 API 通信のステータス確認時のインターバル                              |
| ROUTE53_CHANGEID_TRY_COUNT    | 5 (回)  | Route 53 API 通信のステータス確認時のトライ回数                               |
| ROUTE53_RECORDCHECK_INTERVAL  | 2 (秒)  | Route 53 API 通信による A レコード情報取得に要する時間                          |
| ROUTE53_RECORDCHECK_TRY_COUNT | 4 (回)  | A レコードの情報取得時の Route 53 API 通信のトライ回数                          |

## EC2 リソース

### 動作概要

Openswan リソースと Route 53 リソースの子リソースとして作成されるリソースです。

このリソースは各インスタンス上で動作するリソースが、AZ または Region を挟んで切り替えが行われた場合に、通信を継続するために必要となるルーティング情報の更新を行います。この基本的な動作は、以前より提供している「Recovery Kit for EC2」の「ルートテーブルシナリオ」と同等となります。詳細につきましては、[Recovery Kit for EC2 管理ガイド](#)の情報をご確認ください。

### EC2 リソースの監視とリカバリー動作

Amazon EC2 API Tools を使用し、ルートテーブル内にある保護されている IP ルートのターゲットがアクティブサーバの ENI に正しく設定されていることを確認します。正しいことが確認されない場合は EC2 のローカルリカバリープロセスを実行します。

### EC2 リソースの切り替えが発生した際の route table の更新の動作

Openswan リソースとともに作成される EC2 リソースは、対向 Region への通信をする際の出口となる ENI を更新し、Route53 とともに作成される EC2 リソースは AZ 間で仮想 IP アドレスが切り替えられた場合に、関連付ける ENI の更新を行います。

リソースの切り替えが発生した場合の route table の遷移例は以下の通りです。

- 最初に図 3 のサービス用インスタンスの LK\_P\_A で Route53 リソースが Active、VPN 用インスタンスの VPN\_P\_A で Openswan リソースが Active であるとしてします。その場合、Route table は以下のような状態となります。

(注記：VPC と IGW の設定につきましては、デフォルト値が設定されており、また制御対象ではないため省略します。)

| Destination   | Target           | Description                  |
|---------------|------------------|------------------------------|
| 10.1.0.10/32  | LK_P_AのENI       | Route53リソースがLK_P_A上でActive   |
| 10.2.0.0/16   | VPN_P_AのENI      | OpenswanリソースがVPN_P_A上でActive |
| 172.17.0.0/24 | 上記と同じVPN_P_AのENI | OpenswanリソースがVPN_P_A上でActive |

- この状態から、Openswan リソースが VPN\_P\_A から VPN\_P\_B へ切り替えられた場合、Route table は以下ようになります。

| Destination  | Target     | Description |
|--------------|------------|-------------|
| 10.1.0.10/32 | LK_P_AのENI | 1の状態から変化なし  |

| Destination   | Target      | Description                            |
|---------------|-------------|--|
| 10.2.0.0/16   | VPN_P_BのENI | スイッチオーバーのターゲットであるのVPN_P_BのENIへ情報が更新される |
| 172.17.0.0/24 | VPN_P_BのENI | スイッチオーバーのターゲットであるVPN_P_BのENIへ情報が更新される  |

3. さらにRoute53リソースがLK\_P\_A から LK\_P\_B へ切り替えられた場合、Route tableは以下ようになります。

| Destination   | Target      | Description                                      |
|---------------|-------------|--|
| 10.1.0.10/32  | LK_P_BのENI  | スイッチオーバーのターゲットであるのLK_P_B上で仮想IPを関連付けるENIに情報が更新される |
| 10.2.0.0/16   | VPN_P_BのENI | 2の状態から変化なし                                       |
| 172.17.0.0/24 | VPN_P_BのENI | 2の状態から変化なし                                       |

このように切り替えに応じて通信のTargetとなる情報をEC2リソースが更新を行っています。

## IP リソース

### 動作概要

IPリソースとは、LifeKeeper Core製品に含まれるIPリカバリキットを使用して生成した仮想IPアドレスです。すべてのLifeKeeperリソースと同様に、IPリソースインスタンスは、クラスタ内のすべてのノード間で切り替えることが可能です。IPリソースインスタンスをEC2 Cross Region構成で使用する場合、作成中に指定した仮想IPアドレスは、リソースが他のリージョン内のノードに切り替えられると有効になりません。したがって、このような構成において、他のリージョン内のノードへのリソース拡張が生じた場合、LifeKeeperでは、該当するネットワークセグメント内に存在する適切な仮想IPアドレスを指定することが可能です。

さらに詳細な情報につきましては、[IP Recovery Kit テクニカルドキュメンテーション](#)をご覧ください

## Chapter 6: 本構成における設定および運用上の留意点

### Quorum/Witness Server の利用を検討してください

EC2 Cross Region を使用する環境では、Region を挟んで HA クラスタを構成します。

Region 間のコミュニケーションパスは、同一 Region 内のコミュニケーションパスと比較すると、途切れやすくなることが想定されます。

そのため EC2 Cross Region 環境では、コミュニケーションパスの通信が全て途切れてスプリットブレイン状態に陥る可能性が、一般的な HA クラスタ構成よりも高くなります。

これを踏まえて、本構成ではより安全に運用できるように、LifeKeeper の I/O フェンシング機能の一つである Quorum/Witness Server の利用をご検討ください。

特に、Quorum モードの TCP\_REMOTE 設定を利用すれば、別途 Quorum サーバを立てずに I/O フェンシング機能を実装できるため、クラウド環境においては利用しやすいと考えられます。Quorum/Witness の利用については以下の URL をご確認ください。

[Quorum/Witness](#)

### Route53 リソース起動にともなうレコードの更新に時間がかかる場合があります

Route53 の DNS レコードの更新の伝播速度に関して、Amazon では以下のような情報を提供しています。

Amazon Route 53 よくある質問

Q: Amazon Route 53 上の DNS 設定の変更はどのくらい速く世界中に広まりますか?

<http://aws.amazon.com/jp/route53/faqs/>

Route53 リソースでは Route53 API を使用して DNS へのレコード更新の状況について確認しています。INSYNC ステータスを得られた場合には更新が完了したと判断し、PENDING ステータスが帰ってきた場合には、更新状況の確認のリトライを行います。このような仕様のため、Route53 リソースの起動処理としては正しくレコード更新がかけられているにも関わらず、DNS への更新情報の伝播に時間がかかった場合には、Route53 リソースとしては起動失敗の状態になってしまう場合があります。

もし、Route53 リソースの起動に失敗した場合には、Route53 の管理コンソールをみて A レコードが正しく更新されていることを確認してください。更新されている場合、該当する DNS サービスの更新は完了しており、DNS サービスの更新を反映するには、LifeKeeper のみ更新が必要となります。もう一度 LifeKeeper GUI から Route53 リソースを起動してください。

常時前述のようなケースで Route53 リソースの起動ステータスが失敗する場合には、`/etc/default/LifeKeeper` ファイルの「ROUTE53\_CHANGEID\_TRY\_COUNT」の値をデフォルトの 4 回から 1 回から 2 回、回数を増やす設定して、状況が改善するかご確認ください。この設定変更には LifeKeeper や OS の再起動は必要ありません。

## Chapter 7: 既知の問題とトラブルシューティング

LifeKeeper for Linux v9.2.1 リリース時点での情報はありません。