

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure Step by step Guide

初版



目次

1.	はじめに	5
2.	本ドキュメントについて	5
2.1.	対象読者について	5
2.2.	サイオステクノロジーについて	5
2.3.	サイオステクノロジーへのお問い合わせ	5
2.4.	サポートへのお問い合わせ	6
2.5.	製品に関する情報	6
3.	構成環境	7
3.1.	ネットワーク構成図	7
3.2.	リソース構成図	7
3.3.	既知の問題の確認	8
3.4.	本ガイドを進めるにあたり、必要なソフトウェアイメージ	8
4.	Azure 上の環境構築	9
4.1.	リソースグループの作成	9
4.2.	仮想ネットワークの作成	11
4.3.	可用性セットの作成	17
4.4.	仮想マシンの作成	19
4.5.	インターナルロードバランサーの作成	41
5.	OS の設定	49
5.1.	仮想マシンへのログイン	49
5.2.	sshd_config の編集	51
5.3.	NIC アドレスの固定化	53
5.4.	SELinux の無効化	55
5.5.	ファイアウォールの無効化	55
5.6.	名前解決	55
6.	ローカルリポジトリの設定	56
6.1.	/home の拡張	56
6.2.	OS イメージの転送	56
6.3.	fstab の編集	57
6.4.	ローカルリポジトリの作成	58
6.5.	rh-cloud-base.repo の無効化	58
6.6.	ローカルリポジトリの確認	58
6.7.	GUI 設定	59
7.	LifeKeeper のインストール	60
7.1.	LifeKeeper インストールイメージの転送	60
7.2.	root パスワードの変更	60
7.3.	Witness ノードへの LifeKeeper インストール	61
8.	コミュニケーションパスの作成	69
8.1.	GUI を使用したコミュニケーションパスの作成	69
8.2.	GUI クライアントのステータス情報	73

9.	各種リソースの作成	74
9.1.	リソースの作成前に	75
9.2.	DataKeeper リソースの作成	77
9.3.	PostgreSQL リソースの作成	83
9.4.	LB Health Check リソースの作成	88
9.5.	IP リソースの作成	91
9.6.	階層関係の定義	93
10.	Quorum / Witness Kit の導入	98
11.	お問い合わせ前に	103
11.1.	製品サポートへお問い合わせいただく際に収集すべき情報について	103
11.2.	よく利用する LifeKeeper のコマンド	104
12.	お問い合わせ	105
13.	免責事項	106

改訂履歴

版	更新日	変更情報
初版	2023/06/12	新規作成

1. はじめに

本ドキュメントに含まれる情報は、公表の日付におけるサイオステクノロジー株式会社の考え方に基づいています。サイオステクノロジー株式会社は記載されている内容をお約束しているわけではありません。また、それらの内容を保証するものでもありません。本ドキュメントは情報提供のみを目的としています。また、記載内容は予告無く変更する場合があります。予めご了承ください。

2. 本ドキュメントについて

本ドキュメントは、Azure 上で LifeKeeper for Linux - Quorum Witness Kit majority モードでの構築のガイドになります。LifeKeeper の使い方や、運用方法に関する情報を提供するものではありません。LifeKeeper の使い方に関する詳しい情報は、ユーザーポータルやサイオステクノロジーの Web サイト内のドキュメントをご参照ください。

2.1. 対象読者について

本ドキュメントは、Azure 上で LifeKeeper for Linux Quorum/Witness Kit の構築を計画している方が対象になります。そのため、LifeKeeper の利用や Azure の操作について基本的な知識を持っている技術者を対象としています。

2.2. サイオステクノロジーについて

サイオステクノロジーは、1997 年の創業以来、オープンソースソフトウェアを軸に、Web アプリケーションや OS、IT、システムの開発/基盤構築/運用サポート等の事業を展開し、現在はこれらにクラウド技術を加え、新たな価値創造とそのご提供に取り組んでおります。サイオステクノロジーに関する詳細については、<https://sios.jp/>をご参照ください。

2.3. サイオステクノロジーへのお問い合わせ

サイオステクノロジー株式会社
〒106-0047 東京都港区南麻布 2 丁目 12-3 サイオスビル
bc-sa@sios.com

日本国内および海外の事業所の情報に関しては、弊社の Web サイトをご参照ください。

2.4. サポートへのお問い合わせ

お問い合わせの一次窓口が弊社ではない場合があります。お問い合わせの際はサポート証書よりサポート窓口をご確認ください。サポート窓口が弊社になっている場合は、下記の Web サイトよりお問い合わせください。

<https://bccs.sios.jp/contact/>

2.5. 製品に関する情報

製品ドキュメントに関する情報は、下記のリンクよりご参照ください。製品のリリースノートや Recovery Kit の管理ガイドがあります。

https://support.us.sios.com/asp/jpdocs_us_sios_com_home/

Recovery Kit の動作概要、製品の Errata 情報、ライセンスの取得方法などに関する情報は、ユーザーポータルでご確認いただけます。

<https://lkdkuserportal.sios.jp/hc/ja/>

3. 構成環境

今回 Azure 上で構築するクラスター環境についてです。

本ガイドでは Red Hat Enterprise Linux 9.0 を使用し LifeKeeper for Linux v9.7.0 を用いて HA クラスター環境を構成します。

各 LifeKeeper と、サポート対象 OS につきましては、テクニカルドキュメンテーションより、使用する LifeKeeper のサポートマトリックスをご確認ください。

3.1. ネットワーク構成図

本ガイドで構成した仮想ネットワークは図に示す通りになります。

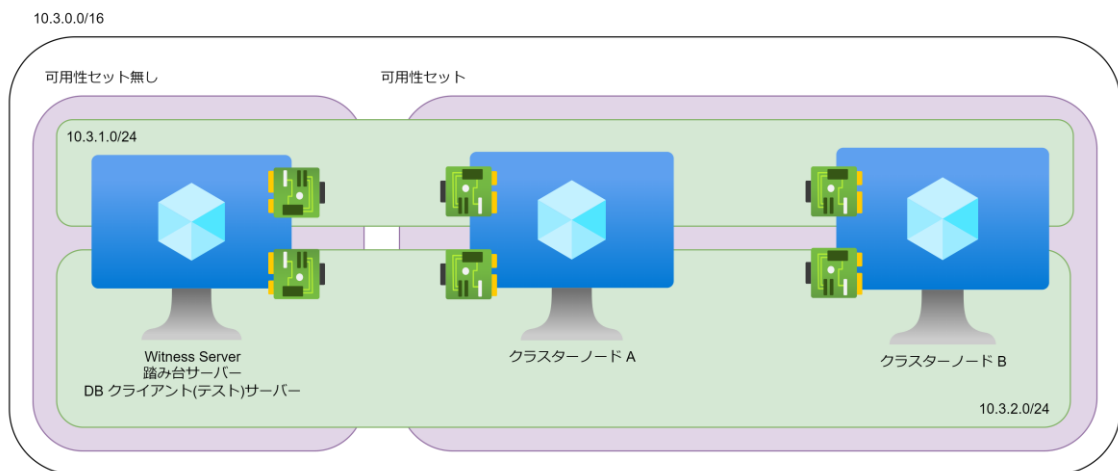


図 3.1-1 ネットワーク構成図

3.2. リソース構成図

本ガイドのリソース構成図は図に示す通りになります。

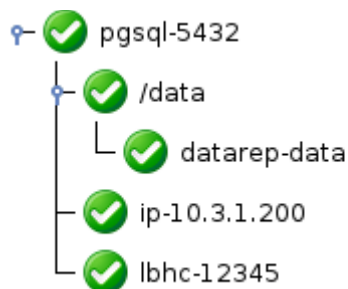


図 3.2-1 リソース構成図

以下は保護するリソースのリソース名と内容になります。

リソース名	説明
pgsql	PostgreSQL リソース
/data	ファイルシステムリソース
Datarep-data	データレプリケーションリソース
Ip-10.3.1.200	仮想 IP リソース
Lbhc-12345	LB Health Check リソース

3.3. 既知の問題の確認

既知の問題は、テクニカルドキュメンテーション内、「トラブルシューティング」セクションの「既知の問題と制限」に記載しています。ご利用の環境に該当する既知の問題がないかご確認ください。また、最新の情報につきましては、ユーザーポータル「既知の問題と制限」に記載されている場合がございますので、こちらも合わせてご確認ください。

3.4. 本ガイドを進めるにあたり、必要なソフトウェアイメージ

- LifeKeeper for Linux インストールメディア
- Red Hat Enterprise Linux OS イメージ

本ガイドは、クラスターノードをインターネットに公開せず、に LifeKeeper をインストールする方法をご案内しております。LifeKeeper のインストールにあたり、パッケージマネージャーがソフトウェアの依存関係を解決する必要があるため、Red Hat Enterprise Linux の OS イメージが必要になります。

4. Azure の上の環境構築

4.1. リソースグループの作成

今回のクラスターを構成する Azure のリソースグループを作成します。

- ① リソースグループページを開き「作成」をクリックします



図 4.1-1 リソースグループ 一覧画面

- ② 有効なサブスクリプションを選択し、リソースグループに名前を付けます。
以降は、本手順で作成したリソースグループ内にすべてのコンピューティングリソースを配置します。

基本 タグ 確認および作成

リソースグループ - Azure ソリューションの関連リソースを保持するコンテナ。リソースグループには、ソリューションのすべてのリソースを含めることも、グループとして管理したいリソースのみを含めることもできます。組織にとって最も有用なことに基づいて、リソースグループにリソースを割り当てる方法を決めてください。 [詳細情報](#)

プロジェクトの詳細

サブスクリプション * ⓘ

リソースグループ * ⓘ

リソースの詳細

リージョン * ⓘ

図 4.1-2 リソースグループ作成ウィザード「基本」タブ

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

- ③ 作成者がわかるようにタグをつけます。

リソース グループを作成します ...

基本 **✖ タグ** 確認および作成

Azure リソースをカテゴリに分けて論理的に整理するため、タグを運用します。タグは、キー (名前) と値で構成されます。タグ名は大文字と小文字が区別されず、タグ値は大文字と小文字が区別されます。 [詳細情報](#)


名前 ①	値 ①	リソース	
creator		リソースグループ	
		リソースグループ	

図 4.1-3 タグ設定画面

- ④ 作成を押すことで、リソースグループの作成は終了になります。

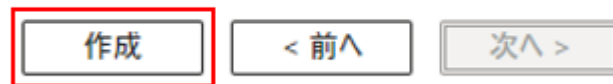


図 4.1-4 作成ボタン

4.2. 仮想ネットワークの作成

1. サイドメニューから仮想ネットワークを開き「作成」をクリックします。



図 4.2-1 仮想ネットワーク 一覧画面

2. 仮想ネットワークは以下の表にある通り設定を行います。

Resource group	LKL-QWK-majority
仮想ネットワーク名	Vnet-LKL-QWK-majority
地域	(Asia Pacific) 東日本



図 4.2-2 仮想ネットワーク 基本設定画面

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

次へ進みます。



図 4.2-3 仮想ネットワーク作成ウィザード下部

3. 「セキュリティ」タブは特に設定を行わず、デフォルトのまま進めます。

4. IP アドレス空間の設定を行います。

- ① デフォルトの IP アドレス空間を削除します。
- ② アドレス空間の追加を押下し、ウィザードに沿って作成します。

基本 セキュリティ IP アドレス タグ 確認および作成

必要な IPv4 および IPv6 アドレスとサブネットを使用して、仮想ネットワーク アドレス空間を構成します。 [詳細情報](#)

1 つ以上の IPv4 または IPv6 アドレスの範囲を使用して、仮想ネットワークのアドレス空間を定義します。アプリケーションで使用するために、仮想ネットワーク アドレス空間を小さい範囲にセグメント化するサブネットを作成します。サブネットにリソースをデプロイすると、Azure によってサブネットの IP アドレスがリソースに割り当てられます。 [詳細情報](#)

IP アドレス空間の追加 ②

サブネット	IP アドレスの範囲	サイズ	NAT ゲートウェイ
default	10.0.0.0 - 10.0.0.255	/24 (256 個のアドレス)	-

① アドレス空間のサイズ変更
アドレス空間の削除

① サブネットからの送信インターネット アクセスには、NAT ゲートウェイをお勧めします。サブネットを編集して NAT ゲートウェイを追加します。 [詳細情報](#)

図 4.2-4 仮想ネットワークの IP アドレス設定画面

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

新しいアドレス空間の設定は以下の表にある通りに設定を行います。

IP アドレス空間	
アドレス空間の種類	IPv4
開始アドレス	10.3.0.0
アドレス空間のサイズ	/16 (65536 個のアドレス)

IP アドレス空間の追加

×

仮想ネットワークのアドレス空間に、重複しないアドレスの範囲が 1 つ以上あります。プライベート (RFC 1918)、共有 (RFC 6598)、またはローカル (RFC 4193) のアドレスの範囲を使用することをお勧めします。 [詳細](#)。

アドレス空間の種類 IPv4
 IPv6

開始アドレス

アドレス空間のサイズ

IP アドレス空間 10.3.0.0 - 10.3.255.255 (65536 アドレス)

図 4.2-5 IP アドレス空間の作成画面

正常に追加されると、以下の画像のような表示になります。

IP アドレス空間の追加

10.0.0.0/16 + サブネットの追加

10.0.0.0 - 10.0.255.255 (65536 アドレス)

サブネット	IP アドレスの範囲	サイズ	NAT ゲートウェイ
-------	------------	-----	------------

図 4.2-6 IP アドレス空間の作成と追加終了後の画面

5. サブネットの作成

本ガイドは以下の設定でサブネットを2つ作成します。

- すべての仮想マシンが属するサブネット

IP アドレス空間	10.3.0.0
名前	ZoneAll
開始アドレス	10.3.1.0
サブネットサイズ	/24 (256 個のアドレス)
NAT ゲートウェイ	なし
ネットワークセキュリティグループ	なし
ルートテーブル	なし

- DataKeeper がデータレプリケーションに使用するサブネット

IP アドレス空間	10.3.0.0
名前	DataRep
開始アドレス	10.3.2.0
サブネットサイズ	/24 (256 個のアドレス)
NAT ゲートウェイ	なし
ネットワークセキュリティグループ	なし
ルートテーブル	なし

- ① サブネットの追加を押下します



図 4.2-7 サブネット追加ボタン

- ② ZoneAll のサブネットを作成します。

サブネットの追加

✕

アドレス空間を選択し、サブネットを構成します。選択したサービスを後で追加する予定の場合は、既定のサブネットをカスタマイズするか、サブネットテンプレートから選択できます。 [詳細情報](#)

IP アドレス空間 ①	<input type="text" value="10.3.0.0/16"/>
	10.3.0.0 - 10.3.255.255 (65536 アドレス)
サブネットの詳細	
サブネットテンプレート ①	<input type="text" value="Default"/>
名前 * ①	<input type="text" value="ZoneAll"/>
開始アドレス * ①	<input type="text" value="10.3.1.0"/>
サブネットサイズ ①	<input type="text" value="/24 (256 個のアドレス)"/>
IP アドレス空間 ①	10.3.1.0 - 10.3.1.255 (256 アドレス)
セキュリティ	
ネットワークアドレス変換ゲートウェイを使用して、仮想マシンのインターネットアクセスを簡略化します。ネットワークセキュリティグループを使用してサブネットトラフィックをフィルター処理します。 詳細情報	
NAT ゲートウェイ ①	<input type="text" value="なし"/>
	新規作成
ネットワークセキュリティグループ ①	<input type="text" value="なし"/>
	新規作成
ルートテーブル	<input type="text" value="なし"/>

図 4.2-8 Zone All サブネット設定画面

- ③ 同様の手順で DataRep のサブネットも作成します。

設定する設定値については、前ページの表をご参照ください。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

- ④ 2つのサブネットの作成が終了すると、Azure ポータル上では以下のような表示になります。



サブネット	IP アドレスの範囲	サイズ	NAT ゲートウェイ
ZoneAll	10.3.1.0 - 10.3.1.255	/24 (256 個のアドレス)	-
DataRep	10.3.2.0 - 10.3.2.255	/24 (256 個のアドレス)	-

図 4.2-9 サブネット追加後

6. 作成者を識別するために仮想ネットワークにもタグをつけます。

基本 セキュリティ IP アドレス タグ 確認および作成

タグは名前と値のペアで、同じタグを複数のリソースやリソースグループに適用することでリソースを分類したり、統合した請求を表示したりできるようにします。 [タグに関する詳細情報](#)

タグを作成してから別のタブでリソースの設定を変更すると、タグは自動的に更新されることにご注意ください。

名前	値	リソース
creator		すべてのリソースが選択されまし
		すべてのリソースが選択されまし

図 4.2-10 タグの設定画面

4.3. 可用性セットの作成

Azure の可用性セットを利用し、インフラストラクチャの冗長設定を行います。

1. 可用性セットの設定画面を開く。

[リソース、サービス、ドキュメントの検索] から可用性セットを検索します。

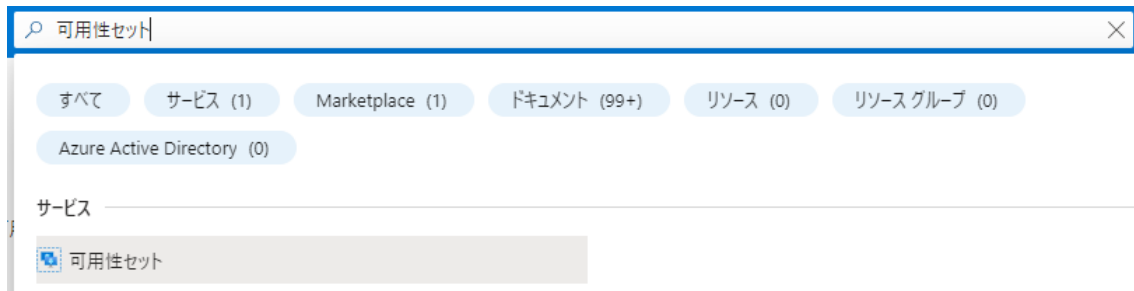


図 4.3-1 Azure ポータル検索画面

2. 可用性セットの作成

作成ボタンをクリックし、可用性セット作成ウィザードに移ります。



図 4.3-2 可用性セット一覧画面

3. 可用性セットの設定

以下の表の通り、設定をおこないます。

項目	設定値
リソースグループ	LKL-QWK-majority
名前	availability-set

可用性セットの作成

i 新しいお客様は、幅広いフィーチャーを備えた高可用性のために、フレキシブル オークストレーション モードの仮想マシン スケール セットを選択することをお勧めします。仮想マシン スケール セットを使用すると、VM インスタンスを一元的に管理、構成、更新でき、要求または定義されたスケジュールに応じて VM インスタンスの数を自動的に増減します。可用性セットは高可用性のみを提示します。

基本 詳細 タグ 確認および作成

可用性セットは、VM リソースの配置時に互いを分離する論理的なグループ化機能です。Azure では、可用性セット内に配置した VM が、複数の物理サーバー、コンピューティング ラック、ストレージ ユニット、ネットワーク スイッチで実行されるようにします。ハードウェアまたはソフトウェアの障害が発生した場合は、VM のサブセットのみが影響を受け、全体的なソリューションは動作可能のままです。可用性セットは、信頼性の高いクラウドソリューションを構築するために必要です。 [可用性セットの詳細情報](#)。

プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダーのようなリソース グループを使用して、すべてのリソースを整理し、管理します。

サブスクリプション *

リソース グループ * 新規作成

インスタンスの詳細

名前 *

地域 *

障害ドメイン

更新ドメイン

マネージド ディスクを使用 いいえ (クラシック) はい (配置)

図 4.3-3 可用性セット作成画面

4. タグの作成

作成者がわかるように、タグをつけておきます。

Azure リソースをカテゴリに分けて論理的に整理するため、タグを適用します。タグは、キー (名前) と値で構成されます。タグ名は大文字と小文字が区別されず、タグ値は大文字と小文字が区別されます。 [詳細情報](#)

名前	値	リソース
creator		リソース グループ
		リソース グループ

図 4.3-4 タグの設定画面

4.4. 仮想マシンの作成

本ガイドを構成する Azure 仮想マシンを Azure ポータル上で作成します。

① Witness サーバー兼 PostgreSQL クライアントサーバーの作成

本ガイドで Witness サーバーとして使用する仮想マシンを作成します。

また、Witness サーバーの他にも、各クラスターノードへの踏み台サーバーと、PostgreSQL のクライアントとしても使用します。本ガイドでは、以降 Witness ノードと呼称します

1. 仮想マシンの作成

Virtual Machines の画面を開き、作成を押下します。

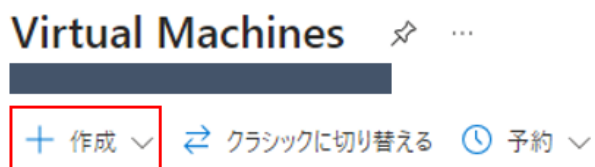


図 4.4-1 仮想マシンの一覧画面

2. 仮想マシンイメージの選択

OS の選択画面で「Red Hat Enterprise Linux」と入力し、Red Hat Inc が提供している Red Hat Enterprise Linux を選択します。

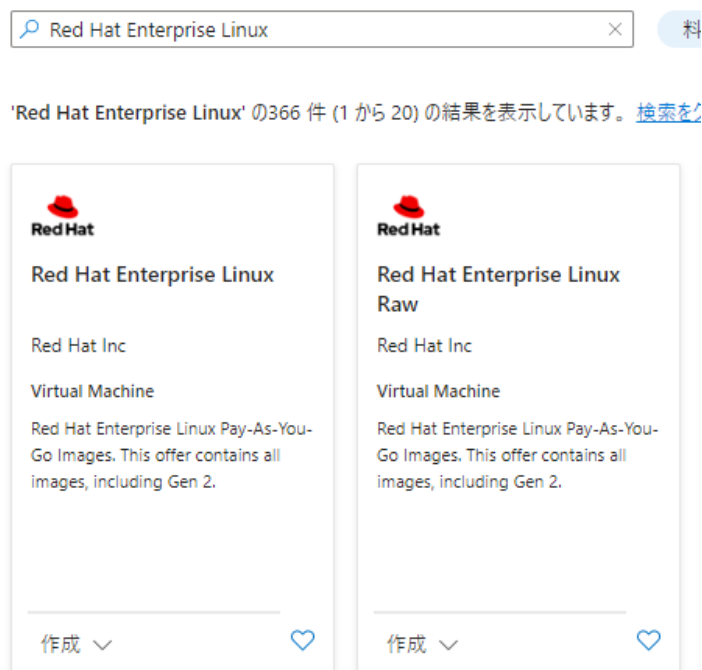


図 4.4-2 仮想マシンイメージ検索結果画面

3. 仮想マシンイメージから仮想マシンウィザードの起動

①Red Hat Enterprise Linux 仮想マシンイメージから 9.0 を選択し、②作成ボタンを押下します。

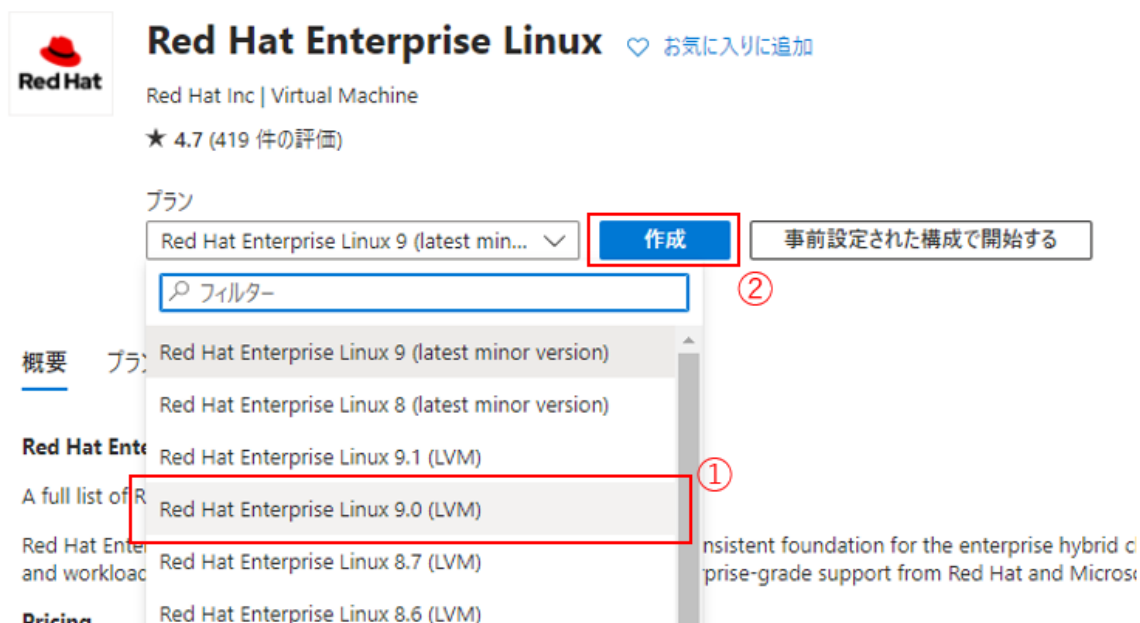


図 4.4-3 RedHat 仮想マシンイメージ選択画面

4. 基本の設定

Witness ノードは、以下の表のとおり、設定を行います。

項目	設定値
リソースグループ	LKL-QWK-majority
仮想マシン名	Clstr-node-witness
地域	(Asia Pacific) Japan East
可用性オプション	インフラストラクチャ冗長は必要ありません
セキュリティの種類	Standard
サイズ	Standard_B1s - 1 vcpu、1 GiB のメモリ
認証の種類	パスワード
ユーザー名	lkadmin
パスワード	xxxxxx (自分で設定した値 - SSH ログインで使用)
パスワードの確認	xxxxxx
パブリック受信ポート	選択したポートを許可する
受信ポートを選択	SSH (22)

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

表を見ながら赤枠部分を設定値に変更します。

基本 ディスク ネットワーク 管理 監視 詳細 タグ 確認および作成

Linux または Windows を実行する仮想マシンを作成します。Azure Marketplace からイメージを選択するか、独自のカスタマイズされたイメージを使用します。[基本] タブに続いて [確認と作成] を完了させて既定のパラメーターで仮想マシンをプロビジョニングするか、それぞれのタブを確認してフル カスタマイズを行います。 [詳細情報](#)

プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダーのようなリソース グループを使用して、すべてのリソースを整理し、管理します。

サブスクリプション *

リソースグループ * 新規作成

インスタンスの詳細

仮想マシン名 *

地域 *

可用性オプション

セキュリティの種類

イメージ * すべてのイメージを表示 | VM の世代の構成

VM アーキテクチャ ARM64 x64
Arm64 は、選択したイメージではサポートされていません。

サイズ * すべてのサイズを表示

管理者アカウント

認証の種類 SSH 公開キー パスワード

ユーザー名 * ✓

パスワード * ✓

パスワードの確認 * ✓

受信ポートの規則

パブリック インターネットからアクセスできる仮想マシン ネットワークのポートを選択します。[ネットワーク] タブで、より限定的または細かくネットワーク アクセスを指定できます。

パブリック受信ポート * なし 選択したポートを許可する

受信ポートを選択 *

インターネットからのすべてのトラフィックは、既定でブロックされます。受信ポートのルールは、[VM] > [ネットワーク] ページから変更できます。

図 4.4-4 仮想マシンの基本設定

5. ディスクの設定

ディスクをデフォルトの Premium SSD から Standard SSD に変更します。

基本 ディスク ネットワーク 管理 監視 詳細 タグ 確認および作成

Azure VM には、1 つのオペレーティング システム ディスクと短期的なストレージの一時的ディスクがあります。追加のデータ ディスクをアタッチできます。VM のサイズによって、使用できるストレージの種類と、許可されるデータ ディスクの数が決まります。 [詳細情報](#)

VM ディスクの暗号化

Azure Disk Storage の暗号化では、クラウドへ保持する場合に、既定では保存時に Azure マネージド ディスク (OS ディスクおよびデータ ディスク) に保存されるデータが自動的に暗号化されます。

ホストでの暗号化

i 選択したサブスクリプションには、ホストでの暗号化が登録されていません。
[この機能の有効化に関する詳細情報](#)

OS ディスク

OS ディスクの種類 *

Standard SSD (ローカル冗長ストレージ)

より短い待機時間、より高い IOPS と帯域幅、およびハードには、Premium SSD のディスクを選択してください。Premium SSD ディスクを持つ単一インスタンスの仮想マシンは、99.9% の継続性 SLA の資格があります。 [詳細情報](#)

VM と共に削除

キーの管理

プラットフォーム マネージド キー

Ultra Disk の互換性を有効にする

Ultra Disk は、可用性セットまたは VMSS ではサポートされていません。

図 4.4-5 ディスクの変更

6. ネットワークの設定

Witness ノードのネットワークの設定を行います。

設定項目は以下の表の通り設定を行ってください。

項目	設定値
仮想ネットワーク	vnet-LKL-QWK-majority
サブネット	ZoneAll
パブリック IP	新規作成 (デフォルトの値を使用)
NIC ネットワークセキュリティグループ	Basic
パブリック受信ポート	選択したポートを許可する
受信ポートを選択	SSH (22)
VM が削除されたときに NIC を削除する	チェックを入れる
負荷分散のオプション	なし

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

基本 ディスク ネットワーク 管理 監視 詳細 タグ 確認および作成

ネットワーク インターフェイスカード (NIC) 設定を構成して仮想マシンのネットワーク接続を定義します。セキュリティグループの規則によりポートや受信および送信接続を制御したり、既存の負荷分散ソリューションの背後に配置したりすることができます。 [詳細情報](#)

ネットワーク インターフェイス

仮想マシンの作成中に、ユーザー用にネットワーク インターフェイスが作成されます。

仮想ネットワーク *	<input type="text" value="vnet-LKL-QWK-majority"/>	▼
サブネット *	<input type="text" value="ZoneAll (10.3.1.0/24)"/>	▼
パブリック IP	<input type="text" value="(新規) clstrnodewitnessip577"/>	▼
NIC ネットワーク セキュリティグループ	<input type="radio"/> なし <input checked="" type="radio"/> Basic <input type="radio"/> 詳細	
パブリック受信ポート *	<input type="radio"/> なし <input checked="" type="radio"/> 選択したポートを許可する	
受信ポートを選択 *	<input type="text" value="SSH (22)"/>	▼

⚠ これにより、すべての IP アドレスが仮想マシンにアクセスできるようになります。これはテストにのみ推奨されます。[ネットワーク] タブの詳細設定コントロールを使用して、受信トラフィックを既知の IP アドレスに制限するための規則を作成します。

VM が削除されたときにパブリック IP と NIC を削除する

負荷分散

既存の Azure 負荷分散ソリューションのバックエンドプールにこの仮想マシンを配置できます。 [詳細情報](#)

負荷分散のオプション	<input checked="" type="radio"/> なし <input type="radio"/> Azure Load Balancer すべての TCP または UDP ネットワーク トラフィック、ポート フォワーディング、送信フローをサポートしています。 <input type="radio"/> アプリケーションゲートウェイ URL ベースのルーティング、SSL 終了、セッション永続化、Web アプリケーション ファイアウォールを含む HTTP または HTTPS の Web トラフィック ロード バランサーです。
------------	--

図 4.4-6 仮想マシンのネットワーク設定

7. 管理・監視・詳細 の設定

LifeKeeper のクラスター構築に際して必要な設定はないためスキップします。
必要に応じ、お客様の環境に合わせて適宜設定してください。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

8. タグの設定

タグを設定します。

タグは名前と値のペアで、同じタグを複数のリソースやリソースグループに適用することでリソースを分類したり、統合した請求を表示したりできるようにします。 [タグに関する詳細情報](#)

タグを作成してから別のタブでリソースの設定を変更すると、タグは自動的に更新されることにご注意ください。


名前	値	リソース
creator		すべてのリソースが選択されました 
		すべてのリソースが選択されました

図 4.4-7 タグの設定画面

9. 確認及び作成

上記手順で作成した仮想マシンの確認を行います。

特にネットワークの設定に間違いがない事を再度ご確認ください。

② クラスターノードの作成

LK クラスターを構成するクラスターノード clstr-node-A, clstr-node-B (以降、ノード A, ノード B と呼称します) をそれぞれ作成します。

ノード A の作成

1. 仮想マシンの作成

仮想マシンイメージの選択は Witness ノードの手順 [3. 仮想マシンイメージから仮想マシンウィザードの起動] までと同様に行います

2. 基本の設定

ノード A は、以下の表のとおり、設定を行います。

項目	設定値
リソースグループ	LKL-QWK-majority
仮想マシン名	clstr-node-A
地域	(Asia Pacific) Japan East
可用性オプション	可用性セット
可用性セット	availability-set
セキュリティの種類	Standard
サイズ	Standard_B1s - 1 vcpu, 1 GiB のメモリ
認証の種類	パスワード
ユーザー名	lkadmin
パスワード	xxxxxx (自分で設定した値 - SSH ログインで使用)
パスワードの確認	xxxxxx
パブリック受信ポート	なし

次ページ画像の赤枠部分を設定値に変更します。

仮想マシンの作成 ...

⚠ 基本オプションを変更すると、選択した内容がリセットされることがあります。仮想マシンを作成する前に、すべてのオプションを確認してください。

基本 ディスク ネットワーク 管理 監視 詳細 タグ 確認および作成

Linux または Windows を実行する仮想マシンを作成します。Azure Marketplace からイメージを選択するか、独自のカスタマイズされたイメージを使用します。[基本] タブに続いて [確認と作成] を完了させて既定のパラメーターで仮想マシンをプロビジョニングするか、それぞれのタブを確認してフルカスタマイズを行います。 [詳細情報](#)

プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダーのようなリソース グループを使用して、すべてのリソースを整理し、管理します。

サブスクリプション * ⓘ

リソースグループ * ⓘ
[新規作成](#)

インスタンスの詳細

仮想マシン名 * ⓘ

地域 * ⓘ

可用性オプション ⓘ

i 入力に基づいて、このリソースを仮想マシン スケール セットとして作成することを検討してください。これにより、負荷分散された仮想マシンを管理、構成、スケーリングできます。 [VMSSとして作成](#)

可用性セット * ⓘ
[新規作成](#)

セキュリティの種類 ⓘ

イメージ * ⓘ
[すべてのイメージを表示](#) | [VM の世代の構成](#)

図 4.4-8 仮想マシン (ノード A) の設定画面 1

VM アーキテクチャ ① ARM64
 x64
i Arm64 は、選択したイメージではサポートされていません。

Azure Spot 割引で実行する ①

サイズ * ①
[すべてのサイズを表示](#)

管理者アカウント

認証の種類 ① SSH 公開キー
 パスワード

ユーザー名 * ① ✓

パスワード * ① ✓

パスワードの確認 * ① ✓

受信ポートの規則

パブリック インターネットからアクセスできる仮想マシン ネットワークのポートを選択します。[ネットワーク] タブで、より限定的または細かくネットワーク アクセスを指定できます。

パブリック受信ポート * ① なし
 選択したポートを許可する

受信ポートを選択

i インターネットからのすべてのトラフィックは、既定でブロックされます。受信ポートのルールは、[VM] > [ネットワーク] ページから変更できます。

図 4.4-9 仮想マシン (ノード A) の設定画面 2

3. ディスクの設定

OS ディスクの種類の変更 及び PostgreSQL で使用するディスクの追加を行います。

1. ディスクの種類を変更

OS ディスクの種類をデフォルトの Premium SSD から Standard SSD に変更します。

基本 ディスク ネットワーク 管理 監視 詳細 タグ 確認および作成

Azure VM には、1 つのオペレーティング システム ディスクと短期的なストレージの一時ディスクがあります。追加のデータ ディスクをアタッチできます。VM のサイズによって、使用できるストレージの種類と、許可されるデータ ディスクの数が決まります。 [詳細情報](#)

VM ディスクの暗号化

Azure Disk Storage の暗号化では、クラウドへ保持する場合に、既定では保存時に Azure マネージド ディスク (OS ディスクおよびデータ ディスク) に保存されるデータが自動的に暗号化されます。

ホストでの暗号化

i 選択したサブスクリプションには、ホストでの暗号化が登録されていません。
[この機能の有効化に関する詳細情報](#)

OS ディスク

OS ディスクの種類 *

より短い待機時間、より高い IOPS と帯域幅、およびハードウェアには、Premium SSD のディスクを選択してください。Premium SSD ディスクを持つ単一インスタンスの仮想マシンは、99.9% の継続性 SLA の資格があります。 [詳細情報](#)

VM と共に削除

キーの管理

Ultra Disk の互換性を有効にする

Ultra Disk は、可用性セットまたは VMSS ではサポートされていません。

図 4.4-10 ディスクの設定画面

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

2. PostgreSQL で使用するディスクの追加

追加するディスクは以下の設定の通りです。

項目	設定値
ディスクの種類	Standard SSD
ディスクのサイズ	16 GiB ¹

1. 新しいディスクの作成

新しいディスクの作成と接続 を押下します。

clstr-node-A の データ ディスク

仮想マシンに別のデータ ディスクを追加および構成したり、既存のディスクを接続したりすることができます。この VM には、一時ディスクも付属しています。

LUN	名前	サイズ (...)	ディスクの種類	ホスト キャッ...	VM と共に削除 ①

[新しいディスクを作成し接続する](#) [既存のディスクの接続](#)

図 4.4-11 ディスクの追加ボタン

2. 新規ディスクのサイズの変更

デフォルトの追加ディスクのサイズは 1024 GiB のため、16 GiB に変更します。

新しいディスクを作成する ...

VM にアプリケーションとデータを格納するための新しいディスクを作成します。ディスクの料金は、ディスク サイズ、ストレージの種類、およびトランザクションの数などの要因に応じて異なります。 [詳細情報](#)

名前 *	<input type="text" value="clstr-node-A_DataDisk_0"/>
ソースの種類 * ①	<input type="text" value="なし (空のディスク)"/>
サイズ * ①	<div>1024 GiB Premium SSD LRS サイズを変更します</div>
キーの管理 ①	<input type="text" value="プラットフォーム マネージドキー"/>
共有ディスクを有効にする	<input type="radio"/> はい <input checked="" type="radio"/> いいえ
VM と共にディスクを削除	<input type="checkbox"/>

図 4.4-12 追加ディスクのサイズ変更ボタン

¹ テスト環境なので 16GB にしました。実際の運用では必要なサイズに変更ください。

3. ディスクサイズの設定

ストレージの種類 から、Standard SSD に変更します。

利用可能なディスク サイズとその機能を参照します。

ストレージの種類

Standard SSD (ローカル冗長ストレージ)

ローカル冗長ストレージ (データは単一のデータセンター内でレプリケートされま す)	プロビジョニングされた IOPS	プロビジョニングされたスループ ット
Premium SSD 実稼働およびパフォーマンスが要求されるワークロードに最適です	500	60
Standard SSD Web サーバー、使用頻度の低いエンタープライズ アプリケーション、Dev/Test に 最適です	500	60
Standard HDD バックアップ用、重要でない、頻度の低いアクセスに最適です	500	60
ゾーン冗長ストレージ (データは 3 つのゾーンにレプリケートされます)	500	60
Premium SSD ゾーン障害に対するストレージの回復性が必要な実稼働ワークロードに最適で す	500	60
Standard SSD ゾーン障害に対するストレージの回復性が必要な Web サーバー、使用頻度の 低いエンタープライズ アプリケーション、Dev/Test に最適です	2000	400
	4000	600
	6000	750

カスタム ディスク サイズ (GiB) *

1024

図 4.4-13 追加ディスクの種類選択

また、ディスクのサイズを 16GiB に変更します。

ストレージの種類

Standard SSD (ローカル冗長ストレージ)

サイズ	パフォーマンス レベル	プロビジョニングされた IOPS
4 GiB	E1	500
8 GiB	E2	500
16 GiB	E3	500
32 GiB	E4	500
64 GiB	E6	500
128 GiB	E10	500
256 GiB	E15	500
512 GiB	E20	500
1024 GiB	E30	500
2048 GiB	E40	500
4096 GiB	E50	500
8192 GiB	E60	2000
16384 GiB	E70	4000
32767 GiB	E80	6000

カスタム ディスク サイズ (GiB) *

16

図 4.4-14 ディスクのサイズ変更画面

3. 確認

以上の手順が終了すると、以下の画像のように表示が変わります。

新しいディスクを作成する ...

VM にアプリケーションとデータを格納するための新しいディスクを作成します。ディスクの料金は、ディスク サイズ、ストレージの種類、およびトランザクションの数などの要因に応じて異なります。 [詳細情報](#)

名前 *	<input type="text" value="clstr-node-A_DataDisk_0"/>
ソースの種類 * ①	<input type="text" value="なし (空のディスク)"/>
サイズ * ①	<div style="border: 1px solid red; padding: 2px;"> 16 GiB Standard SSD LRS サイズを変更します </div>
キーの管理 ①	<input type="text" value="プラットフォーム マネージド キー"/>
共有ディスクを有効にする	<input type="radio"/> はい <input checked="" type="radio"/> いいえ
VM と共にディスクを削除	<input checked="" type="checkbox"/>

図 4.4-15 ディスクの確認画面

表示が変わった事を確認したら、OK を押下します。

4. ネットワークの設定

ノード A のネットワークの設定を行います。

設定項目は以下の表の通り設定を行ってください。

項目	設定値
仮想ネットワーク	vnet-LKL-QWK-majority
サブネット	ZoneAll
パブリック IP	新規作成 (デフォルトの値を使用)
NIC ネットワークセキュリティグループ	Basic
パブリック受信ポート	なし
VM が削除されたときに NIC を削除する	チェックを入れる
負荷分散のオプション	なし

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

基本 ディスク ネットワーク 管理 監視 詳細 タグ 確認および作成

ネットワーク インターフェイスカード (NIC) 設定を構成して仮想マシンのネットワーク接続を定義します。セキュリティグループの規則によりポートや受信および送信接続を制御したり、既存の負荷分散ソリューションの背後に配置したりすることができます。 [詳細情報](#)

ネットワーク インターフェイス

仮想マシンの作成中に、ユーザー用にネットワーク インターフェイスが作成されます。

仮想ネットワーク *	<input type="text" value="vnet-LKL-QWK-majority"/> 新規作成
サブネット *	<input type="text" value="ZoneAll (10.3.1.0/24)"/> サブネット構成の管理
パブリック IP	<input type="text" value="(新規) clstrnodewitnessip577"/> 新規作成
NIC ネットワーク セキュリティグループ	<input type="radio"/> なし <input checked="" type="radio"/> Basic <input type="radio"/> 詳細
パブリック受信ポート *	<input type="radio"/> なし <input checked="" type="radio"/> 選択したポートを許可する
受信ポートを選択 *	<input type="text" value="SSH (22)"/>

⚠ これにより、すべての IP アドレスが仮想マシンにアクセスできるようになります。これはテストにのみ推奨されます。 [ネットワーク] タブの詳細設定コントロールを使用して、受信トラフィックを既知の IP アドレスに制限するための規則を作成します。

VM が削除されたときにパブリック IP と NIC を削除する

負荷分散

既存の Azure 負荷分散ソリューションのバックエンド プールにこの仮想マシンを配置できます。 [詳細情報](#)

負荷分散のオプション	<input checked="" type="radio"/> なし <input type="radio"/> Azure Load Balancer すべての TCP または UDP ネットワーク トラフィック、ポート フォワーディング、送信フローをサポートしています。 <input type="radio"/> アプリケーションゲートウェイ URL ベースのルーティング、SSL 終了、セッション永続化、Web アプリケーション ファイアウォールを含む HTTP または HTTPS の Web トラフィック ロード バランサーです。
------------	--

図 4.4-16

5. 管理・監視・詳細 の設定

LK のクラスター構築に際して特に必要な設定はないため、スキップします。
必要に応じ、お客様の環境に合わせて適宜設定してください。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

6. タグの設定

タグを設定します。

タグは名前と値のペアで、同じタグを複数のリソースやリソースグループに適用することでリソースを分類したり、統合した請求を表示したりできるようにします。 [タグに関する詳細情報](#)

タグを作成してから別のタブでリソースの設定を変更すると、タグは自動的に更新されることにご注意ください。


名前	値	リソース
creator		すべてのリソースが選択されました 
		すべてのリソースが選択されました

図 4.4-17 タグの設定画面

7. 確認及び作成

上記手順で作成した仮想マシンの確認を行います。

特にネットワークの設定に間違いがない事を再度ご確認ください。

ノード B の作成

ノード A と同様の手順でノード B を作成します。各設定項目は以下の表の通りです。

基本 タブの設定

項目	設定値
リソースグループ	LKL-QWK-majority
仮想マシン名	clstr-node-B
地域	(Asia Pacific) Japan East
可用性オプション	可用性セット
可用性セット	availability-set
セキュリティの種類	Standard
サイズ	Standard_B1s - 1 vcpu、1 GiB のメモリ
認証の種類	パスワード
ユーザー名	lkadmin
パスワード	xxxxxx (自分で設定した値 - SSH ログインで使用)
パスワードの確認	xxxxxx
パブリック受信ポート	なし

ディスクの設定

項目	設定値
ディスクの種類	Standard SSD
ディスクのサイズ	16 GiB

ネットワークの設定

項目	設定値
仮想ネットワーク	vnet-LKL-QWK-majority
サブネット	ZoneAll
パブリック IP	新規作成 (デフォルトの値を使用)
NIC ネットワークセキュリティグループ	Basic
パブリック受信ポート	なし
VM が削除されたときに NIC を削除する	チェックを入れる
負荷分散のオプション	なし

③ 静的 IP の設定

各 NIC に振られる IP アドレスを静的 IP アドレスに変更します。

1. 仮想マシンのサイドメニューからネットワークを選択

仮想マシンの概要ページを開き、サイドバーからネットワークを選択します。

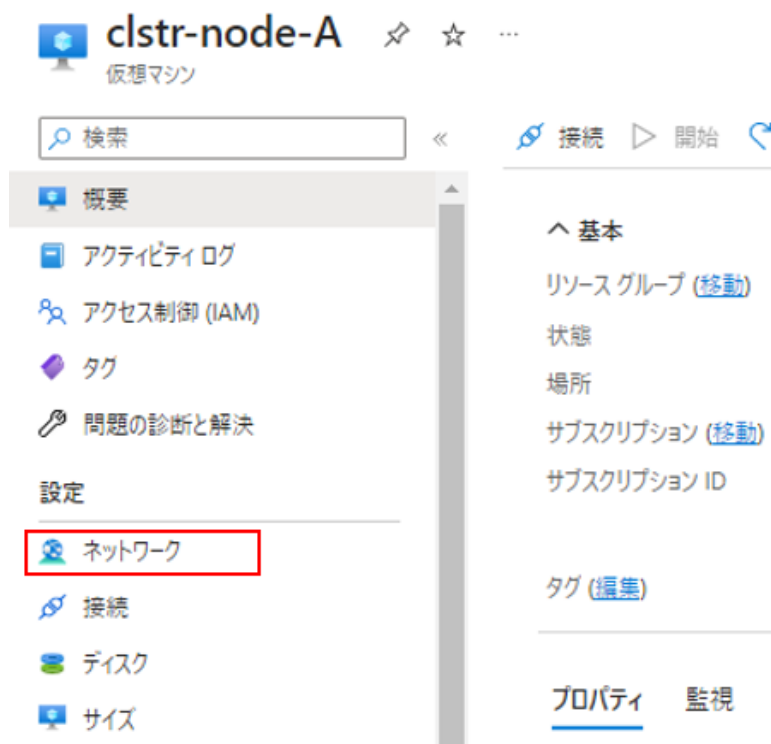


図 4.4-18 ノードのネットワーク設定画面

2. 既存のネットワークインターフェイスの設定に移動

ネットワークインターフェイスをクリックし、ネットワークインターフェイスの設定に移動します。



図 4.4-19 ネットワークインターフェイスの設定ボタン

3. IP 構成を選択

サイドバーから IP 構成をクリックします。

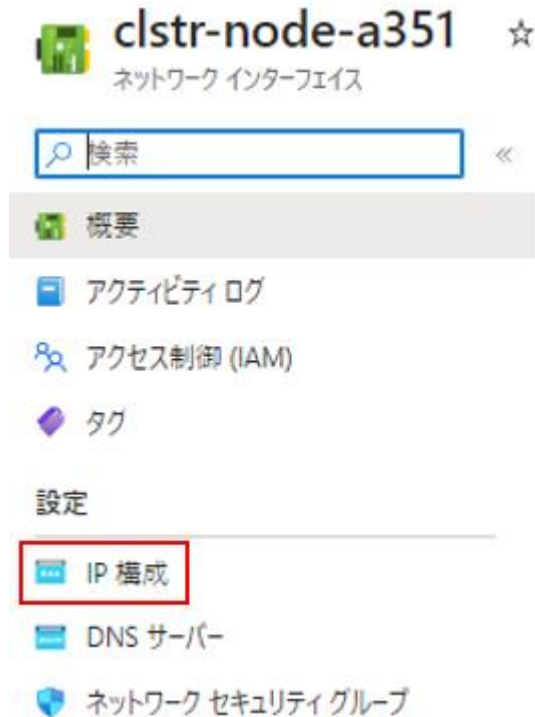


図 4.4-20 NIC 概要画面 サイドバー

4. IP 構成の変更

Ipconfig 1 の行をクリックし、設定画面に移動します。

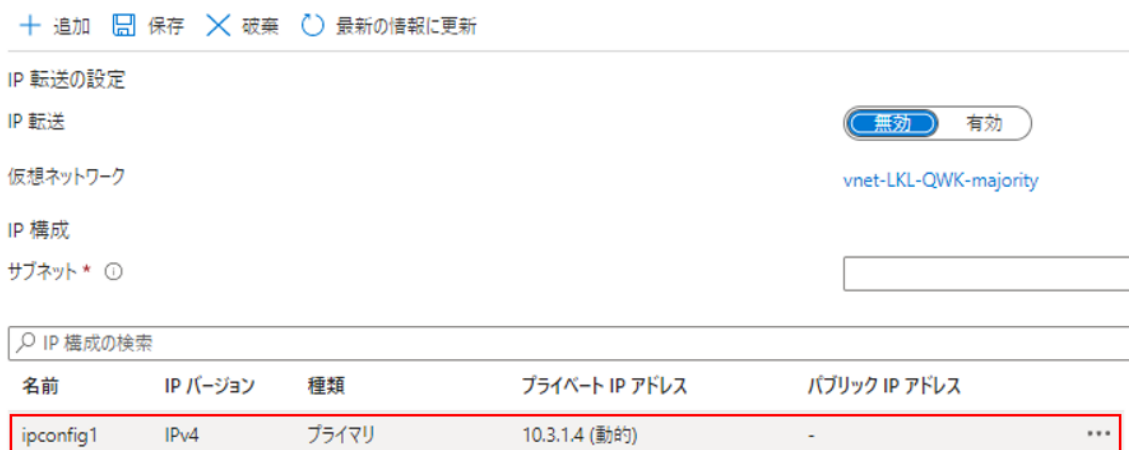


図 4.4-21 IP 構成編集画面

IP の設定を動的から静的に変更し、割り振る IP アドレスを入力します。

ipconfig1 ...
clstr-node-a351

保存 破棄

! このネットワーク インターフェイスに関連付けられた仮想マシンは、新しいプライベート IP アドレスを利用できるようにするため、再起動されます。ネットワーク インターフェイスは再プロビジョニングされますが、セカンダリ IP アドレス、サブネット マスク、デフォルト ゲートウェイを含むネットワーク構成設定は仮想マシン内で手動で再構成する必要があります。[詳細情報](#)

パブリック IP アドレスの設定

パブリック IP アドレス

関連付け解除 関連付け

プライベート IP アドレスの設定

仮想ネットワーク/サブネット
vnet-LKL-QWK-majority/ZoneAll

割り当て

動的 静的

IP アドレス *

10.3.1.11 ✓

図 4.4-22 IP 静的アドレスの割り当てと

5. 上記手順をすべてのノードで行います。

各種設定は以下の表の通りです。

仮想マシン名	IP アドレス
clstr-node-witness	10.3.1.50
clstr-node-A	10.3.1.11
clstr-node-B	10.3.1.12

④ 各ノードに NIC を追加

ノード A にデータレプリケーション用の NIC を追加します。

1. 仮想マシンの再度メニューからネットワークを選択

静的 IP の 1 の設定と同様の手順で仮想マシンのネットワーク設定画面に移動します。



図 4.4-23 仮想マシンのネットワーク設定画面

2. ネットワークインターフェイスの新規作成

ネットワークインターフェイスの作成と接続を押下します。

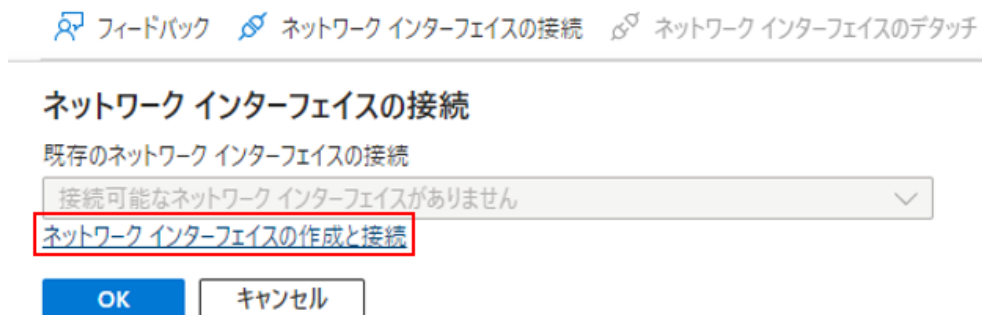



図 4.4-24 ネットワークインターフェイスの作成と接続


3. ネットワークインターフェイスの設定


新規 NIC を以下の表の設定で作成します。

項目	設定値
名前	Clstr-vnic-node-A
サブネット	DataRep (10.3.2.0/24)
NIC ネットワークセキュリティグループ	Basic
パブリック受信ポート	なし
プライベート IP アドレスの割り当て	静的
プライベート IP アドレス	10.3.2.11


プロジェクトの詳細


サブスクリプション 

リソース グループ * 

LKL-QWK-majority 


新規作成


場所 

(Asia Pacific) Japan East 


ネットワーク インターフェイス


名前 *


clstr-vnic-node-A 

仮想ネットワーク 

vnet-LKL-QWK-majority

サブネット * 


DataRep (10.3.2.0/24) 

NIC ネットワーク セキュリティ グループ 

なし

Basic


詳細


パブリック受信ポート * 

なし

選択したポートを許可する

受信ポートを選択

1 つ以上のポートを選択してください 


 インターネットからのすべてのトラフィックは、既定でブロックされます。受信ポートのルールは、[VM] > [ネットワーク] ページから変更できます。

プライベート IP アドレスの割り当て

動的

静的

プライベート IP アドレス *

10.3.2.11 

プライベート IP アドレス (IPv6)

図 4.4-25 新しく追加するネットワークインターフェイスの設定画面

4. 他ノードへの NIC の追加

上記の手順で Witness ノード及びノード B にも追加していきます。

なお、追加する NIC の設定は以下の表の通りになります。

Witness ノード

項目	設定値
名前	Clstr-vnic-node-witness
サブネット	DataRep (10.3.2.0/24)
NIC ネットワークセキュリティグループ	Basic
パブリック受信ポート	なし
プライベート IP アドレスの割り当て	静的
プライベート IP アドレス	10.3.2.50

ノード B

項目	設定値
名前	Clstr-vnic-node-B
サブネット	DataRep (10.3.2.0/24)
NIC ネットワークセキュリティグループ	Basic
パブリック受信ポート	なし
プライベート IP アドレスの割り当て	静的
プライベート IP アドレス	10.3.2.12

4.5. インターナルロードバランサーの作成

Azure では IP リソースが保護する仮想 IP アドレスを Azure の仮想ネットワークで認識する事が出来ません。

この影響で、通常 LifeKeeper for Linux が想定している IP リソースが保護する仮想 IP アドレスによるネットワーク通信を行うことが出来ません。

その為 LifeKeeper では、インターナルロードバランサー (以降、ILB と呼称) の導入を行い、ILB が設定する仮想 IP アドレスをネットワーク通信経路として設定します。

詳細はテクニカルドキュメンテーションをご参照ください。

<https://docs.us.sios.com/spslinux/9.7.0/ja/topic/lifekeeper-specific-configurations-in-azure>

① ロードバランサーの作成

Azure ポータルの検索画面からロードバランサーをクリックします。

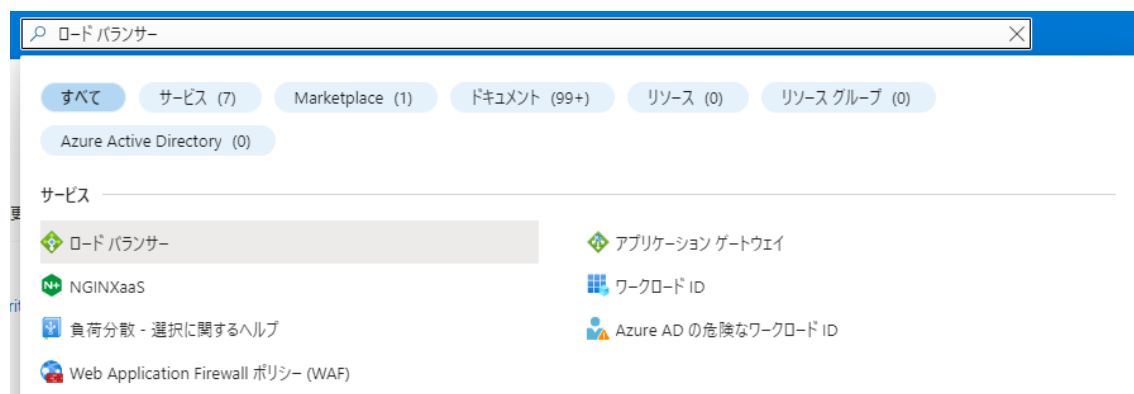


図 4.5-1 Azure ポータル検索画面

② ロードバランサー作成ウィザードを開く

作成ボタンをクリックし、ロードバランサーの作成ウィザードを開きます。



図 4.5-2 ロードバランサーの作成

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

③ ロードバランサー 基本の設定

以下の設定でロードバランサーの基本の設定を行います。

項目	設定値
リソースグループ	LKL-QWK-majority
名前	LKL-QWK-majority
SKU	Standard
種類	内部

基本 フロントエンド IP 構成 バックエンドプール インバウンド規則 送信規則 タグ 確認および作成

Azure Load Balancer は、正常な仮想マシン インスタンス間で着信トラフィックを分散する、第 4 層のロード バランサーです。ハッシュベースの分散アルゴリズムを使用します。既定では、5 つの組 (ソース IP、ソース ポート、接続先 IP、接続先ポート、プロトコルの種類) のハッシュを使用して、使用可能なサーバーにトラフィックをマップします。インターネットに接続してパブリック IP アドレスでアクセスできるようにすることや、内部に配置して仮想ネットワークからのみアクセスできるようにすることができます。また、ネットワーク アドレス変換 (NAT) を使用して、パブリック IP アドレスとプライベート IP アドレス間でトラフィックをルーティングすることもできます。 [詳細](#)。

プロジェクトの詳細

サブスクリプション *

リソースグループ * 新規作成

インスタンスの詳細

名前 * ✓

地域 *

SKU * ⓘ Standard
 ゲートウェイ
 Basic

i Microsoft では、運用ワークロードには Standard SKU ロード バランサーをお勧めしています。
[Standard と Basic の SKU の価格差に関する詳細情報](#)

種類 * ⓘ パブリック
 内部

レベル * 地域
 グローバル

図 4.5-3 ロードバランサー基本設定

④ フロントエンド IP 構成の作成

1. フロントエンド IP の設定

フロントエンド IP 構成の追加をクリックし、IP 構成の追加ウィザードを開きます。

基本 フロントエンド IP 構成 バックエンドプール インバウンド規則 送信規則 タグ 確認および作成

フロントエンド IP 構成とは、負荷分散、インバウンド NAT、アウトバウンド規則内で定義されているインバウンドまたはアウトバウンド通信に使用される IP アドレスです。

+ フロントエンド IP 構成の追加

図 4.5-4 フロントエンド IP 構成

2. フロントエンド IP 構成の設定

表に従い、フロントエンド IP の構成を作成します。

項目	設定値
名前	LKL-QWK-LB
仮想ネットワーク	Vnet-LKL-QWK-majority
サブネット	ZoneAll (10.3.1.0/24)
割り当て	静的
IP アドレス	10.3.1.200

フロントエンド IP 構成の追加 ×

名前 *

LKL-QWK-LB ✓

仮想ネットワーク *

vnet-LKL-QWK-majority (LKL-QWK-majority) ▾

サブネット *

ZoneAll (10.3.1.0/24) ▾

割り当て

動的 静的

IP アドレス *

10.3.1.200 ✓

可用性ゾーン * ⓘ

ゾーン冗長 ▾

図 4.5-5 フロントエンド IP 構成の変更画面

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

3. 確認

フロントエンド IP 構成に作成したフロントエンド IP が正常に追加されている事を確認します。

基本 フロントエンド IP 構成 バックエンドプール インバウンド規則 送信規則 タグ 確認および作成

フロントエンド IP 構成とは、負荷分散、インバウンド NAT、アウトバウンド規則内で定義されているインバウンドまたはアウトバウンド通信に使用される IP アドレスです。

+ フロントエンド IP 構成の追加

名前 ↑↓	IP アドレス ↑↓	仮想ネットワーク ↑↓	サブネット ↑↓	
QWK-QWK-majority	10.0.1.100	vnet-LKL-QWK-majority	Zone1	

図 4.5-6 フロントエンド IP 構成追加後の画面

⑤ バックエンドプール

1. バックエンドプールの追加

バックエンドプールの追加ボタンを押し、ウィザードを開きます。

基本 フロントエンド IP 構成 バックエンドプール インバウンド規則 送信規則 タグ 確認および作成

バックエンドプールは、ロードバランサーによるトラフィックの送信先にすることができるリソースのコレクションです。バックエンドプールには、仮想マシン、仮想マシン スケール セット、コンテナを含めることができます。

+ バックエンドプールの追加

名前	仮想ネットワーク	リソース名	ネットワーク インターフェ...	IP アドレス	可用性ゾーン
作業を開始するにはバックエンドプールを追加します					

図 4.5-7 バックエンドプールの一覧画面

バックエンドプールに名前を付けます。

バックエンドプールの追加 ... ×

名前 *

仮想ネットワーク  LKL-QWK-majority

バックエンドプールの構成

NIC

IP アドレス

図 4.5-8 バックエンドプールの作成画面

2. NIC の選択

バックエンドプールに追加する NIC をチェックします。

追加する NIC はノード A とノード B の 10.3.1.0/24 サブネット内に配置した NIC を設定します。

リソース名	リソースグループ	種類	IP 構成	IP アドレス	可用性セット
仮想マシン (3)					
<input checked="" type="checkbox"/> clstr-node-A	LKL-QWK-MAJORITY	仮想マシン	ipconfig1	10.3.1.11	AVAILABILITY-SET
<input checked="" type="checkbox"/> clstr-node-B	LKL-QWK-majority	仮想マシン	ipconfig1	10.3.1.12	AVAILABILITY-SET
<input type="checkbox"/> clstr-node-witness	LKL-QWK-majority	仮想マシン	ipconfig1	10.3.1.50	-

図 4.5-9 バックエンドプールに設定する NIC の選択

⑥ 負荷分散規則の作成

1. 負荷分散規則の追加

インバウンド規則から負荷分散規則の追加を押下します。

基本 フロントエンド IP 構成 バックエンドプール インバウンド規則 送信規則 タグ 確認および作成

負荷分散規則

負荷分散ルールでは、選択した IP アドレスとポートの組み合わせに送信される着信トラフィックを、バックエンド プール インスタンスのグループ全体に分散します。負荷分散ルールでは、正常性プローブを使用して、トラフィックを受信する資格のあるバックエンド インスタンスを特定します。

負荷分散規則の追加

図 4.5-10

2. 負荷分散規則の設定

負荷分散規則は以下の通りに設定を行います。

項目	設定値
名前	LKL-QWK-LB-PostgreSQL
フロントエンド IP アドレス	LKL-QWK-LB (10.3.1.200)
バックエンドプール	LKL-LB-backpool
ポート	5432
バックエンドポート	5432
正常性プローブ	新規作成
フローティング IP	有効 ²

² クラスターへ向けた通信は、仮想 IP 宛の packets 通信とする必要があるため、Floating IP address は必ず有効にしてください。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure
Step by step Guide

名前 *
 ✓

IPバージョン *
 IPv4
 IPv6

フロントエンド IP アドレス * ⓘ
 ▼

バックエンドプール * ⓘ
 ▼

高可用性ポート ⓘ

プロトコル *
 TCP
 UDP

ポート *
 ✓

バックエンドポート * ⓘ
 ✓

正常性プローブ * ⓘ
 ▼

セッション永続化 ⓘ
 ▼

アイドルタイムアウト(分) * ⓘ
 4

TCPリセット
 無効
 有効

フローティング IP ⓘ
 無効
 有効

図 4.5-11 負荷分散規則の作成

正常性プローブは以下の通りに設定します。

項目	設定値
名前	LKL-QWK-probe
プロトコル	TCP
ポート	12345 ³
間隔	5

正常性プローブの追加

i 正常性プローブは、バックエンドプール インスタンスの状態を確認するために使用されます。正常性プローブがバックエンド インスタンスから応答を取得できない場合、正常性プローブが再び成功するまで、そのバックエンド インスタンスに新しい接続は送信されません。

名前 *

 ✓

プロトコル *

 ▼

ポート * ⓘ

 ✓

間隔 * ⓘ

 秒

図 4.5-12 正常性プローブの作成

すべて正常に設定を行うと次ページで示すような表示になります。

³ 本ガイドではポート番号を 12345 で設定します。お客様の環境に合わせてご設定ください。なお、ここで指定したポート番号は LB Health Check リソースの作成時に使用します

名前 *

LKL-QWK-LB-PostgreSQL ✓

IP バージョン *

IPv4

IPv6

フロントエンド IP アドレス * ⓘ

LKL-QWK-LB (10.3.1.200) ▾

バックエンド プール * ⓘ

LKL-LB-backpool ▾

高可用性ポート ⓘ

プロトコル *

TCP

UDP

ポート *

5432 ✓

バックエンド ポート * ⓘ

5432 ✓

正常性プローブ * ⓘ

(新規) LKL-QWK-probe ▾

[新規作成](#)

セッション永続化 ⓘ

なし ▾

アイドルタイムアウト (分) * ⓘ

4

TCP リセット

無効

有効

フローティング IP ⓘ

無効

有効

図 4.5-13 負荷分散規則の確認

以上で Azure の操作はすべて終了になります。

次のセクションでは LifeKeeper をインストールする前に行う OS の設定を行います。

5. OS の設定

5.1. 仮想マシンへのログイン

使用する ssh クライアントについて

サーバー上の GUI を操作するために X Window システムを利用します。そのため、X Window システムがご利用頂ける ssh クライアントをご利用ください。本ガイドでは Tera Term のスクリーンショットで案内しています。

① Public IP アドレスの確認

Azure ポータル 仮想マシンの概要から ssh ログインに使用する Witness ノードの Public IP アドレスを確認します。

ホーム > LKL-QWK-majority >

The screenshot shows the Azure portal interface for a virtual machine named 'clstr-node-witness'. The left sidebar contains navigation options like '概要' (Overview), 'アクティビティログ' (Activity Log), 'アクセス制御 (IAM)' (Access Control (IAM)), 'タグ' (Tags), '問題の診断と解決' (Troubleshooting), '設定' (Settings), 'ネットワーク' (Network), '接続' (Connections), 'ディスク' (Disks), 'サイズ' (Sizes), and 'Microsoft Defender for Cloud'. The main content area shows the '基本' (Basic) tab with the following details:

Property	Value
リソースグループ (移動)	LKL-QWK-majority
状態	実行中
場所	Japan East
サブスクリプション (移動)	LK SUPPORT
サブスクリプション ID	340f215d-7ad3-49ae-a2fd-0ba1cb3c9d6f
オペレーティング システム	Linux (rhel 9.0)
サイズ	Standard B1s (1 vcpu, 1 GiB メモリ)
パブリック IP アドレス	[Redacted]
仮想ネットワーク/サブネット	vnet-LKL-QWK-majority/ZoneAll
DNS 名	未構成
正常性の状態	-

図 5.1-1 仮想マシン概要からパブリック IP の確認

② ssh ログイン

ローカル PC で ssh クライアントを使用し、仮想マシン作成時に作成したアカウントで、仮想マシンの Public IP に ssh ログインします。

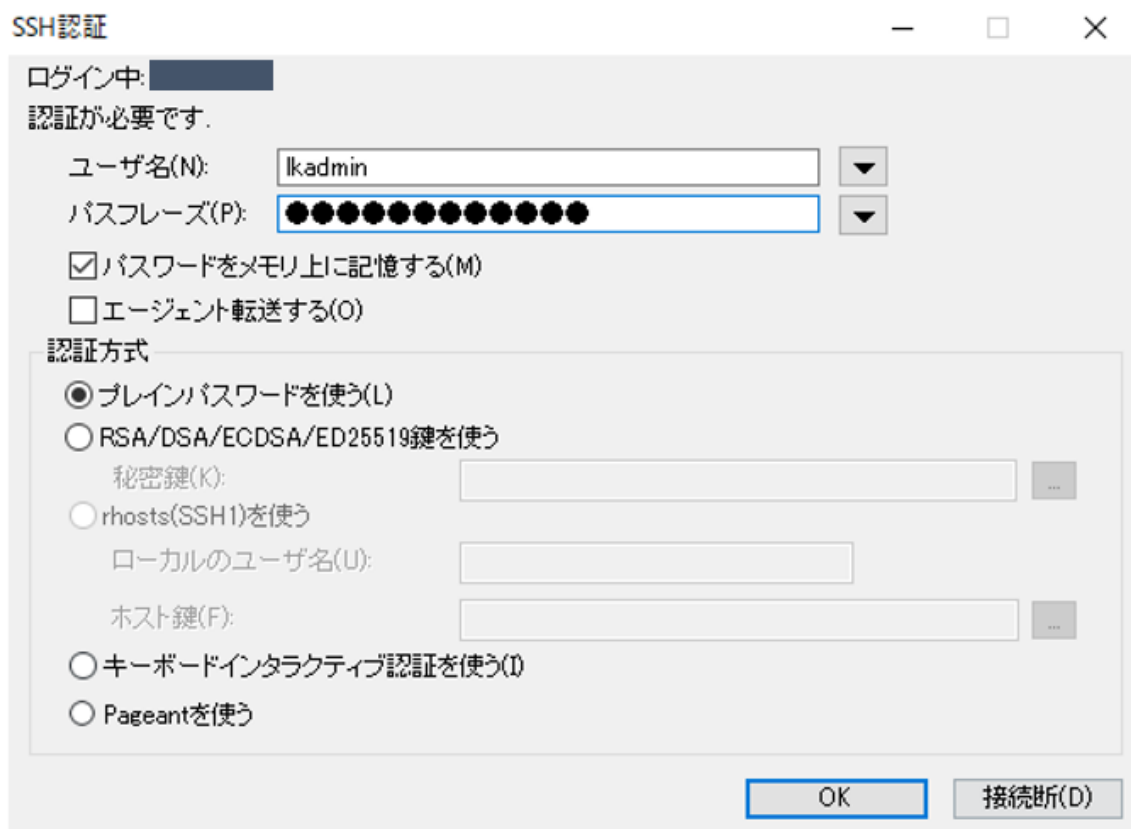


図 5.1-2 ssh クライアントの設定画面

ログインに成功しました。

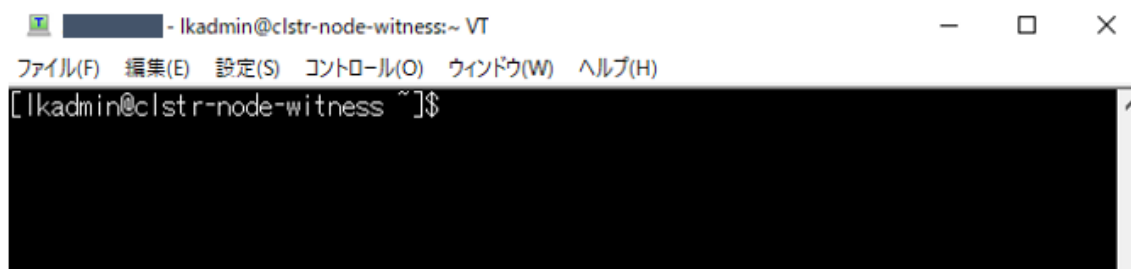


図 5.1-3 ssh ログイン時のプロンプト

5.2. sshd_config の編集

① クラスタースタートへの ssh 接続

クラスタースタートへ正常に ssh 接続ができる事を確認します。

witness サーバー上で以下のコマンドを入力します。

ノード A へ ssh 接続

```
$ ssh 10.3.1.11
```

パスワードの入力が求められるので、仮想マシン作成時に設定したパスワードを入力します。

正常に ssh ログインができれば、以下のコマンドでノード A からログアウトします。

```
$ exit
```

同様の手順でノード B への ssh 接続を行います。

② 公開鍵認証の設定

ノード A、ノード B への ssh が確認出来たら、公開鍵認証の設定を行います。

witness server でキーペアを作成します。

```
$ ssh-keygen -t rsa
```

上記コマンド実行後、/home/lkadmin/.ssh/ 配下に 秘密鍵 id_rsa と、公開鍵 id_rsa.pub が生成されます。

次に、公開鍵をクラスタースタートへ送信します。

ノード A へ公開鍵のコピー。

```
$ ssh-copy-id lkadmin@10.3.1.11
```

同様に、ノード B にも公開鍵をコピーします。

③ パスワード認証と X Server の有効化

パスワード認証の無効化と、X11 Forwarding の有効化を行います。

/etc/ssh/sshd_config をお好きなエディターで開き、以下のように変更を行います。

公開鍵認証の有効化 (クラスターノードのみ)

変更前	変更後
PasswordAuthentication yes	PasswordAuthentication no

X11 Forwarding の有効化 (すべてのノードで実行)

変更前	変更後
X11Forwarding no	X11Forwarding yes

最後に、すべてのノードで sshd サービスを再起動します。

```
# systemctl restart sshd
```

X Server の有効化手順以降は、Witness サーバーからクラスターノードへの ssh 時に-X オプションを付けることで、クラスターノード上の GUI 画面をローカル PC 上で表示する事ができます。

例

```
# ssh -X lkadmin@10.3.1.11
```

本ガイドでは LifeKeeper の構築に GUI を使用しているため、以降クラスターノードの操作のために ssh を実行する際は X オプションを付けてください。

5.3. NIC アドレスの固定化

Azure インフラストラクチャ更新に伴って NIC アドレスが更新されることを防ぐため、NIC アドレスを固定します。全てのノードで、以下の設定を行います。

① インターフェース情報の確認

仮想マシンの NIC 情報を表示し、各インターフェース名と MAC アドレスを控えます。

```
[lkadmin@clstr-node-A ~]$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.3.1.11  netmask 255.255.255.0  broadcast 10.3.1.255
    inet6 fe80::222:48ff:fe68:e144  prefixlen 64  scopeid 0x20<link>
    ether 00:22:48:68:e1:44  txqueuelen 1000  (Ethernet)
    RX packets 4580  bytes 805273 (786.3 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 5800  bytes 1065095 (1.0 MiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.3.2.11  netmask 255.255.255.0  broadcast 10.3.2.255
    inet6 fe80::20d:3aff:fecc:c6b4  prefixlen 64  scopeid 0x20<link>
    ether 00:0d:3a:cc:c6:b4  txqueuelen 1000  (Ethernet)
    RX packets 1  bytes 381 (381.0 B)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 439  bytes 21530 (21.0 KiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

② 設定ファイルの編集

デフォルトでは eth0 のネットワーク設定ファイルしかないため、eth1 のネットワーク設定ファイルを作成します。

```
# cd /etc/sysconfig/network-scripts/  
# cp ifcfg-eth0 ifcfg-eth1
```

eth0、eth1 それぞれの設定ファイルを編集します。HWADDR には、前の手順で控えた値を使用します。

eth0 の例

```
# Created by cloud-init on instance boot automatically, do not edit.  
#  
AUTOCONNECT_PRIORITY=999  
BOOTPROTO=dhcp  
DEVICE=eth0  
HWADDR=00:22:48:68:e1:44  
ONBOOT=yes  
TYPE=Ethernet  
USERCTL=no
```

eth1 の例

```
# Created by cloud-init on instance boot automatically, do not edit.  
#  
AUTOCONNECT_PRIORITY=999  
BOOTPROTO=dhcp  
DEVICE=eth1  
HWADDR=00:0d:3a:cc:c6:b4  
ONBOOT=yes  
TYPE=Ethernet  
USERCTL=no
```

5.4. SELinux の無効化

SELinux が enforcing モードの場合 LifeKeeper は動作しません。

また、必要な場合を除いて permissive モードは推奨されません。

そのため、全ノード上で以下のコマンドを実行し SELinux を無効化させます

```
# sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
```

変更は再起動後に適用されます。

LifeKeeper 上で SELinux を Permissive モードで動作させる事を検討している場合は以下のページをご参照ください。

LifeKeeper for Linux v9.7.0 テクニカルドキュメンテーション - インストール前の要件

<https://docs.us.sios.com/spslinux/9.7.0/ja/topic/installing-the-software>

5.5. ファイアウォールの無効化

このセクションでは、SIOS Protection Suite for Linux をインストールするために、firewalld サービスを無効にします。

ファイアウォールは有効なままにしておくことができますが、SIOS Protection Suite と保護するアプリケーションに必要なポートを設定する必要があります（ここでは示していません）。

以下のコマンドを全ノード上で実行し、ファイアウォールを無効化させます。

```
# systemctl disable firewalld.service --now
```

5.6. 名前解決

ノード間の名前解決のために /etc/hosts ファイルにホスト名と IP アドレスを追記します。

好きなエディターで /etc/hosts を開き、以下を追加します。

```
10.3.1.11 clstr-node-A
10.3.1.12 clstr-node-B
10.3.1.50 clstr-node-witness
```

6. ローカルリポジトリの設定

LifeKeeper のインストール時に、各 OS のリポジトリから依存関係にあるソフトウェアのインストールも同時に行います。

Azure 上の Red Hat 仮想マシンは Azure の RHUI をリポジトリとして利用することができますが、オフラインノード上からは Azure の RHUI にアクセスできません。

そのため、OS のインストールイメージを使用しローカルリポジトリを設定することで、オフラインノード上でも LifeKeeper のインストールを行うことができます。

6.1. /home の拡張

Azure では RHEL 仮想マシンを作成後に明示的に/home の拡張を行う必要があります。

すべてのノード上で以下のコマンドを実行してください。

```
$ sudo su -  
# cd /  
# umount /home  
# lvextend -L +16384 /dev/mapper/rootvg-homelv  
# mount /home  
# xfs_growfs /dev/mapper/rootvg-homelv
```

6.2. OS イメージの転送

次に、RHEL のインストールイメージをローカル PC からすべてのノードに転送します。

今回、仮想マシンのイメージに使用した OS は Red Hat Enterprise Linux 9.0 のため、Red Hat 公式から rhel-baseos-9.0-x86_64-dvd.iso を事前にダウンロードしておき、全ノードに scp 等で転送します。

Witness ノードへの転送 (SCP を利用する場合)

```
> scp rhel-baseos-9.0-x86_64-dvd.iso lkadmin@<Public_IP>:~
```


Witness ノードからクラスターノードへの転送

```
$ scp rhel-baseos-9.0-x86_64-dvd.iso lkadmin@10.3.1.11:~
```

```
$ scp rhel-baseos-9.0-x86_64-dvd.iso lkadmin@10.3.1.12:~
```

6.3. fstab の編集

fstab を編集しておくことで、毎回 OS 起動時に OS イメージのマウントが自動で実行され、ローカルリポジトリがいつでも使えます。

マウントポイントを作成しておきます。

```
# mkdir /media/cdrom
```

/etc/fstab をお好きなエディターで開き、以下を追加します。

```
/home/lkadmin/rhel-baseos-9.0-x86_64-dvd.iso /media/cdrom iso9660 ro,loop  
0 0
```

マウントを行います。

```
# mount /media/cdrom
```

正常に実行されたら、fstab の編集は終了になります。

6.4. ローカルリポジトリの作成

ローカルリポジトリを以下の設定で `/etc/yum.repos.d/rhel-dvd.repo` に作成します。

```
[rhel-dvd-BaseOS]
name=Red Hat Enterprise Linux 9.0 - x86_64 - DVD
baseurl=file:///media/cdrom/BaseOS/
enabled=1
gpgcheck=1
gpgkey=file:///media/cdrom/RPM-GPG-KEY-redhat-release

[rhel-dvd-AppStream]
name=Red Hat Enterprise Linux 9.0 - x86_64 - DVD
baseurl=file:///media/cdrom/AppStream/
enabled=1
gpgcheck=1
gpgkey=file:///media/cdrom/RPM-GPG-KEY-redhat-release
```

6.5. rh-cloud-base.repo の無効化

ローカルリポジトリの設定が終了したため、Azure の RHUI を以下のコマンドを使用し、無効にします。

```
# sed -i 's/enabled=1/enabled=0/g' /etc/yum.repos.d/rh-cloud-base.repo
```

6.6. ローカルリポジトリの確認

ローカルリポジトリが正しく設定されているか確認します。

以下のコマンドでパッケージマネージャーが使用しているリポジトリのリストを確認できます。正常にローカルリポジトリの設定が完了していると、以下のように表示されます。

```
[lkadmin@clstr-node-A ~]$ yum repolist
repo id                repo name
rhel-dvd-AppStream     Red Hat Enterprise Linux 9.0 - x86_64 - DVD
rhel-dvd-BaseOS        Red Hat Enterprise Linux 9.0 - x86_64 - DVD
```

6.7. GUI 設定

全ノード上で GUI の SSH 転送に必要なソフトウェアのインストールを行います。

以下のコマンドを実行し、X11 関連パッケージをインストールしておきます。

```
# dnf install xhost xauth
```

また、LifeKeeper の管理を root ユーザーで行うため、/root/.Xauthority に GUI が使用する DISPLAY 変数を格納する必要があります。

そのため、全ノード上で以下のコマンドを実行します。

```
$ echo "sudo xauth add ¥$(xauth -f /home/lkadmin/.Xauthority list|tail -1)" >> .bashrc
```

ここまで終了したら、LifeKeeper のインストールの前にすべてのノードを再起動します。

7. LifeKeeper のインストール

ここでは LifeKeeper のインストール方法を説明します。

詳しくは LifeKeeper for Linux v9.7.0 スタートアップガイドを参照してください。

7.1. LifeKeeper インストールイメージの転送

次に、LifeKeeper のインストールイメージをローカル PC からすべてのノードに転送します。

Witness ノードへの転送 (SCP を利用する場合)

```
> scp LKL_V970_032823.iso lkadmin@<Public_IP>:~
```

Witness ノードからクラスターノードへの転送

```
$ scp LKL_V970_032823.iso lkadmin@10.3.1.11:~
```

```
$ scp LKL_V970_032823.iso lkadmin@10.3.1.12:~
```

7.2. root パスワードの変更

LifeKeeper の GUI 表示には root パスワードが必要となるため、ルートパスワード の設定を行います。なお、クラスター全体で共通のパスワードにする事で、GUI のログインを 1 回で済ませる事ができます。

```
#su -  
# passwd  
Changing password for user root.  
New password:<新しいパスワード>  
Retype new password:<新しいパスワード(確認)>  
passwd: all authentication tokens updated successfully.
```

7.3. Witness ノードへの LifeKeeper インストール

① LifeKeeper インストールイメージの展開とマウント

```
# mkdir /mnt/iso  
# mkdir /mnt/img  
# mount ~/LKL_V970_032823.iso /mnt/iso/  
# mount /mnt/iso/sps_970.img /mnt/img/
```

② LifeKeeper インストールスクリプトの実行

```
#su -  
# /mnt/img/setup
```

セットアップスクリプトを実行するとターミナルに TUI が表示されます。

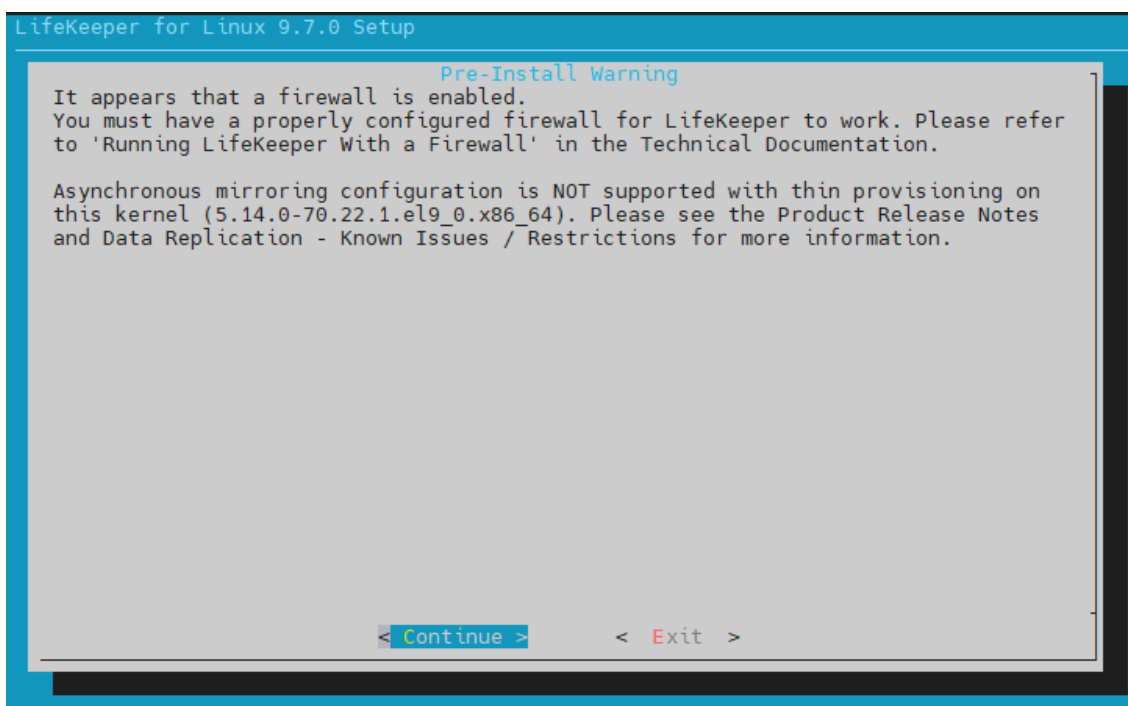


図 7.3-1 setup 実行画面

画像は Red Hat Enterprise Linux 9.0 では非同期モードのレプリケーションが非サポートであることを伝えています。

また、ファイアウォールについての警告は無視していただいて構いません。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure Step by step Guide

メニューは以下のキーで操作します。

↑ ↓ : 選択項目の移動

← → : 最下行ボタンの移動

ENTER : サブメニューを開く

Y / N / SPACE : 選択項目の ON / OFF / 反転

Continue を選択し、Enter で次の画面に移ります。

Use Quorum / Witness Functions にチェックを入れます。

Witness ノードは Quorum/Witness 機能を使用するため、こちらにチェックを入れてください。

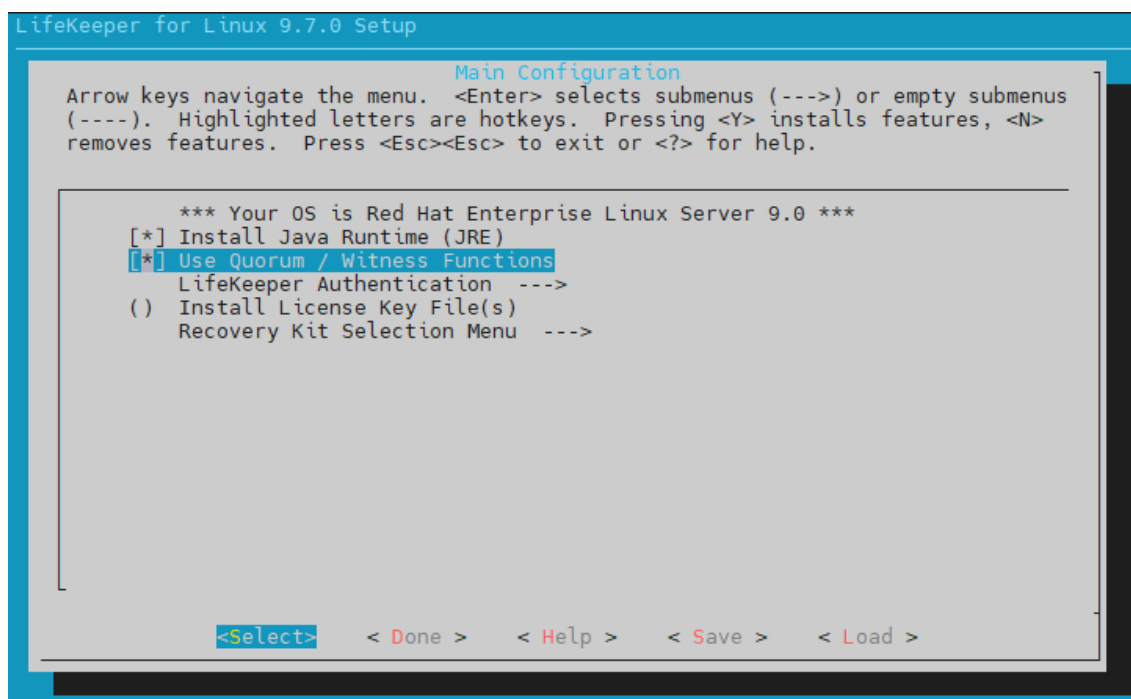


図 7.3-2 Quorum / Witness Functions を選択

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

Done を選択し、インストールを開始します。

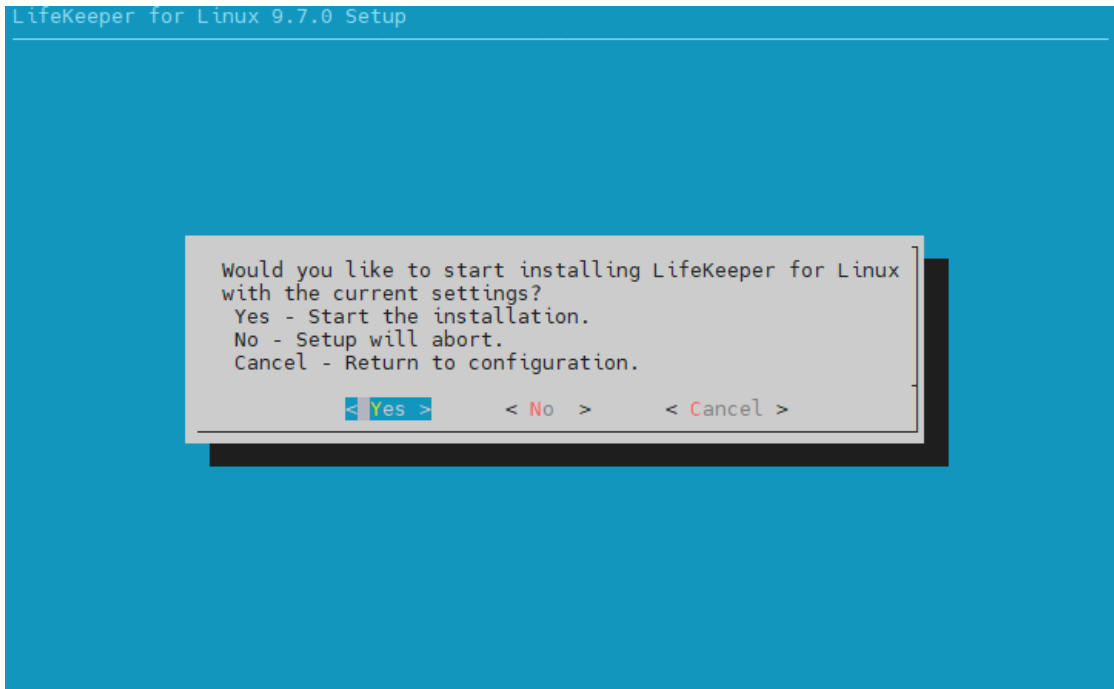


図 7.3-3 LifeKeeper のインストール

③ License キーのインストール

ライセンスキーファイルのパスを lkkeyins に渡すことで、LifeKeeper をアクティベートする事ができます。

```
# /opt/LifeKeeper/bin/lkkeyins <ライセンスキーファイルのパス>
```

④ ブロードキャスト Ping の無効化

Azure では IP リソースが保護する仮想 IP アドレスを Azure の仮想ネットワークで認識する事が出来ません。そのため /etc/default/LifeKeeper で以下のパラメータを変更する必要があります。

変更前	変更後
NOBCASTPING=0	NOBCASTPING=1

⑤ LifeKeeper の起動

```
# /opt/LifeKeeper/bin/lkstart
```

以上で Witness ノードへの LifeKeeper のインストールは終了になります。

クラスターノードへの LifeKeeper インストール

Witness ノードへのインストールとほぼ同様の手順になりますが、インストール時に追加で LB Health Check Kit, DataKeeper, PostgreSQL Recovery Kit を選択する必要があります。

1. LifeKeeper インストールイメージの展開とマウント

```
#mkdir /mnt/iso
#mkdir /mnt/img
#mount ~/LKL_V970_032823.iso /mnt/iso/
#mount /mnt/iso/sps_970.img /mnt/img/
```

2. LifeKeeper インストールスクリプトの実行

```
#su -
# /mnt/img/setup
```

Use Quorum / Witness Functions にチェックを入れます。

クラスターノードも Quorum/Witness 機能を使用するため、こちらにチェックを入れてください。

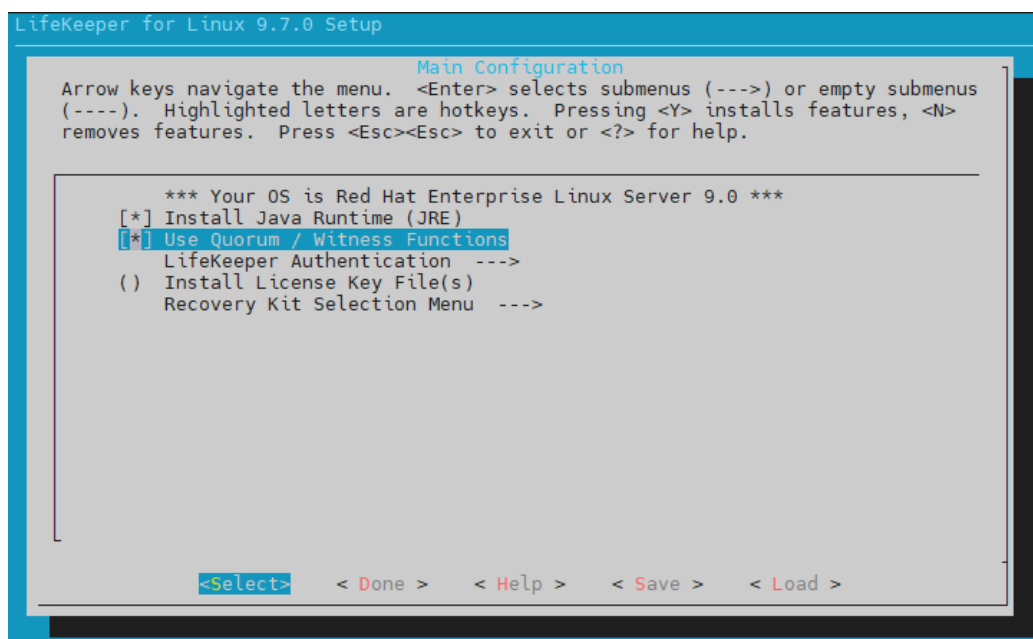


図 7.4-1 Quorum / Witness Functions を選択

Recovery Kit Selection Menu に移動します。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

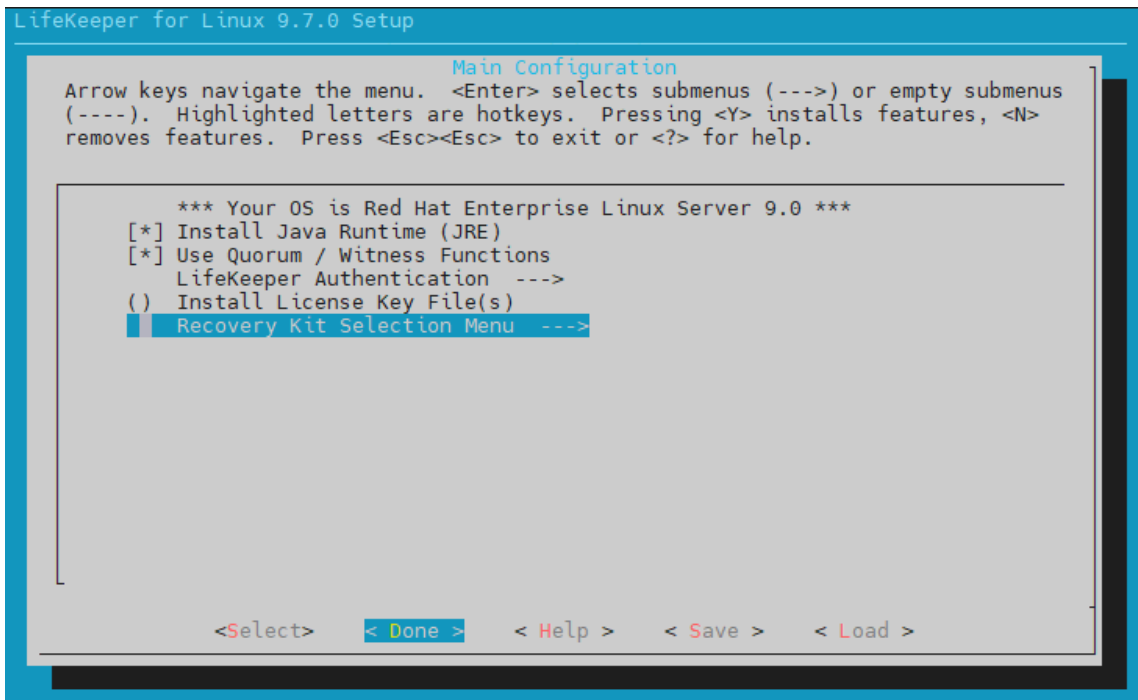


図 7.4-2 Recovery Kit Selection Menu を選択

こちらが、各リカバリーキットのインストール選択画面になります。

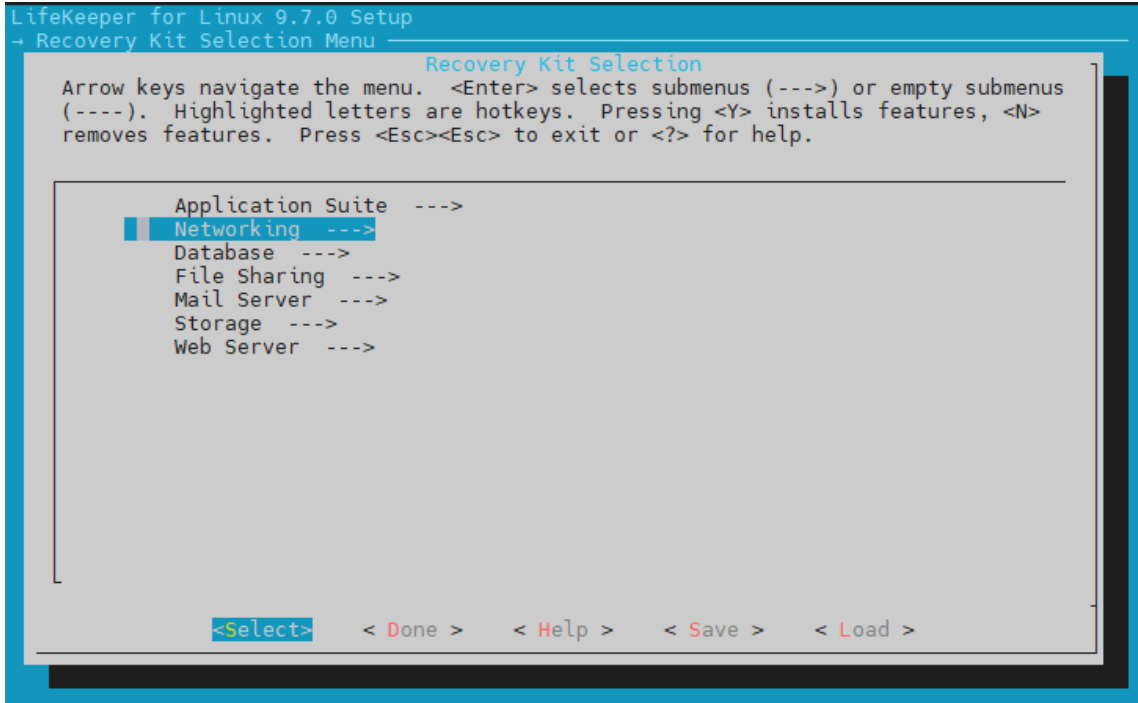


図 7.4-3 リカバリーキットのインストール選択画面

LB Health Check Kit のインストール

Networking → LifeKeeper LB Health Check Kit の順にチェック

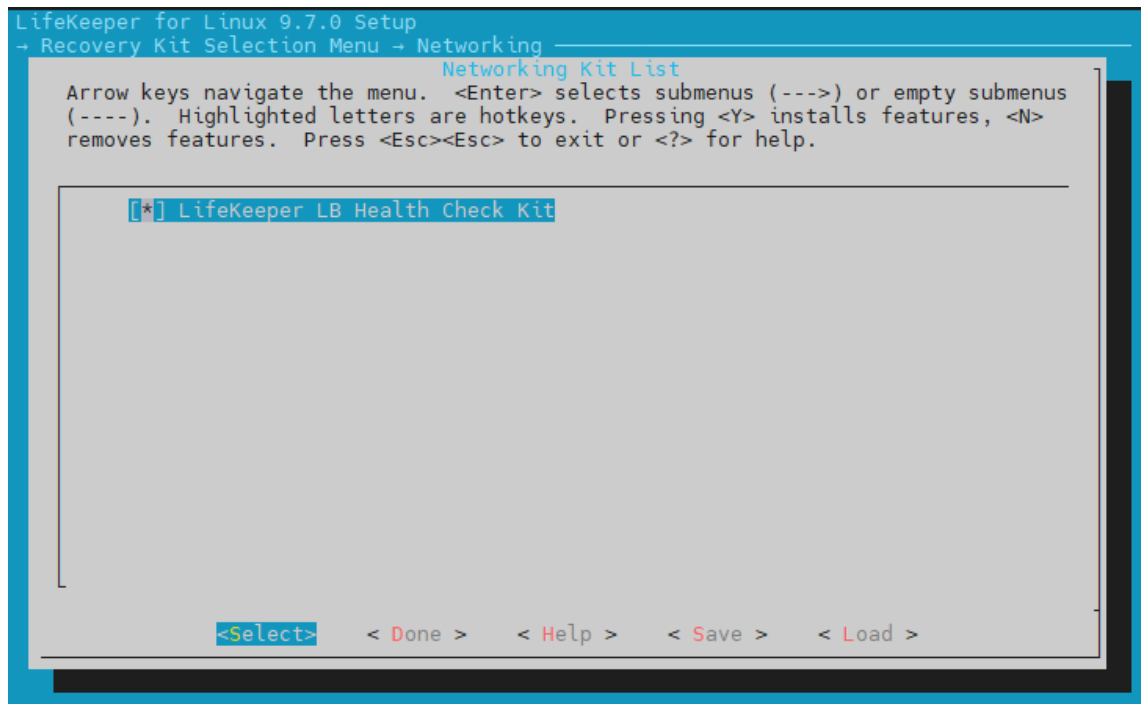


図 7.4-4 LB Health Check Kit を選択

PostgreSQL Recovery Kit のインストール

Database → LifeKeeper PostgreSQL RDBMS Recovery Kit の順にチェック

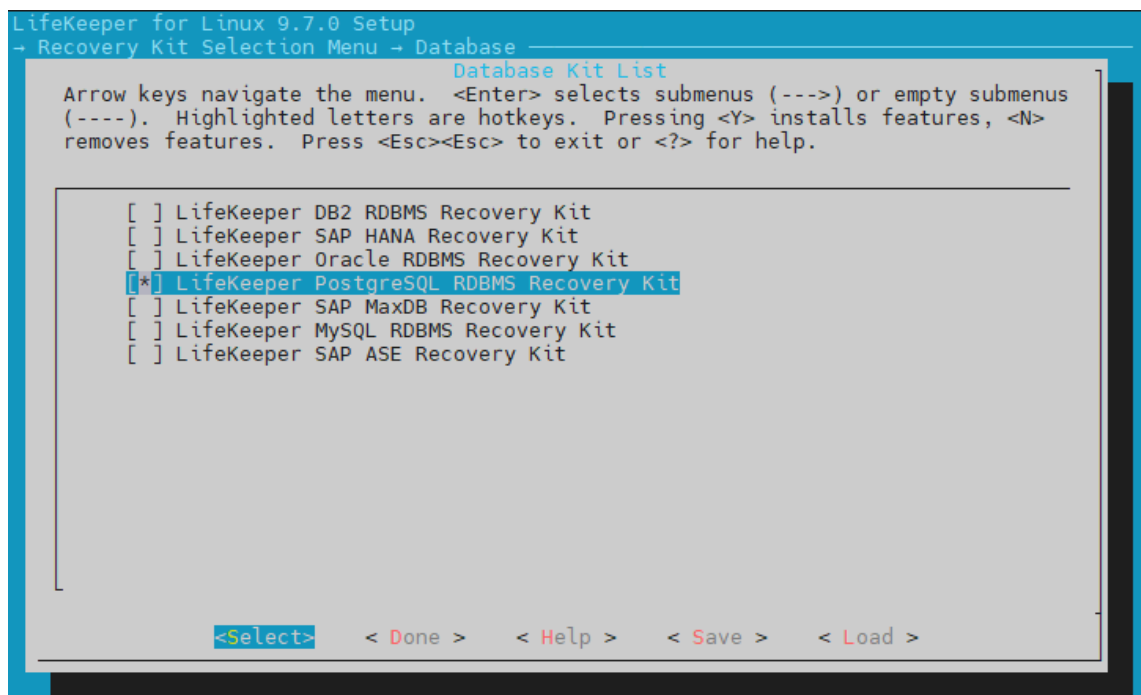


図 7.4-5 PostgreSQL RDBMS Recovery Ki を選択

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

DataKeeper

Storage → DataKeeper for Linux にチェック

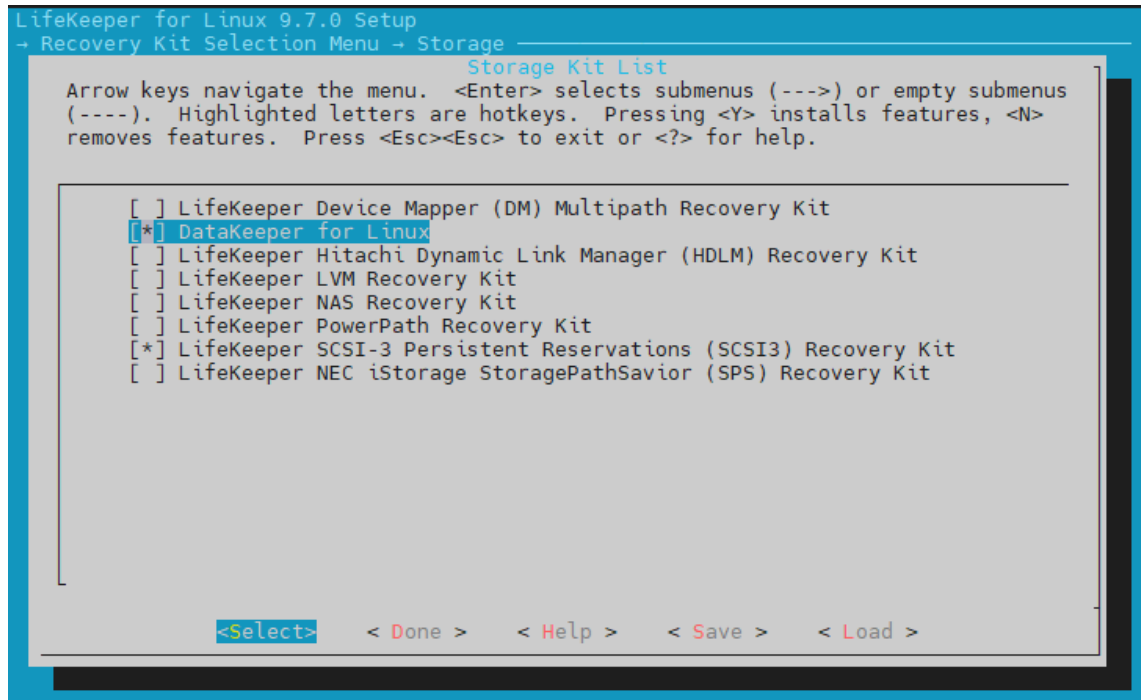


図 7.4-6 DataKeeper for Linux を選択

Done を選択し、インストールを開始します。

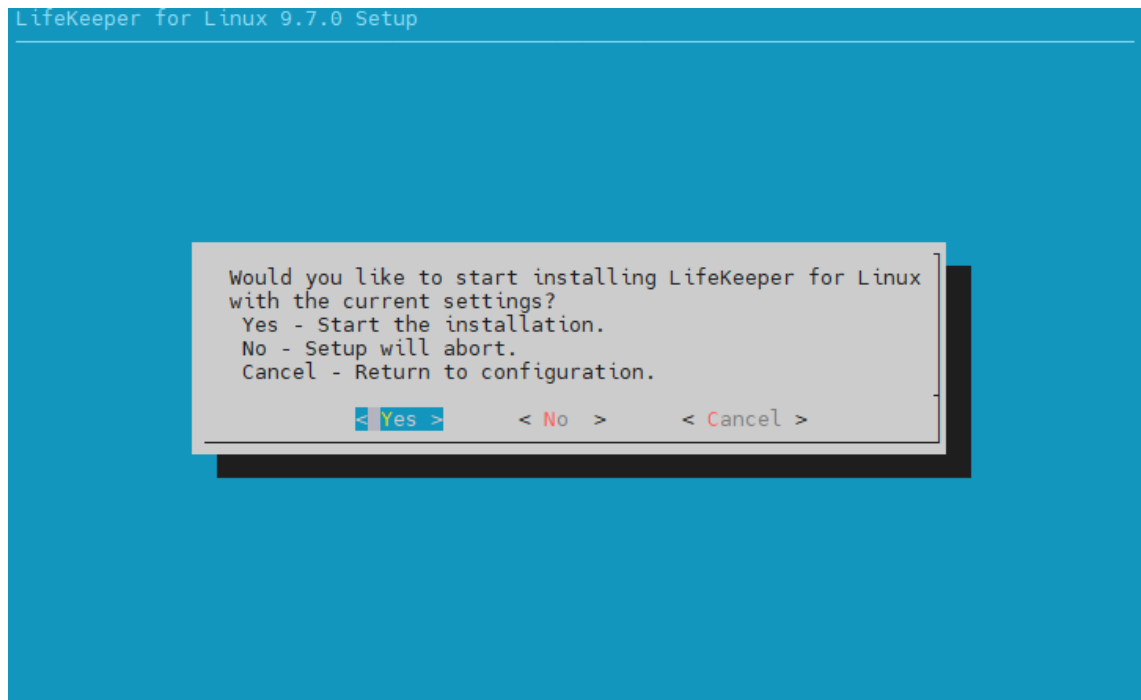


図 7.4-7 インストールの開始

3. License キーのインストール

```
# /opt/LifeKeeper/bin/lkkeyins <ライセンスキーのパス>
```

4. ブロードキャスト Ping の無効化

Azure では IP リソースが保護する仮想 IP アドレスを Azure の仮想ネットワークで認識する事が出来ません。そのため /etc/default/LifeKeeper で以下のパラメータを変更する必要があります。

変更前	変更後
NOBCASTPING=0	NOBCASTPING=1

5. LifeKeeper の起動

```
# /opt/LifeKeeper/bin/lkstart
```

以上でクラスターノードへの LifeKeeper のインストールも終了になります。

補足

LifeKeeper の管理に使用するコマンドの実行ファイルは /opt/LifeKeeper/bin に LifeKeeper の管理コマンドのレファレンスマニュアルは /opt/LifeKeeper/man にそれぞれ保存されています。

そのため、以下のコマンドを実行することで、次回以降の root ログイン以降、絶対パスでコマンドを実行する必要がなくなります。

```
echo "PATH=$PATH:/opt/LifeKeeper/bin" >> /root/.bash_profile
echo "MANPATH=$MANPATH:/opt/LifeKeeper/man" >> /root/.bash_profile
echo "export PATH MANPATH" >> /root/.bash_profile
```

8. コミュニケーションパスの作成

各ノードを LifeKeeper のコミュニケーションパスでつなぎ、クラスターを構築します。

8.1. GUI を使用したコミュニケーションパスの作成

① GUI の起動

Witness ノード上で実行します。

lkGUIapp コマンドを実行する事で、LifeKeeper GUI を表示させる事ができます。

```
# lkGUIapp &
```

② ログイン

root ユーザーと先ほどの手順で設定したパスワードを使用しログインを行います。

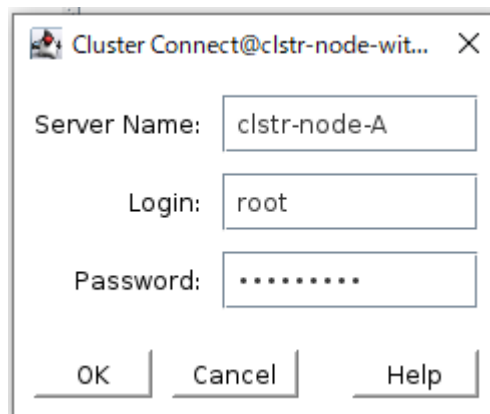


図 8.1-1 LifeKeeper GUI ログイン画面

ログインに成功すると次のように GUI が表示されます。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

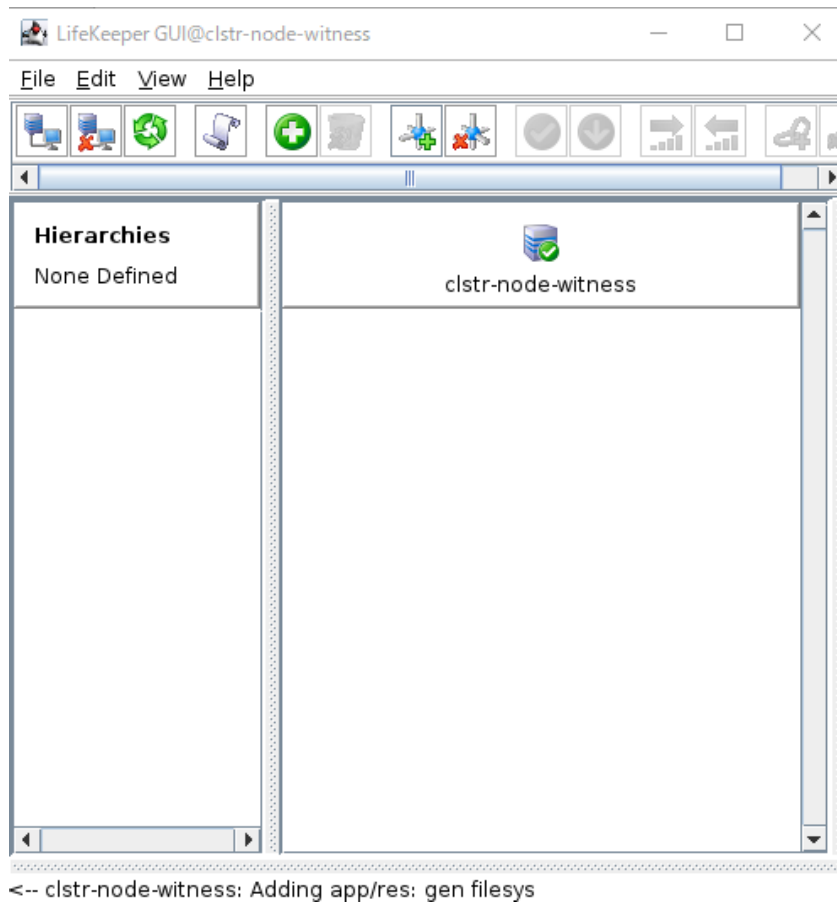


図 8.1-2 ログイン後の LifeKeeper GUI

③ コミュニケーションパスの作成

赤く囲ったボタンを押下します。

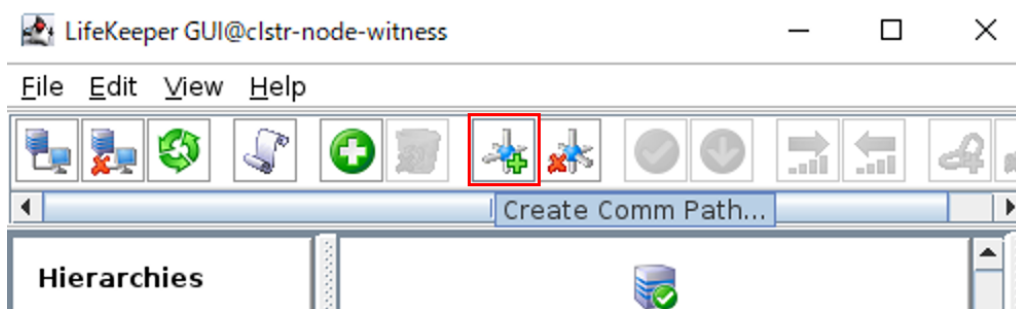


図 8.1-2 コミュニケーション作成ボタン

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

ウィザードに従い、コミュニケーションパスを作成します。

項目	設定値
LocalServer	clstr-node-witness
RemoteSeerver	clstr-node-A と入力し、Add
Device Type	TCP
Local IP Address(es)	10.3.1.50
Remote IP Address	10.3.1.11
Priority	1

正常に追加されると、GUI に次のように表示されます。

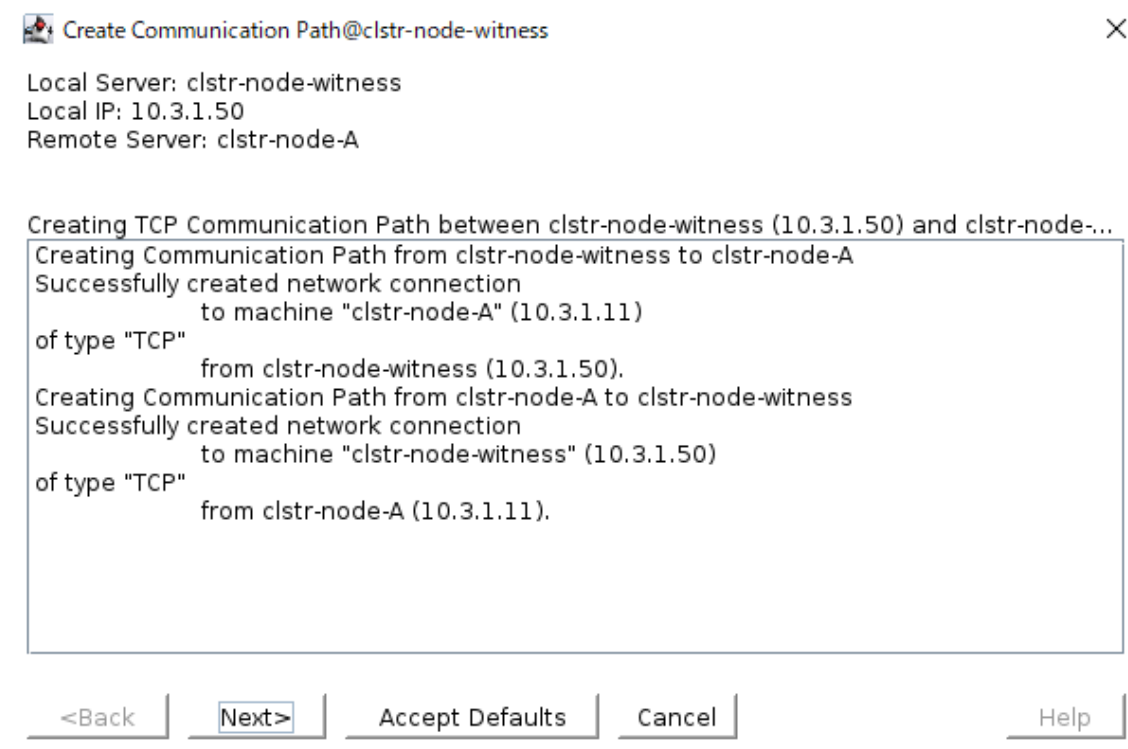


図 8.1-3 コミュニケーションパスの作成ログ

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

上記の手順で下記表のコミュニケーションパスも追加ですべて作成します。

witness ノード - ノード B 間

項目	設定値
LocalServer	clstr-node-witness
RemoteSeerver	clstr-node-B と入力し、Add
Device Type	TCP
Local IP Address(es)	10.3.1.50
Remote IP Address	10.3.1.12
Priority	1

ノード A - ノード B (10.3.1.0/24 セグメント) ノード A 上で作成

項目	設定値
LocalServer	clstr-node-A
RemoteSeerver	clstr-node-B と入力し、Add
Device Type	TCP
Local IP Address(es)	10.3.1.11
Remote IP Address	10.3.1.11
Priority	1

ノード A - ノード B (10.3.2.0/24 セグメント) ノード A 上で作成します

項目	設定値
LocalServer	clstr-node-A
RemoteSeerver	clstr-node-B
Device Type	TCP
Local IP Address(es)	10.3.2.11
Remote IP Address	10.3.2.12
Priority	10

8.2. GUI クライアントのステータス情報

コミュニケーションパスの作成が終了すると、次のようなアイコンが GUI 上に表示されま
す。

それぞれのアイコンがすべて ALIVE である事をご確認ください。

■ サーバー状態の情報



全てのコミュニケーションパスのステータスが ALIVE となっている。



一部のコミュニケーションパスのステータスが DEAD となっている。



全てのコミュニケーションパスのステータスが DEAD となっている。



サーバーのステータスが不明である。

また、これから作成するリソースの状態は以下のようなアイコンで表示されます。

こちらも合わせてご確認ください。

■ リソース状態の情報



Active - リソースの稼働状態 (ISP ステータス※)



Standby - リソースの停止状態 (OSU ステータス※)



Fail - リソースの障害状態 (OSF ステータス※)



UNKNOWN - リソースの状態が確認できない状態

※ CLI 上でリソース状態を表示したときのステータス

9. 各種リソースの作成

実際に LifeKeeper で保護するリソースの作成と設定を行います。

本ガイドでは PostgreSQL データベースのデータを、DataKeeper で保護し、PostgreSQL へのアクセスは仮想 IP アドレス 10.3.1.200 を通して行われます。

そのため、システムの全体図は次の画像のような構成になります。

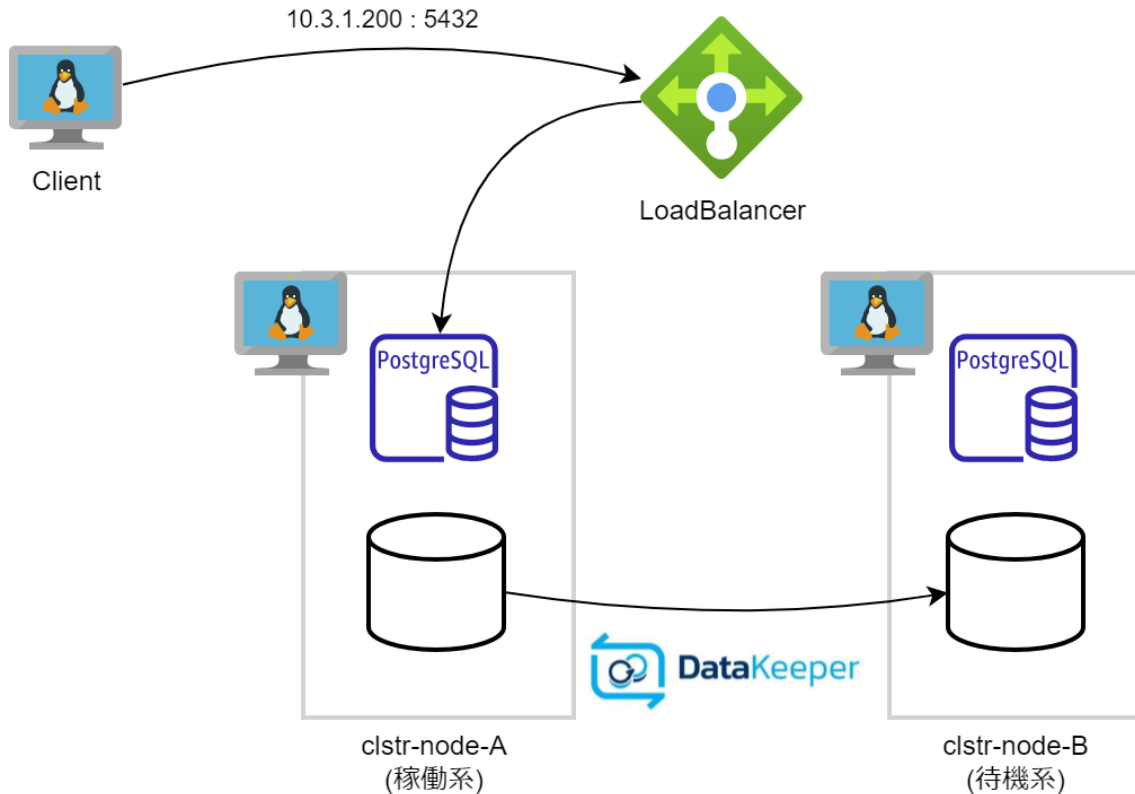


図 9-1 サービスのイメージ図

クライアントから仮想 IP アドレスへのアクセスが行われると、Load Balancer は現在稼働しているノードへパケットを転送します。

また、DataKeeper は稼働系のディスクへの書き込みが行われた場合、同様の内容を待機系に接続されたディスクに書き込みます。

そのため、フェイルオーバーが発生した場合もフェイルオーバー直前のデータを使用し PostgreSQL サーバーの稼働を継続する事ができます。

9.1. リソースの作成前に

各リソースの作成後、必ず作成したリソースのスイッチオーバーテストを実施ください。
すべてのリソースについて、共通の方法で実施することができます。

手順は以下の通りです。

1. GUI から待機サーバー上で稼働を移したいリソースを右クリックします。

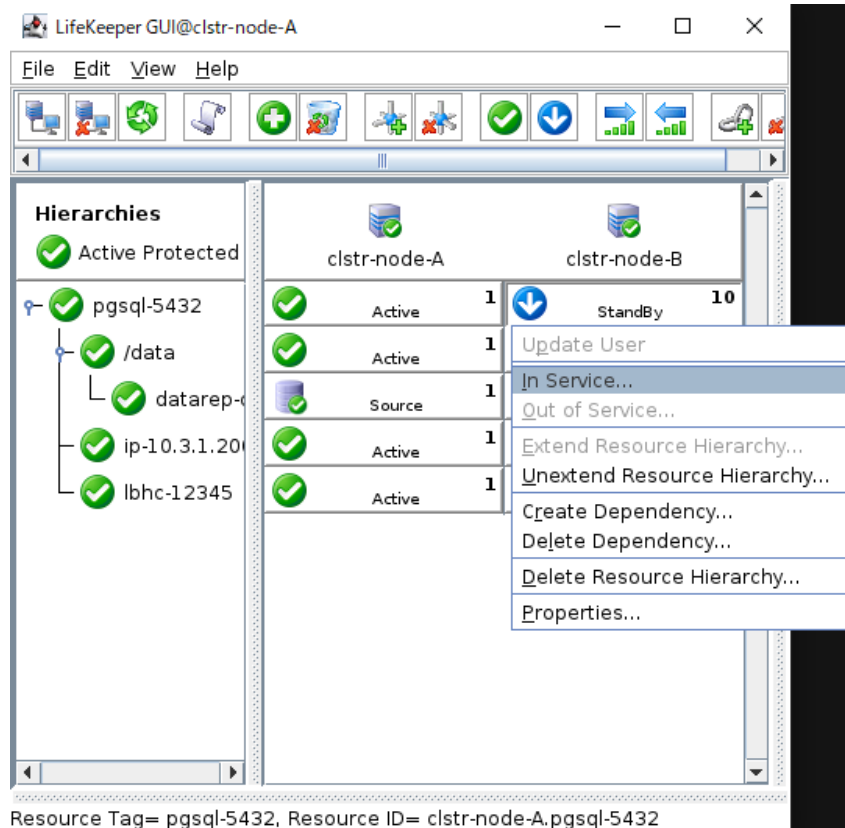


図 9.1-1 リソースのコンテキストメニュー

2. In Service をクリックします。
3. ダイアログに従い、ウィザードを進めます。

Recovery Kit により、ウィザードは異なるため、適宜ウィザードを読み進めて下さい。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

4. 次の画像のように Put “リソース名” in-service successful と表示されれば成功です。

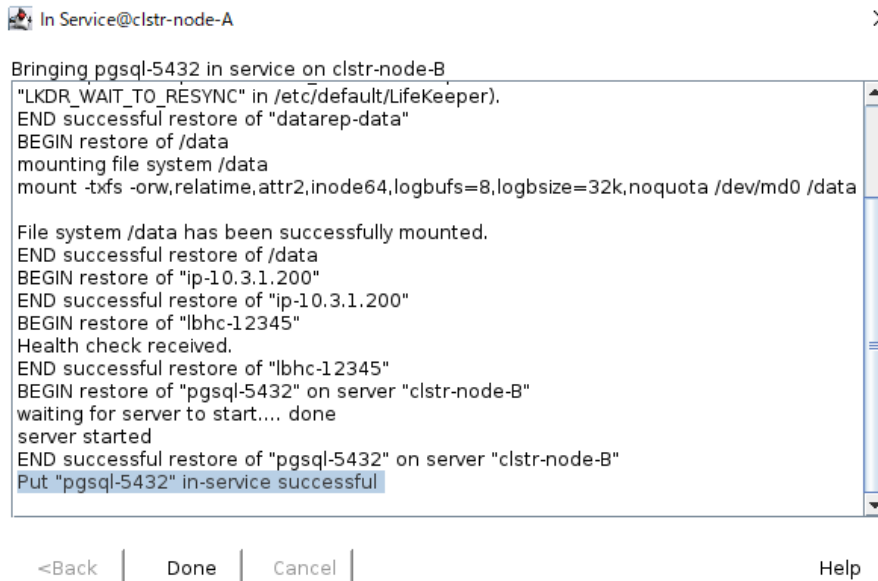


図 9.1-2 スイッチオーバーの終了

スイッチオーバーに成功すると、GUI の表記も、元待機ノード (clstr-node-B) が稼働系として表示されます。

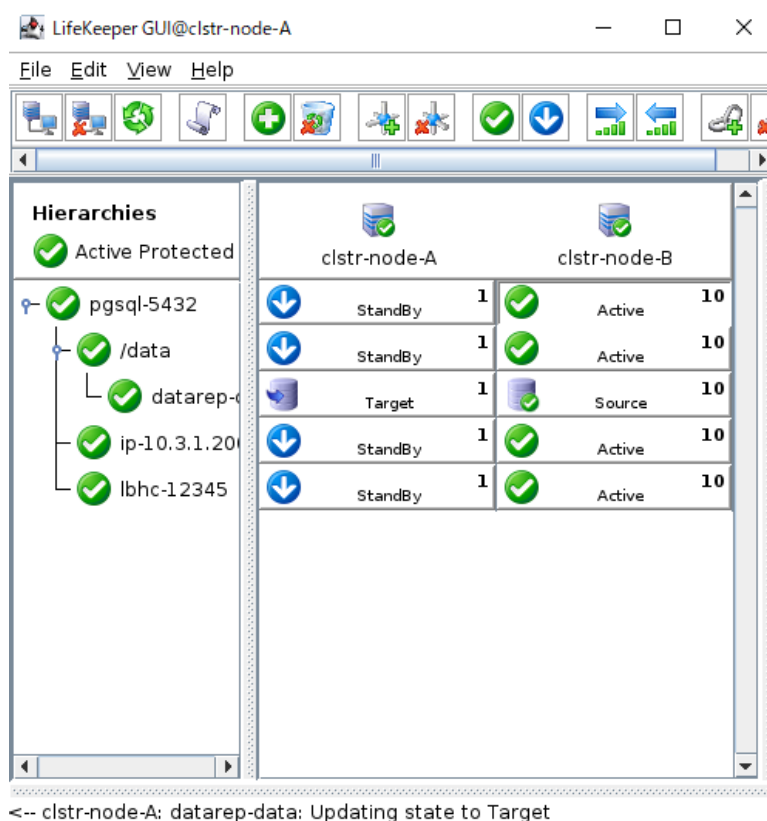


図 9.1-3 スイッチオーバー終了後の LifeKeeper GUI

以上でスイッチオーバーテストは終了となります。

9.2. DataKeeper リソースの作成

Azure 仮想マシンを作成した際に追加したディスクを DataKeeper の保護ディスクとして設定します。

① リソース作成ウィザードの起動

リソースの作成は GUI のサーバーコンテキストツールバーから Create Resource Hierarchy を選択し、リソース作成ウィザードを起動します。

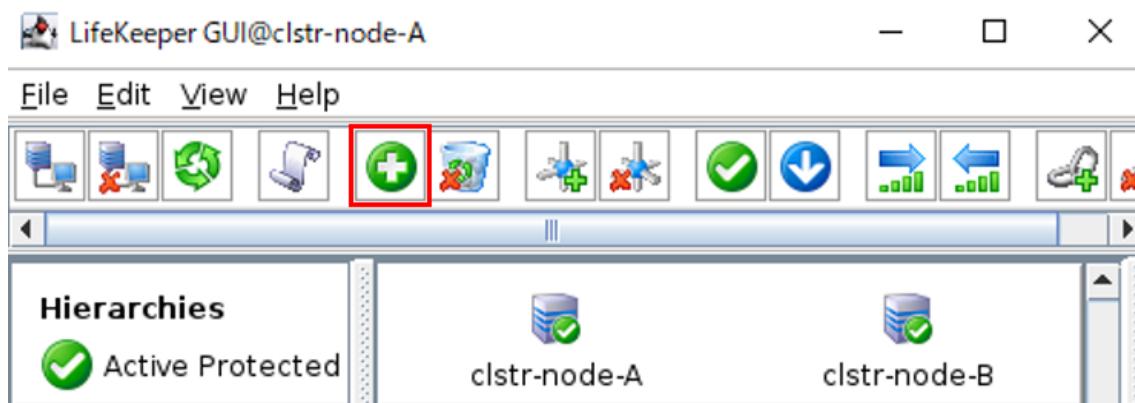


図 9.2-1 リソースの作成ボタン

② Recovery Kit として Data Replication を選択



図 9.2-2 DataReplication を選択

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

③ リソース作成ウィザードで、以下の値を入力

項目	設定値
Switchback Type	Intelligent
Server	clstr-node-A
Hierarchy Type	Replicate New Filesystem
Source Disk	/dev/sdc (16.0 GB) ⁴
New Mount Point	/data
New Filesystem Type	xfs
Data Replication Resource Tag	datarep-data
File System Resource Tag	/data
Bitmap File	/opt/LifeKeeper/bitmap_data ⁵
Enable Asynchronous Replication ?	no

入力が完了すると、確認画面に移ります。

Create をクリックしてください。

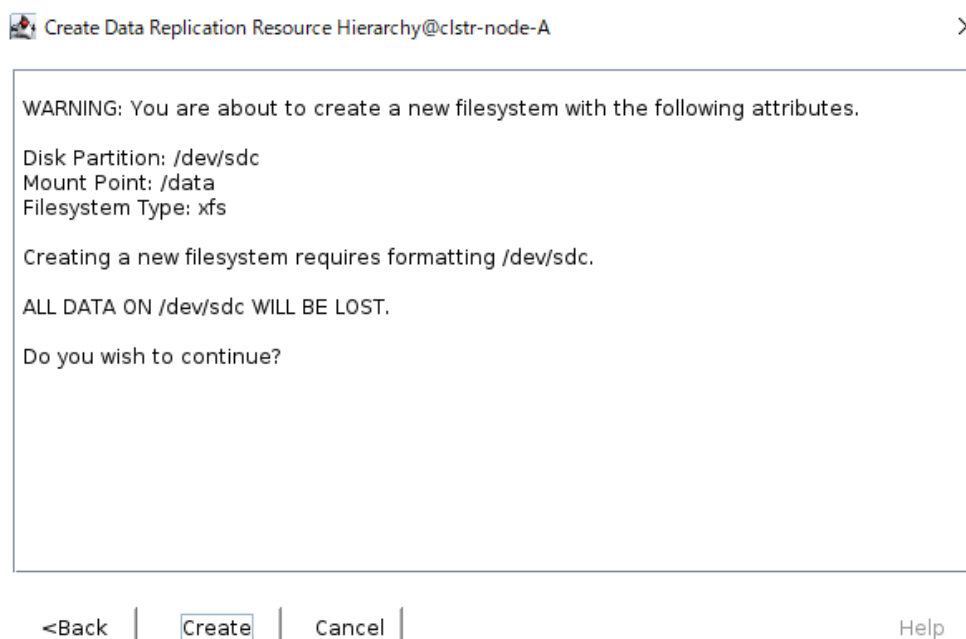


図 9.2-3 DataReplication リソースの新規作成画面

⁴ クラウド環境では追加のディスクが別の名前で認識される事がございます。お客様環境に合わせてご利用ください。

⁵ ウィザードの初期値になります。

successful の文字が表示されている事を確認します。

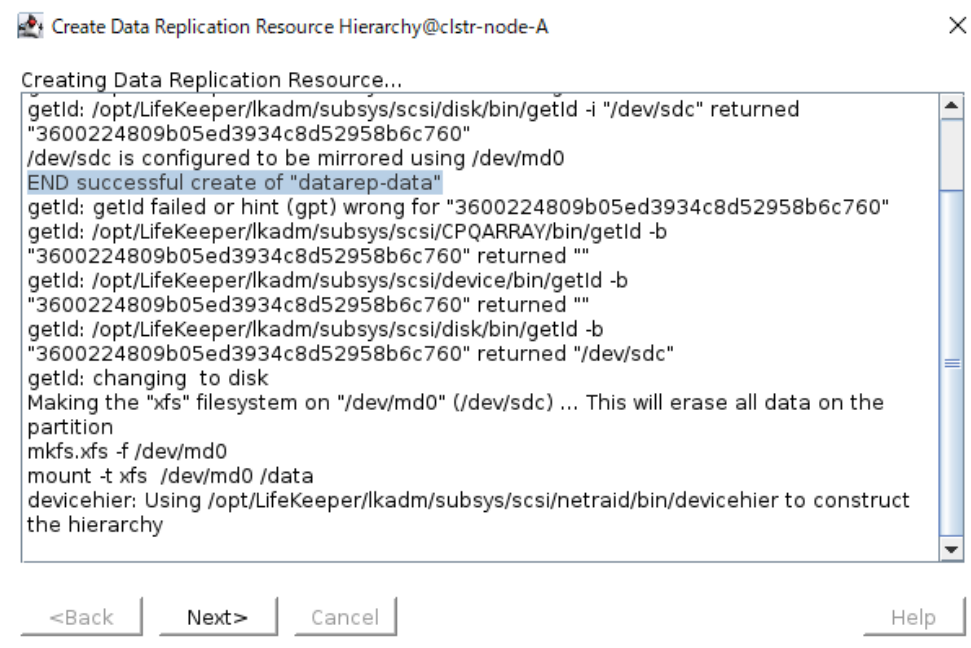


図 9.2-4 DataReplication リソースの作成終了

Next をクリックすると、自動的に Pre-Extend ウィザードに移ります。

④ Pre Extend ウィザードの起動

クラスターノードへリソースの拡張をおこないます。

Target Server に clstr-node-B を選択してください。

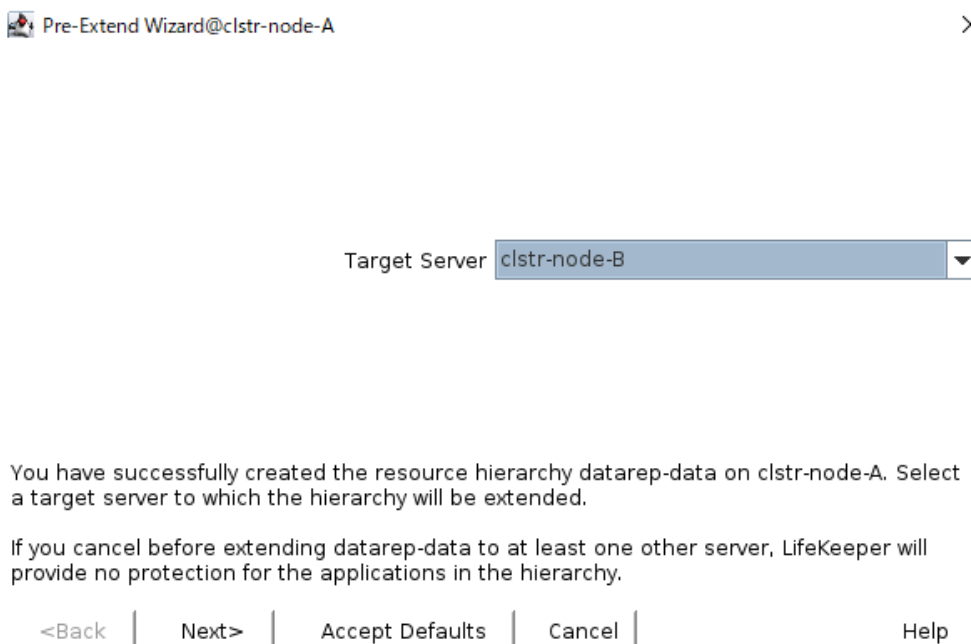


図 9.2-5 Pre Extend ウィザード画面

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

Pre-Extend ウィザードは以下の設定値で進めます。

項目	設定値
Template Priority	1
Target Priority	10

Pre Extend checks were successful と表示されれば成功です。

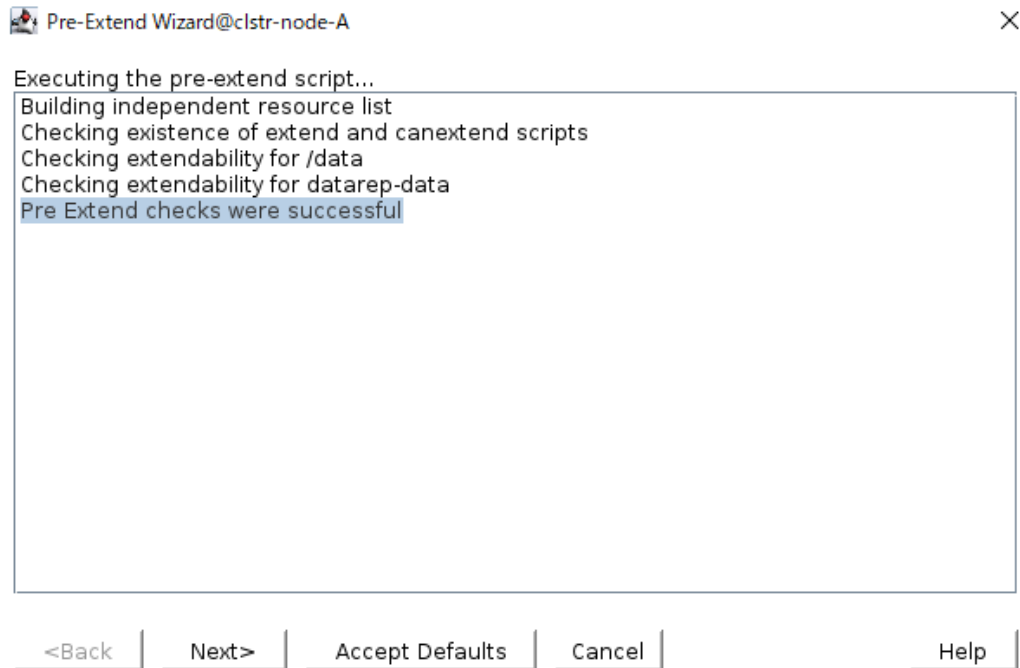


図 9.2-6 Pre Extend check の成功ログ

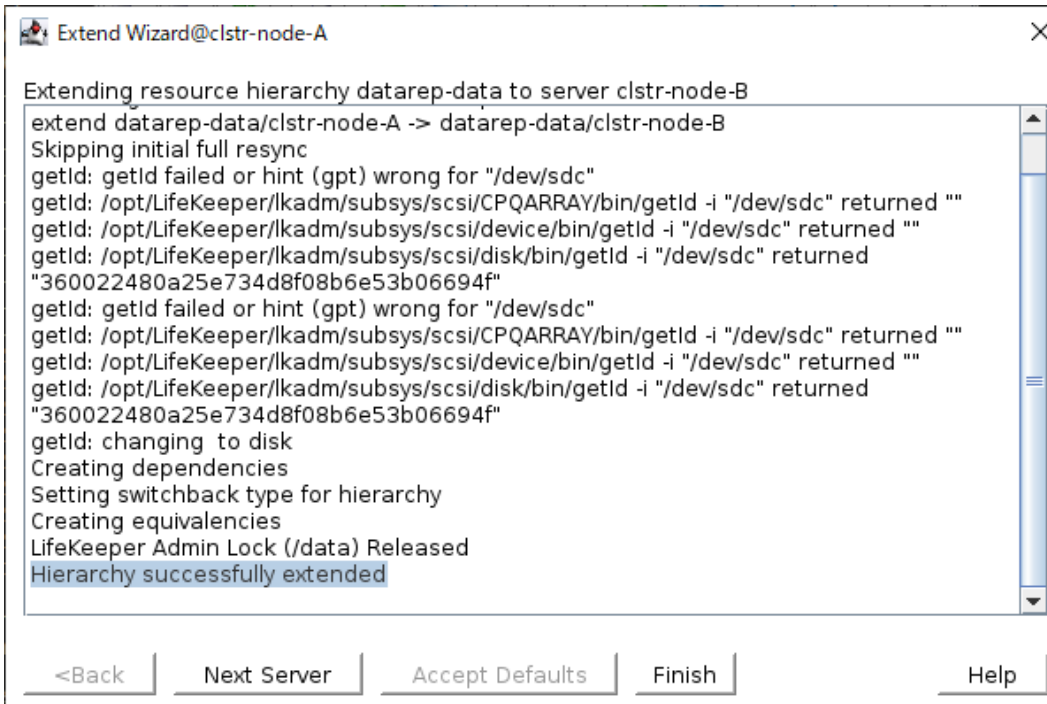
Next をクリックして Extend ウィザードに進めます。

⑤ Extend ウィザード

Extend ウィザードは以下の設定で進めます。

項目	設定値
Mount Point	/data
Root Tag	/data
Target Disk	/dev/sdc (16.0 GB)
DataReplication Resource Tag	datarep-data
Bitmap File	/opt/LifeKeeper/bitmap_data
Replication Path	10.3.2.11/10.3.2.12

Hierarchy successfully extended と出力されたら、リソースの拡張も正常に終了です。



```

Extend Wizard@clstr-node-A
Extending resource hierarchy datarep-data to server clstr-node-B
extend datarep-data/clstr-node-A -> datarep-data/clstr-node-B
Skipping initial full resync
getid: getid failed or hint (gpt) wrong for "/dev/sdc"
getid: /opt/LifeKeeper/lkadm/subsys/scsi/CPQARRAY/bin/getid -i "/dev/sdc" returned ""
getid: /opt/LifeKeeper/lkadm/subsys/scsi/device/bin/getid -i "/dev/sdc" returned ""
getid: /opt/LifeKeeper/lkadm/subsys/scsi/disk/bin/getid -i "/dev/sdc" returned
"360022480a25e734d8f08b6e53b06694f"
getid: getid failed or hint (gpt) wrong for "/dev/sdc"
getid: /opt/LifeKeeper/lkadm/subsys/scsi/CPQARRAY/bin/getid -i "/dev/sdc" returned ""
getid: /opt/LifeKeeper/lkadm/subsys/scsi/device/bin/getid -i "/dev/sdc" returned ""
getid: /opt/LifeKeeper/lkadm/subsys/scsi/disk/bin/getid -i "/dev/sdc" returned
"360022480a25e734d8f08b6e53b06694f"
getid: changing to disk
Creating dependencies
Setting switchback type for hierarchy
Creating equivalencies
LifeKeeper Admin Lock (/data) Released
Hierarchy successfully extended
  
```

<Back | Next Server | Accept Defaults | Finish | Help

図 9.2-7 リソースの拡張終了

Finish を押して、DataKeeper リソースの作成は以上になります。

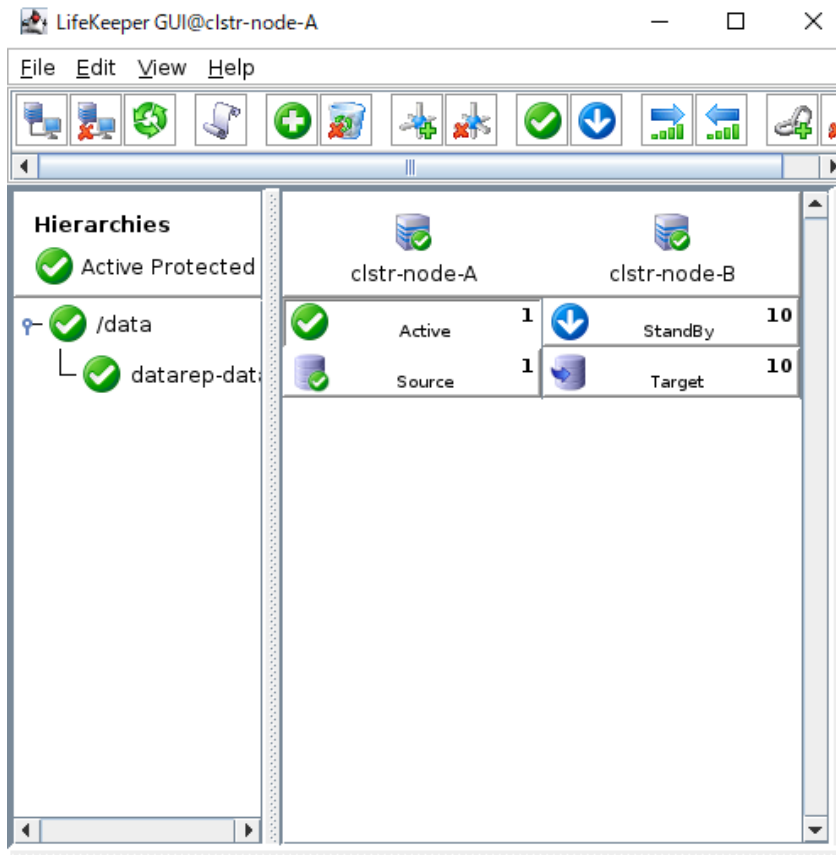
⑥ スイッチオーバーテスト

DataReplication リソースを作成した後すぐは、データレプリケーションリソースが Wait to Resync (再同期中)となります。Wait to Resync から Target の表示に変わるまで少しお待ちください。



図 9.2-8 DataReplication リソースの再同期と GUI 表示

DataReplication リソースが正常に作成されると、GUI では以下のような表示になります。



<-- clstr-node-B: datarep-data: Updating state to Target

図 9.2-9 DataReplication リソース終了後の LifeKeeper GUI

この状態で、/data のスイッチオーバーテストを必ず実施して下さい。

9.3. PostgreSQL リソースの作成

① PostgreSQL サーバーおよびクライアントのインストール

PostgreSQL サーバーを立てるために、ノード A, ノード B に以下のコマンドで PostgreSQL をインストールします。

```
# dnf install postgresql-server
```

また、接続の確認を行う Witness ノードでは、以下のコマンドで PostgreSQL クライアントをインストールしてください。

```
# dnf install postgresql
```

② 稼働ノードの確認

LifeKeeper GUI を確認し、現在の稼働系が clstr-node-A であることをご確認ください。

clstr-node-A が待機系となっている場合は、clstr-node-A へのスイッチバックをご実施ください。

③ データディレクトリの作成(稼働系のみ)

PostgreSQL のデータディレクトリを DataKeeper が保護しているディスク上に作成し、ディレクトリの所有者を postgres ユーザーに変更します。

以下のコマンドを実施してください。

```
# mkdir -m 755 -p /data/pgsql/data  
# chown -R postgres:postgres /data/pgsql
```

以上でデータディレクトリの作成は終了です。

④ データベースの初期化(稼働系のみ)

postgres ユーザーになり、データベースの初期化を行います。

```
# su - postgres  
$ initdb -D /data/pgsql/data
```

⑤ PostgreSQL の設定変更(稼働系のみ)

PostgreSQL を起動する前に、設定を変更します。

1. /data/pgsql/data/pg_hba.conf の設定

次の行を書き換えます。

変更前	変更後
# IPv4 local connections: host all all 127.0.0.1/32 trust	# IPv4 local connections: host all all 10.3.0.0/16 trust
# replication privilege. local replication all trust host replication all 127.0.0.1/32 trust	# replication privilege. local replication all trust host replication all 10.3.0.0/16 trust

2. /data/pgsql/data/postgresql.conf の設定

次の行を書き換えます。

変更前	変更後
# - Connection Settings - #listen_addresses = 'localhost' #port = 5432	# - Connection Settings - listen_addresses = '*' port = 5432

⑥ PostgreSQL デーモンの起動(稼働系のみ)

以下のコマンドで PostgreSQL を起動します。

```
$ pg_ctl start -D /data/pgsql/data -o "-p 5432" -w
```

⑦ 起動確認と接続(稼働系と Witness ノード)

稼働系では、以下のコマンドで PostgreSQL の起動を確認できます。

(postgres ユーザーで実施してください)

```
$ psql -l
```

また、Witness ノード(PostgreSQL クライアント)からは以下のコマンドで正常に接続できる事を確認します。

```
psql -h 10.3.1.11 -U postgres -d postgres
```

正常に接続できた場合、次のようなプロンプトが返ってきます。

```
psql (13.10, server 13.5)
```

```
Type "help" for help.
```

```
postgres=#
```

接続が確認できたら¥q でプロンプトを抜けます。

⑧ リソース作成ウィザードの起動

リソース作成ウィザードは、DataReplication リソース作成時と同様に、GUIのサーバーコンテキストツールバーから Create Resource Hierarchy を選択し、リソース作成ウィザードを起動します。

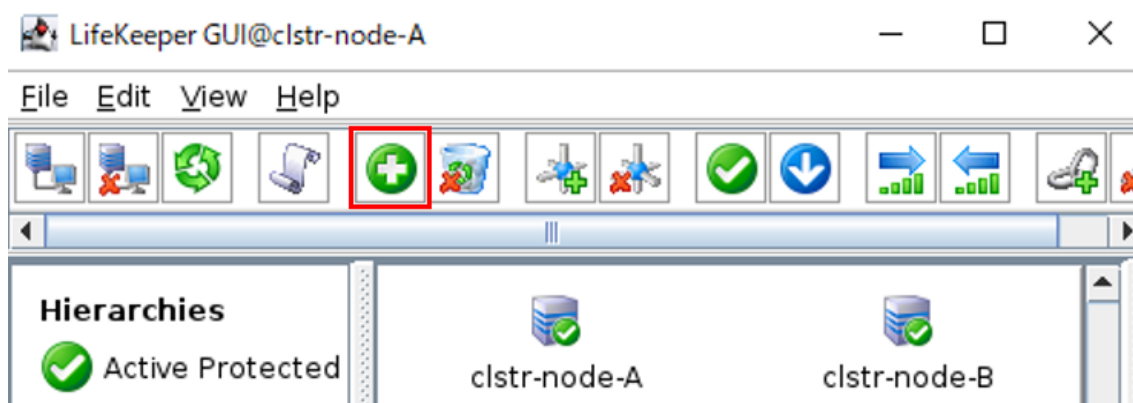


図 9.3-1 リソースの作成ボタン

⑨ Recovery Kit として PostgreSQL Database を選択

⑩ リソース作成ウィザードで以下の値を入力

リソース作成ウィザードは以下の設定で進めます。

項目	設定値
Select Recovery Kit	PostgreSQL Database
Switchback Type	intelligent
Server	clstr-node-A
PostgreSQL Executable Location	/usr/bin
PostgreSQL Client Executable Location	/usr/bin/psql
PostgreSQL Administration Executable Location	/usr/bin/pg_ctl
PostgreSQL Data Directory	/data/pgsql/data

PostgreSQL Port	5432
PostgreSQL Socket Path	/var/run/postgresql/.s.PGSQL.5432
Enter Database Administrator User	postgres
PostgreSQL Logfile	/tmp/pgsql-5432.lk.log
PostgreSQL Database Tag	pgsql-5432

⑪ Pre Extend ウィザードの起動

Pre Extend ウィザードは以下の設定で進めます。

項目	設定値
Target Server	clstr-node-B
Switchback Type	intelligent
Template Priority	1
Target Priority	10

⑫ Extend ウィザード

Extend ウィザードは以下の設定で進めます。

項目	設定値
PostgreSQL Executable Location	/usr/bin
PostgreSQL Database Tag	Pgsql-5432

正常に PostgreSQL リソースが追加されると、GUI 上で PostgreSQL が追加されます。

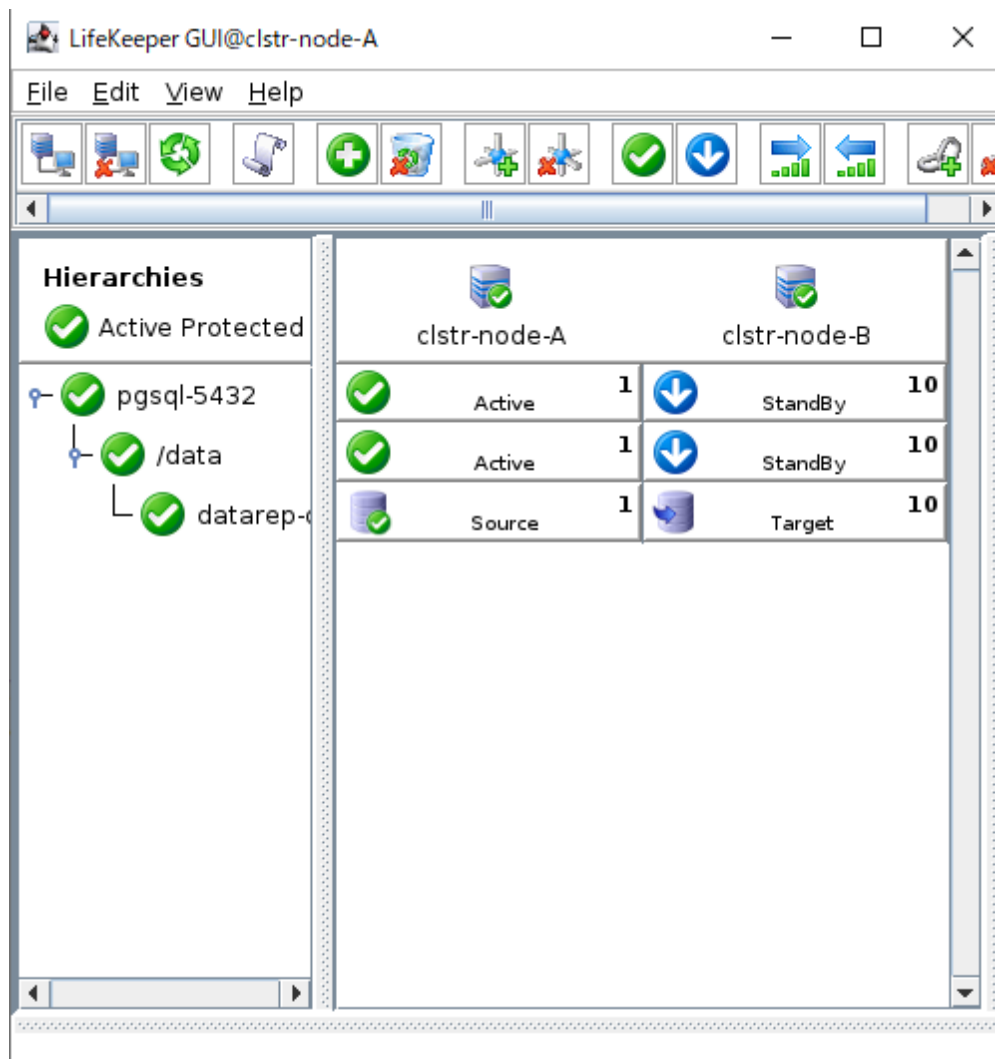


図 9.3-2 PostgreSQL リソース追加後の LifeKeeper GUI

⑬ スイッチオーバーテスト

PostgreSQL リソース作成後も、必ずスイッチオーバーのテストを実施してください。

9.4. LB Health Check リソースの作成

① リソース作成ウィザードの起動

GUIのサーバーコンテキストツールバーから Create Resource Hierarchy を選択し、リソース作成ウィザードを起動します。

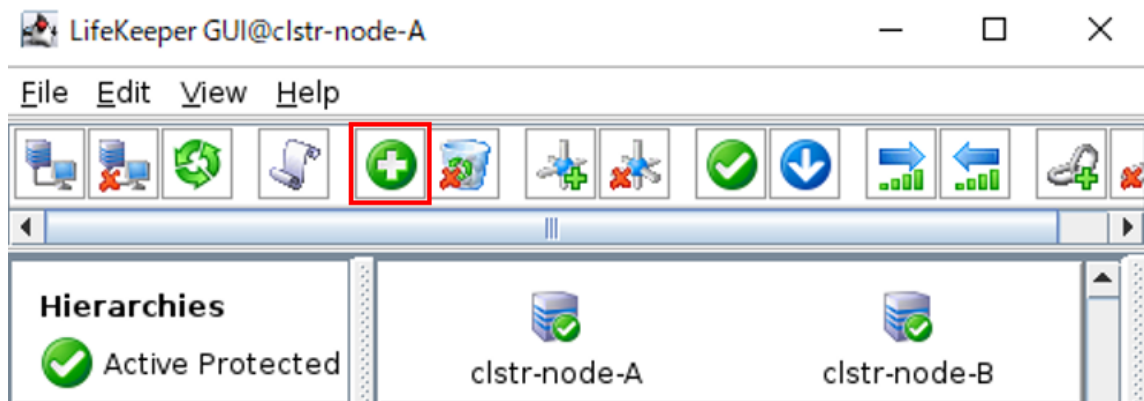


図 9.4-1 リソースの作成ボタン

② Recovery Kit として LB Health Check を選択



図 9.4-2 LB Health Check を選択

③ リソース作成ウィザードで、以下の値を入力

項目	設定値
Switchback Type	intelligent
Server	clstr-node-A
Reply daemon Port	12345 ⁶
Reply daemon message	(空欄)
LB Health Check Resource Tag	lbhc-12345

④ Pre Extend ウィザードの起動

項目	設定値
Target Server	clstr-node-B
Switchback Type	intelligent
Template Priority	1
Target Priority	10

⑤ Extend ウィザード

項目	設定値
LB Health Check Resource Tag	lbhc-12345

⑥ スイッチオーバーテスト

LB Health Check リソース作成後も、必ずスイッチオーバーのテストを実施してください。

⁶ 手順 4.5 正常性プローブのポート番号で 12345 以外の値を設定した方は、正常性プローブで設定したポート番号を設定してください。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

⑦ Azure 上での LB Health Check リソースの動作確認

Azure ポータル上から LB Health Check リソースの動作を確認する事ができます。

1. ロードバランサーの概要ページサイドメニューから「分析情報」を選択します。

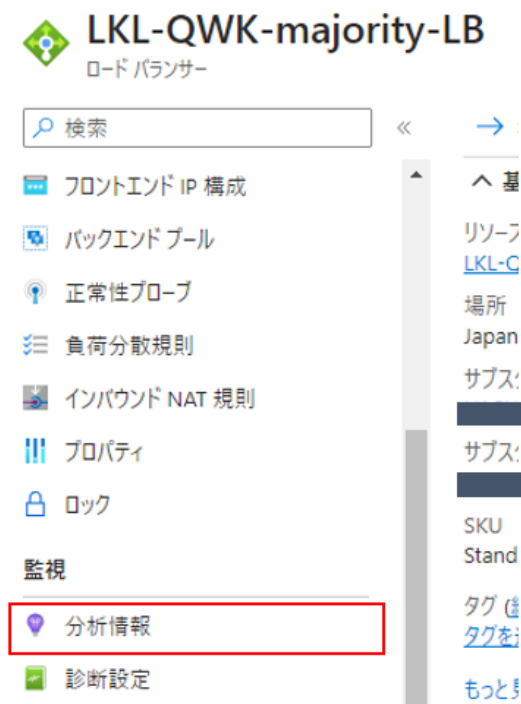


図 9.4-3 Azure ポータル ロードバランサーの概要画面

2. バックエンド IP アドレスごとの正常性プローブの状態を確認します。
スイッチバックテストが正常に動作している場合は次の図のように、切り替わりのポイントを確認できます。

バックエンド IP アドレスごとの正常性プローブの状態

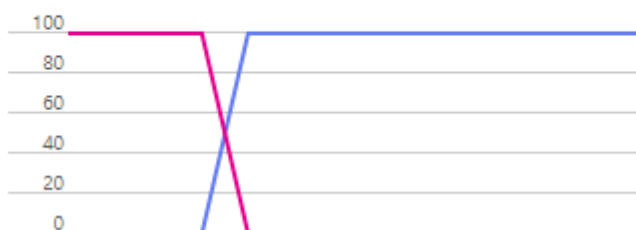


図 9.4-4 正常性プローブの確認

9.5. IP リソースの作成

① リソース作成ウィザードの設定

GUIのサーバーコンテキストツールバーから Create Resource Hierarchy を選択し、リソース作成ウィザードを起動します。

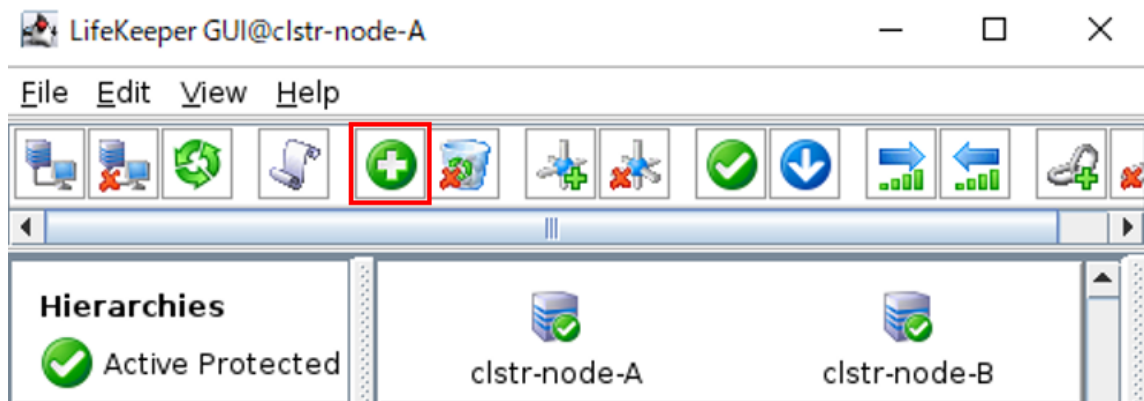


図 9.5-1 リソースの作成ボタン

② Recovery Kit として IP を選択

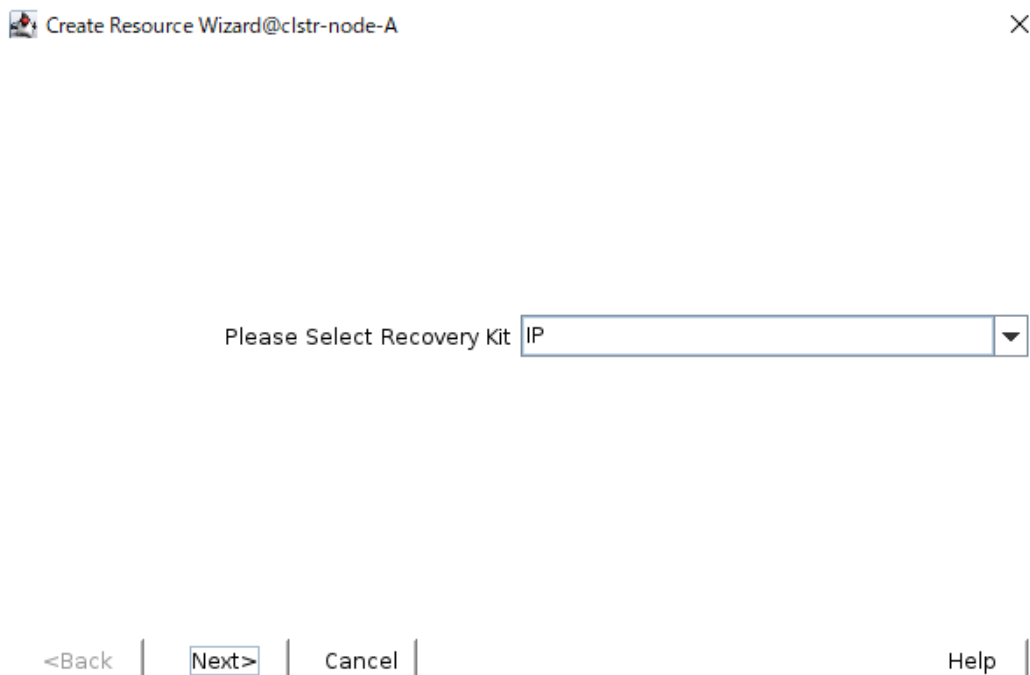


図 9.5-2 IP を選択

③ リソース作成ウィザードで、以下の値を入力

項目	設定値
Switchback Type	intelligent
IP Resource	10.3.1.200 ⁷
Netmask	255.255.255.255
Network Interface	eth0
IP Resource Tag	ip-10.3.1.200

④ Pre Extend ウィザードの起動

項目	設定値
Target Server	clstr-node-B
Switchback Type	intelligent
Template Priority	1
Target Priority	10

⑤ Extend ウィザード

項目	設定値
Target Server	clstr-node-B
Switchback Type	intelligent
Template Priority	1
Target Priority	10
IP Resource	10.3.1.200
Netmask	255.255.255.255
Network Interface	eth0
IP Resource Tag	ip-10.3.1.200

⑥ スイッチオーバーテスト

IP リソース作成後も、必ずスイッチオーバーのテストを実施してください。

⁷ 必ず Azure Load Balancer のフロント IP に設定した IP アドレスを使用してください。

9.6. 階層関係の定義

各リソースの依存関係を定義します。

手順通り一通りリソースを作成すると、次の画像のようにリソース関係がバラバラに作成された状態に出来上がります。

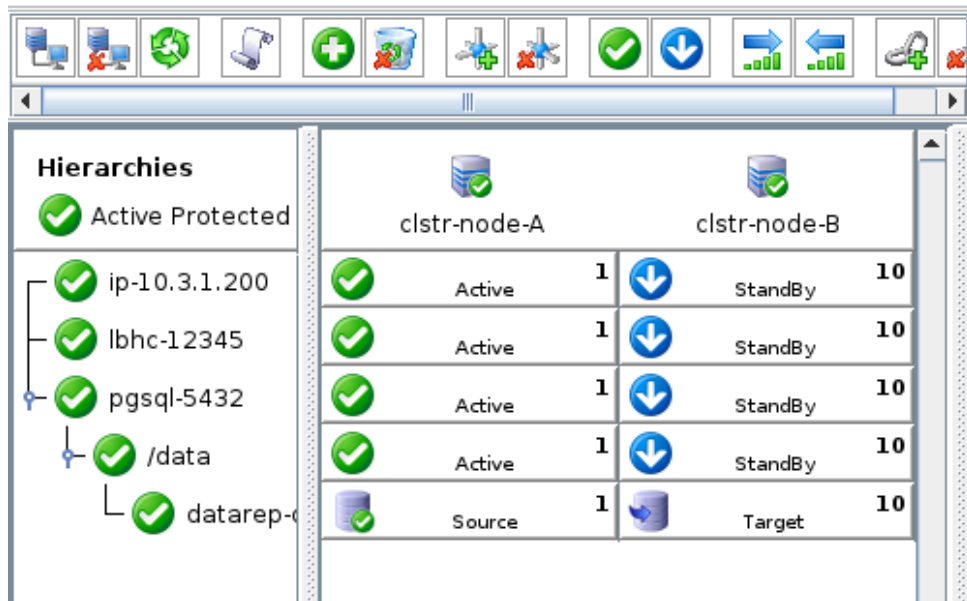


図 9.6-1 すべてのリソース作成後の LifeKeeper GUI

この状態だと、IP リソース, LB Health Check リソース, PostgreSQL リソースの 3 つに依存関係が定義されておらず、IP リソースはノード A で、それ以外のリソースはノード B で稼働され、サービスが提供できない事態が起こります。そのため、リソース間に適切な依存関係を作成する必要があります。

本手順では PostgreSQL リソースの下にもともとあった Data Replication リソースのほかに、IP リソースと LB Health Check リソースを配置していきます。

完成図は次の画像の通りです。

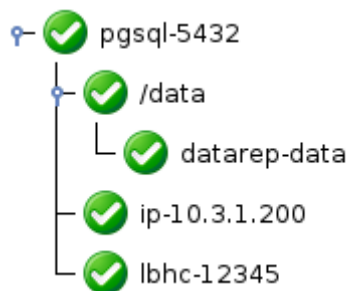


図 9.6-2 階層関係作成後のリソース構成図

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

① PostgreSQL リソースを右クリック

PostgreSQL リソースを右クリックし、Create Dependency を選択します。

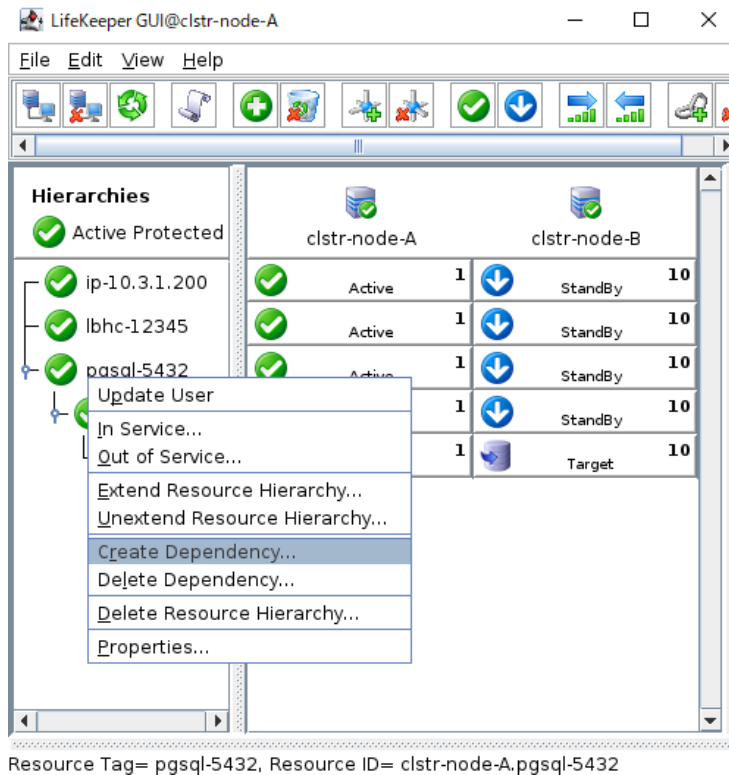


図 9.6-3 PostgreSQL リソースのコンテキストメニュー

② 子リソースを選択

子リソース(依存元)に IP リソース(ip-10.3.1.200)を選択します。

The dependency creation was successful と表示されたら、成功です。

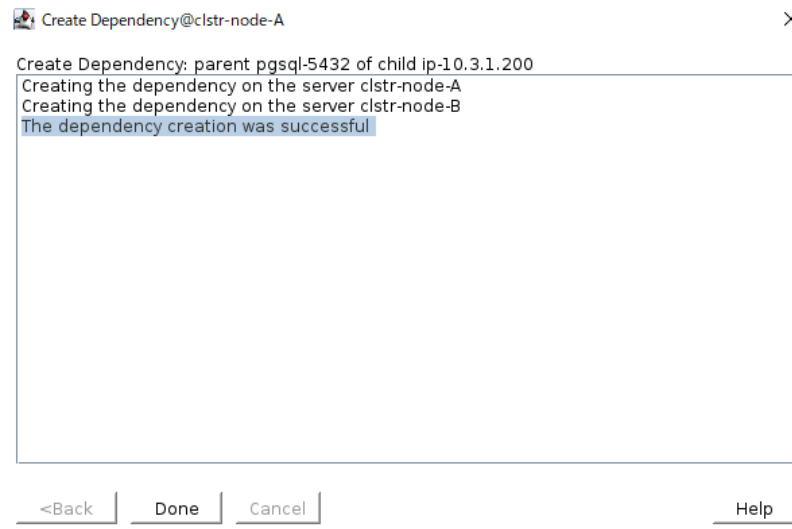


図 9.6-4 依存関係の作成に成功したログ

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

③ GUIの確認

PostgreSQL リソースと IP リソースに正常に依存関係が作成されると、GUI は次のような表示になります。

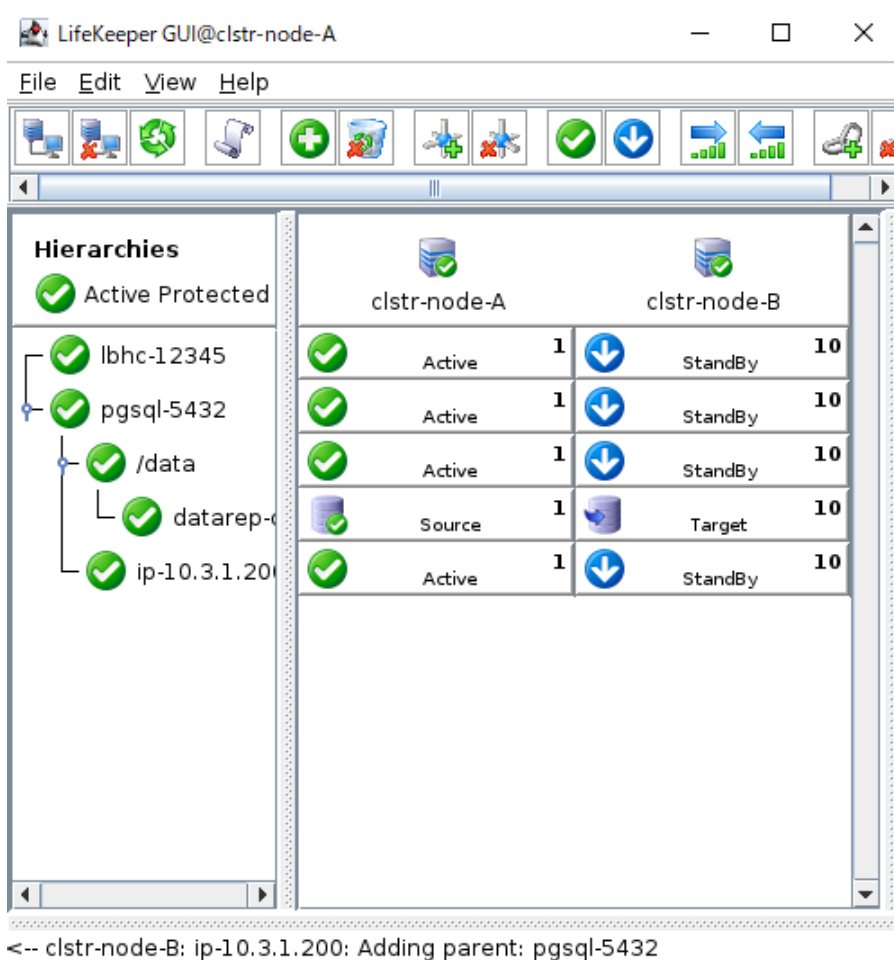


図 9.6-5 IP リソースの依存関係終了後の LifeKeeper GUI

④ LB Health Check リソースの依存関係の作成

①から③の手順を再度繰り返し、LB HealthCheck リソースにも依存関係を作成します。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

LB Health check リソースに依存関係を作成した後、GUI 表示は以下のようになります。

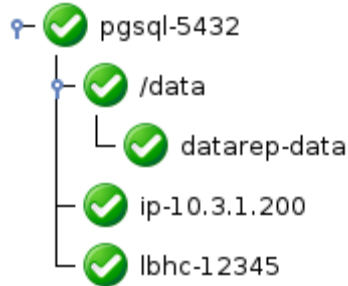


図 9.6-6 すべてのリソースの依存関係作成後のリソースツリー画面

正常に作成されると依存関係の定義は終了です。

⑤ スイッチオーバーテストの実行

依存関係も含めて、最後のスイッチオーバーテストをおこないます。

PostgreSQL リソースの稼働をノード B に移し、クライアントから通信ができる事を確認します。

1. PostgreSQL リソースのスイッチオーバー

GUI を使用し、PostgreSQL リソースの稼働をノード B に移します。

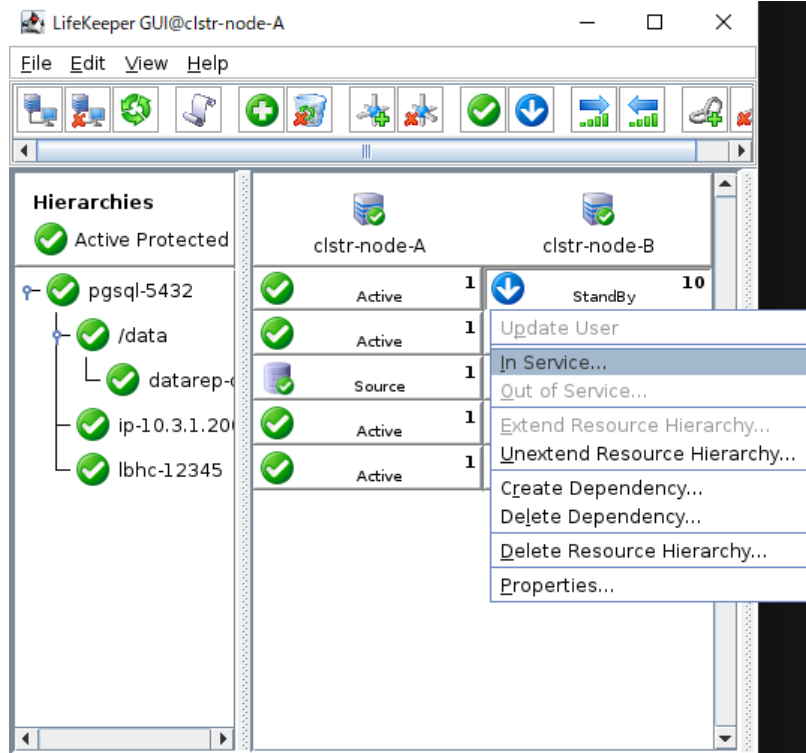


図 9.6-7 PostgreSQL リソースのコンテキストメニュー

2. クライアントからの通信確認

クライアントノードから、以下のコマンドを使用し、正常に PostgreSQL に接続できるか検証を行います。

```
$ psql -h 10.3.1.200 -U postgres -d postgres
```

正常に接続が確認出来たら、依存関係の定義も終了です。

10. Quorum / Witness Kit の導入

LifeKeeper では、スプリットブレインの発生を防ぐために、Quorum / Witness Kit を提供しております。

詳細はテクニカルドキュメンテーションをご覧ください。

<https://docs.us.sios.com/spslinux/9.7.0/ja/topic/quorum-witness>

本ガイドでは Quorum モードに majority モードを使用しております。

以下、Quorum / Witness Kit の設定方法とフェイルオーバーのテスト方法です。

① Quorum / Witness Kit 各パラメーターの確認

全ノード共通

/etc/default/LifeKeeper をお好きなエディタで開きます。

以下のパラメーターの値をそれぞれ確認します

パラメーター	規定値	本ガイド推奨値
QUORUM_MODE	majority	majority
QUORUM_TIMEOUT_SECS	20	20
WITNESS_MODE	remote_verify	remote_verify
QUORUM_LOSS_ACTION	fastkill	fastkill

すべて規定値のため、変更は必要ございません。

各パラメーターの役割はそれぞれ以下の通りです。

パラメーター	役割
QUORUM_MODE	Quorum モードを指定します。
QUORUM_TIMEOUT_SECS	Quorum チェックで tcp/ip 接続を完了するための タイムアウト時間を秒単位で指定します。この時間内に接続が完了しない場合は、失敗、または使用不可となります
WITNESS_MODE	Witness モードを指定します。
QUORUM_LOSS_ACTION	Quorum が失われた場合の動作を指定します。

② フェイルオーバーシナリオ

Quorum Witness Kit が設定されている事を確認できたため、フェイルオーバーのテストを行います。

本テストのフェイルオーバーのシナリオは以下の通りです。

ノード A のネットワークに障害が発生(サービスの提供が不可能)

この場合、ノード A は以下の動作をおこないます。

1. 他ノードとのコミュニケーションパスが全断している事を検知します。
2. ノード B および Witness ノードとの通信ができないため、自ノードは少数派であると判断します。
3. ① Quorum / Witness Kit 各パラメーターの確認で事前に設定しておいた QUORUM_LOSS_ACTION を実行します。

また、ノード B は以下の動作を行います。

1. 元稼働系のノード A とのコミュニケーションパスが全断している事を検知します。
2. Witness ノードとの通信が可能のため、自ノードは多数派であると判断します。
3. ノード A が見えない事を Witness ノードに確認した後、フェイルオーバーを開始します。
4. これにより、保護対象リソースは、ノード B 上で起動されます。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

③ フェイルオーバーテスト

これから、実際に稼働系(ノード A)のネットワークに障害を起こし、フェイルオーバーが行われる事を確認します。

1. すべてのリソースの稼働をノード A に移しておきます。

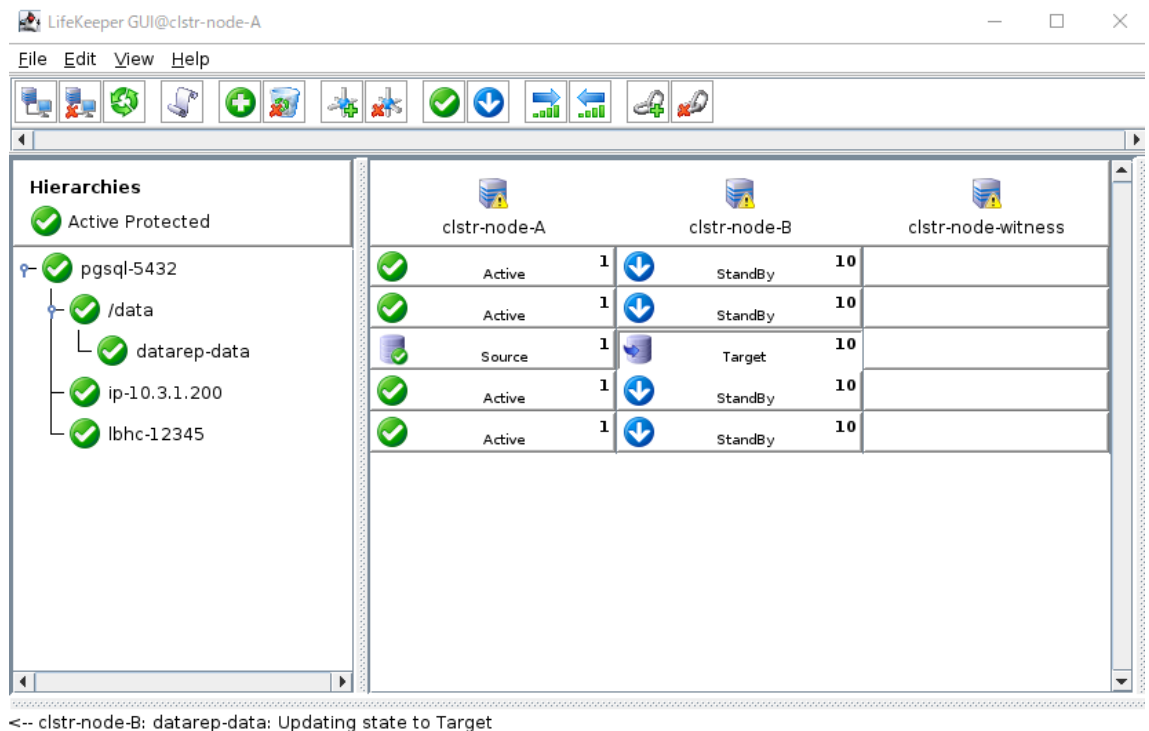


図 10-1 すべてのリソースの稼働が clstr-node-A にある状態

2. この状態でノード A の NIC を無効に設定します。(ノード A のみ)

以下のコマンドで一時的にネットワークインターフェイスを無効に設定します。

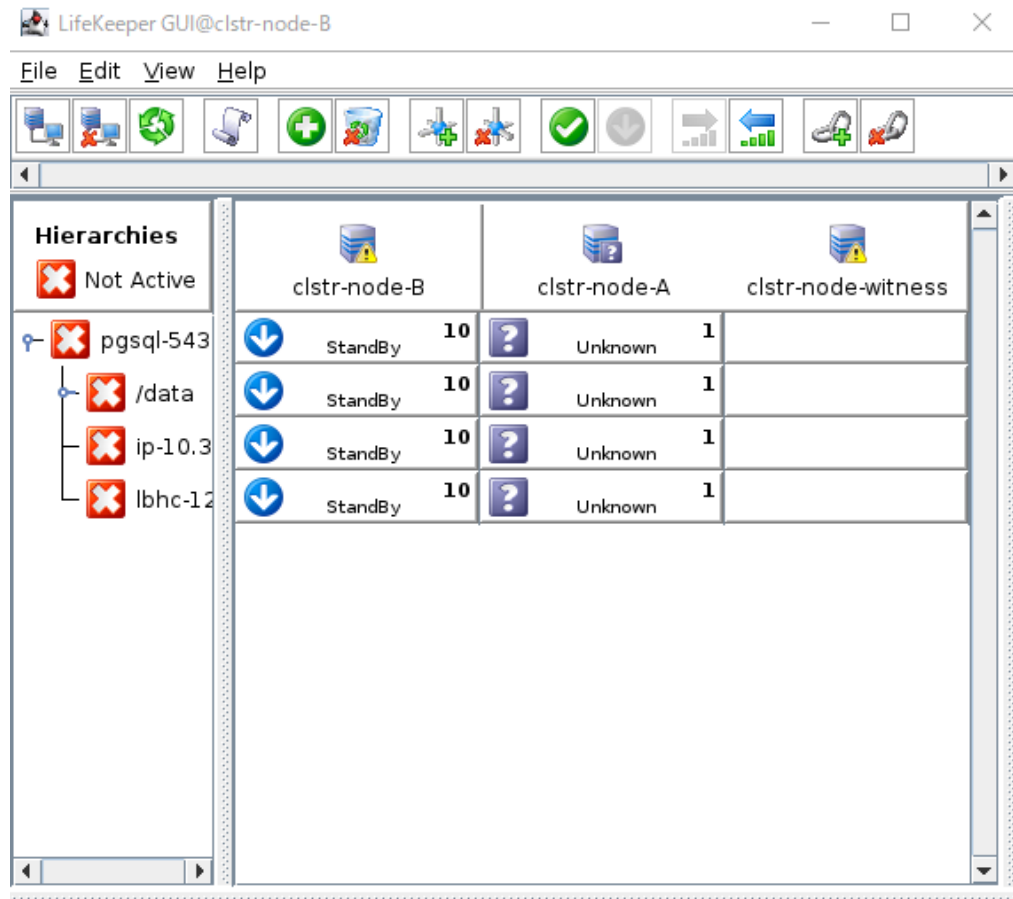
なお、再起動後には、もとに戻ります。

注意点としては、eth1(10.3.2.11)を先に eth0(10.3.1.11)を後に無効化してください。SSH のセッションは最後まで切れないようにする必要があります。

```
ifconfig eth1 down
ifconfig eth0 down
```

3. ノード B の GUI を起動し、フェイルオーバーされる事を確認します。

ノード A の NIC を無効化した直後



```
<-- clstr-node-A: datarep-data: Updating state to Unknown
```

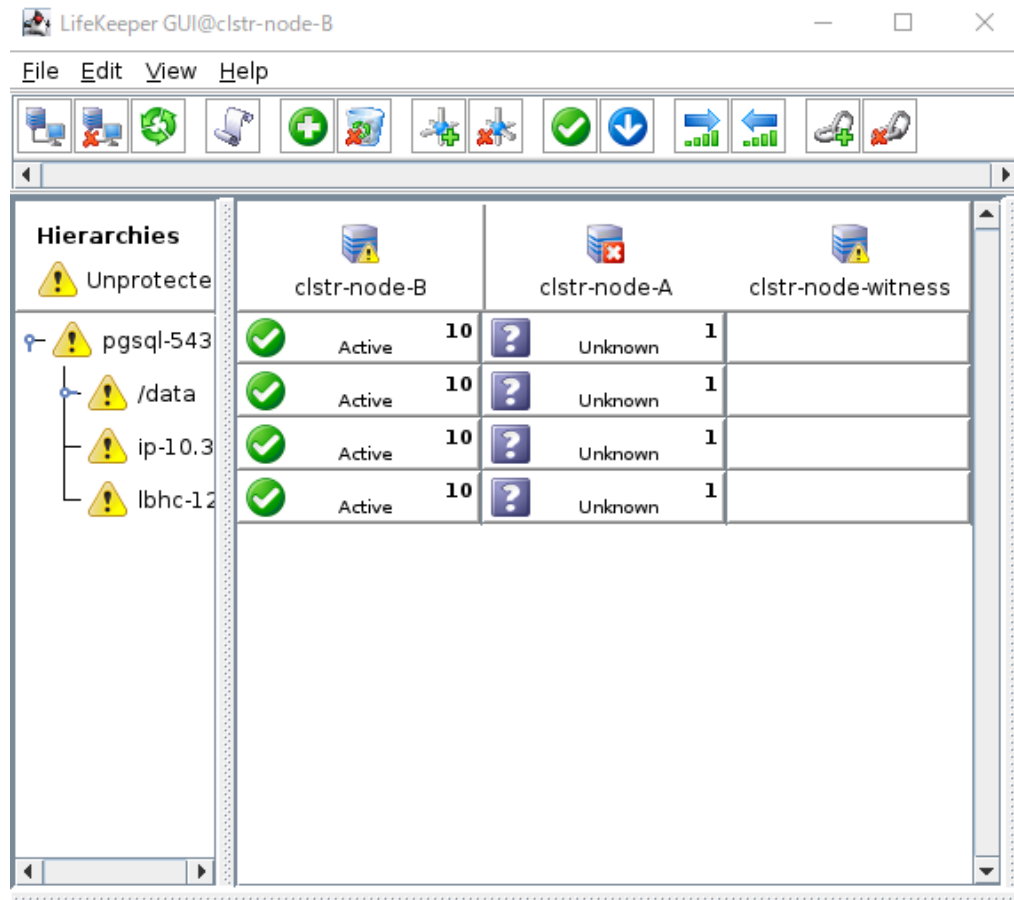
図 10-2 ノード A の NIC を無効化した直後のノード B の GUI

Unknown は、対象ノードへ通信ができないため、リソースの稼働状態が確認できない事を示します。

LifeKeeper for Linux Quorum/Witness Kit majority mode on Azure

Step by step Guide

数秒後、すべてのリソースがノード B 上で稼働状態となります。



<-- clstr-node-B: pgsql-5432: Updating state to Active

図 10-3 Quorum / Witness 機能を使用したフェイルオーバー完了後

4. 以上でフェイルオーバーのテストも終了になります。

この状態のまま稼働を継続した場合、ノード B に障害が発生すると、サービスが停止してしまいます。

そのため、テストが終了しましたら、Azure ポータルでノード A を再起動する必要があります。

11. お問い合わせ前に

11.1. 製品サポートへお問い合わせいただく際に収集すべき情報について

LifeKeeper には構成情報やログを一括取得する lksupport というツールが用意されています。障害解析やインストレーションに関する調査では、ログや構成情報の確認が必要となるケースが大半となります。そのため、障害解析やインストレーションに関する調査をご希望の際は、以下の情報をご提供ください。

- 事象発生時刻
- 全クラスターノードの lksupport
- その他、お気づきの点

補足

lksupport でアーカイブファイルを生成するために以下のコマンドを実行します。

```
# /opt/LifeKeeper/bin/lksupport
```

lksupport の実行に成功すると/tmp 以下に次の命名規則のファイルが生成されます。

このファイルをクラスター全ノード分収集してお送りください。

```
/tmp/lksupport/<ホスト名>.lksupport.<タイムスタンプ>.tar.gz
```

※データレプリケーションの構成では nbd デバイスがロードされ、/dev/nbd*が作成されます。lksupport 実行時や、lvdisplay、vgdisplay コマンド等、/dev/nbd*を走査する処理が実行された場合は、ご利用の環境によっては以下のようなメッセージが出力されることがありますが問題ではありません。

```
Nov 9 01:30:28 lk064 kernel: nbd0: Attempted send on closed socket
Nov 9 01:30:28 lk064 kernel: end_request: I/O error, dev nbd0, sector 0
Nov 9 01:30:28 lk064 kernel: nbd2: Attempted send on closed socket
Nov 9 01:30:28 lk064 kernel: end_request: I/O error, dev nbd2, sector 0
```

11.2. よく利用する LifeKeeper のコマンド

それぞれのコマンドの詳細につきましては、テクニカルドキュメンテーションをご参照ください。

- LifeKeeper GUI クライアントの起動
/opt/LifeKeeper/bin/lkGUIapp

- LifeKeeper の起動
/opt/LifeKeeper/bin/lkcli start
または
/opt/LifeKeeper/bin/lkstart
または
systemctl start lifekeeper.service

- LifeKeeper の停止（リソースも停止する）
/opt/LifeKeeper/bin/lkcli stop
または
/opt/LifeKeeper/bin/lkstop
または
systemctl stop lifekeeper.service

- LifeKeeper の停止（リソースは停止しない）
/opt/LifeKeeper/bin/lkcli stop -f
または
/opt/LifeKeeper/bin/lkstop -f

- LifeKeeper のステータス確認
ステータスを簡易表示するには「-e」オプションを付与します。
/opt/LifeKeeper/bin/lkcli status(もしくは lkcli status -e)
または
/opt/LifeKeeper/bin/lcdstatus(もしくは lcdstatus -e)

- LifeKeeper のログの確認
/var/log/lifekeeper.log を参照します。リアルタイムにログの出力を確認したい場合には以下のように tail コマンドを使用することもできます。
tail -f /var/log/lifekeeper.log

- LifeKeeper の構成情報やログの一括取得

```
# /opt/LifeKeeper/bin/lksupport
```

- LifeKeeper の構成情報のバックアップ・リストア

LifeKeeper の構成情報のバックアップ

```
# /opt/LifeKeeper/bin/lkbackup -c
```

LifeKeeper の構成情報のリストア

```
# /opt/LifeKeeper/bin/lkbackup -x -f archive.<タイムスタンプ>.tar.gz
```

12. お問い合わせ

本書の記載内容についてのお問い合わせ先

- LifeKeeper 製品の導入を検討中のお客様

弊社パートナー営業部までお問い合わせください。

お問い合わせメールフォーム

https://mk.sios.jp/BC_Web_Free-entry_Inquiry.html

- LifeKeeper 製品をご購入済みのお客様

弊社 LifeKeeper 製品サポート窓口までお問い合わせください。

購入後のお問い合わせ

https://bc.sios.jp/support_lk.html

13. 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、その他第三者の行ためにより本書の利用者に生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、転載、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：<https://sios.jp>