

LifeKeeper for Linux on Azure Shared Disk 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.	製品に関する情報	5
3.	構成環境	6
3.1.	Azure 共有ディスク概要	6
3.2.	ネットワーク構成図	7
3.3.	リソース構成図	7
3.4.	既知の問題の確認	8
3.5.	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 の上の環境構築	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	15
4.4.	仮想マシンの作成	16
4.5.	インターナルロードバランサーの作成	28
5.	OS の設定	33
5.1.	仮想マシンへのログイン	33
5.2.	接続確認と X Server の有効化	34
5.3.	NIC アドレスの固定化	35
5.4.	SELinux の無効化	37
5.5.	ファイアウォールの無効化	37
5.6.	名前解決	37
6.	ローカルリポジトリの設定	38
6.1.	/home の拡張	38
6.2.	OS イメージの転送	38
6.3.	fstab の編集	39
6.4.	ローカルリポジトリの作成	39
6.5.	rh-cloud-base.repo の無効化	40
6.6.	GUI 設定	40
7.	LifeKeeper のインストール	41
7.1.	LifeKeeper インストールイメージの転送	41
7.2.	LifeKeeper インストール	41
8.	コミュニケーションパスの作成	45
8.1.	GUI を使用したコミュニケーションパスの作成	45
8.2.	GUI クライアントのステータス情報	47
9.	各種リソースの作成	48
9.1.	リソースの作成前に	49
9.2.	ファイル共有リソースの作成	51

9.3.	PostgreSQL リソースの作成.....	54
9.4.	LB Health Check リソースの作成.....	58
9.5.	IP リソースの作成.....	60
9.6.	階層関係の定義.....	61
10.	動作検証テスト.....	64
11.	お問い合わせ.....	66
12.	免責事項.....	67

改訂履歴

版	更新日	変更情報
初版	2023/12/15	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ポリシーに基づいています。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 Azure 共有ディスクを用いた LifeKeeper for Linux 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 Shared Disk を用いた Life Keeper for Linux 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. 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://lkduserportal.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の説明です。

本ガイドでは Red Hat Enterprise Linux 9.0 を使用し LifeKeeper for Linux v9.8.0 と Azure 共有ディスク構成を用いた 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. Azure 共有ディスク概要

Azure 共有ディスクとは、マネージドディスクを複数の仮想マシン (VM) に同時に接続できるようにする機能です。複数の VM に接続することで、リアルタイムに更新されるデータをどちらのノードからでも読み込むことが可能になります。

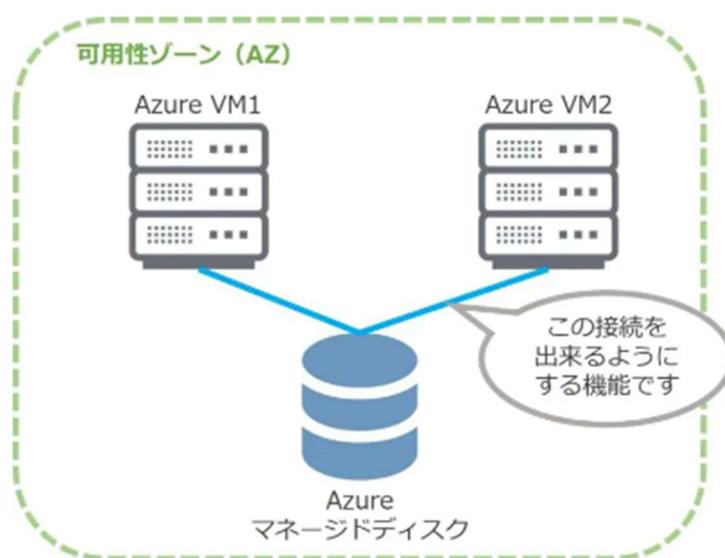


図 4.1-1 Azure 共有ディスクの仕様

Azure 共有ディスクをクラスター共有ディスクとして利用することで、ダウンタイムの短縮やシステムの自動復旧の実現など、可用性を高めることが可能となります。

3.2. 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。

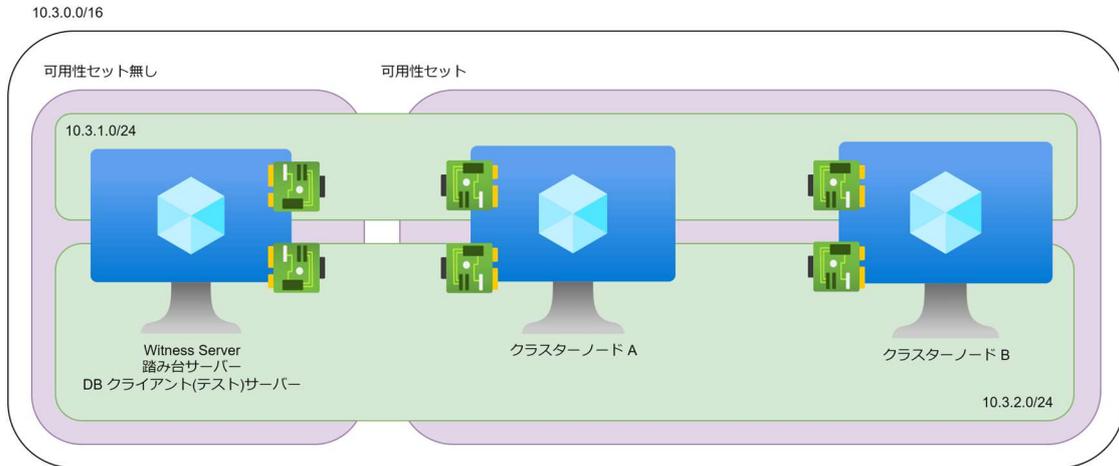


図 3.1-1 ネットワーク構成図

3.3. リソース構成図

本ガイドのリソース構成図は以下に示す通りになります。

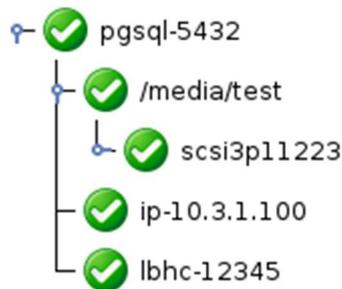


図 3.2-1 リソース構成図

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-5432	PostgreSQL リソース
/media/test	ファイルシステムリソース
Datarep-data	SCSI3 Reserve System リソース
Ip-10.3.1.100	仮想 IP リソース
Lbhc-12345	LB Health Check リソース

3.4. 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。

3.5. 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、仮想マシンに 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。

[ホーム](#) > [リソースグループ](#) >

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

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 * ①

リソースグループ * ②

リソースの詳細

リージョン * ③

図 4.1-2 リソースグループ作成ウィザード「基本」タブ

- ③ 作成者がわかるようにタグをつけます。以降は、本手順同様に作成する全てのリソースにタグをつけます。

LifeKeeper for Linux on Azure Shared Disk

Step by step Guide

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 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。
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- shared-disk
仮想ネットワーク名	vnet-LKL-shared-disk
地域	(Asia Pacific) 東日本

プロジェクトの詳細

デPLOYされたリソースとコストを管理するためのサブスクリプションを選択します。フォルダーなどのリソースグループを使用すると、すべてのリソースを整理して管理することができます。

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

仮想ネットワーク名*

地域 ⓘ * [エッジゾーンにデプロイ](#)

図 4.2-1 仮想ネットワーク 基本設定画面

画面下部の[Next]から次へ進みます。

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。

LifeKeeper for Linux on Azure Shared Disk

Step by step Guide

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

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
default	10.0.0.0 - 10.0.255	/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 4.2-2 仮想ネットワークの IP アドレス設定画面

新しい IP 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)

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。

IPv4 アドレス空間の追加 | v

10.3.0.0/16 🗑️ アドレス空間の削除

10.3.0.0 /16 (65,536 個のアドレス) v

10.3.0.0 - 10.3.255.255 (65536 アドレス)

+ サブネットの追加

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

図 4.2-3 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

- 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 4.2-4 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 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 Default

名前 * ① ZoneAll

開始アドレス * ① 10.3.1.0

サブネットサイズ ① /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)

図 4.2-5 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。
- 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 4.2-6 サブネット追加後

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。

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。

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-shared-disk
名前	availability-set

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 * ①

リソース グループ * ①
[新規作成](#)

インスタンスの詳細

名前 * ① ✓

地域 * ①

障害ドメイン ① 2

更新ドメイン ① 5

マネージド ディスクを使用 ① いいえ (クラシック) はい (配置)

図 4.3-1 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。

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 踏み台サーバー兼 PostgreSQL クライアントサーバーの作成

本ガイドで各ノードへ接続するための踏み台サーバーとして使用する仮想マシンを作成します。また、PostgreSQL のクライアントとしても使用します。本ガイドでは、以降踏み台サーバーと呼称します。簡単に全体の構成図について示します。

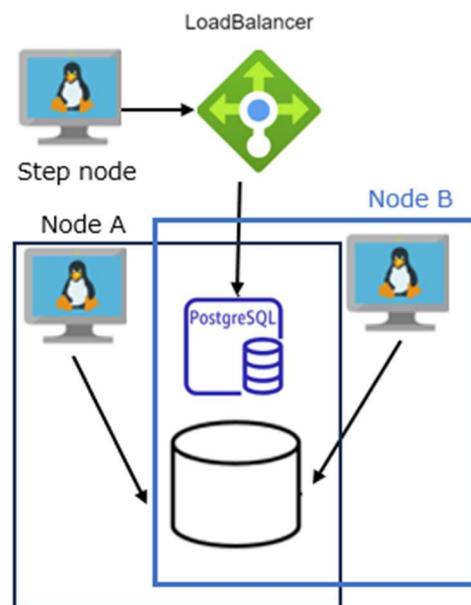


図 4.4-1 環境構成図

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。
2. 仮想マシンイメージの選択
OS の選択画面で Red Hat Inc が提供している Red Hat Enterprise Linux を選択します。
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。

LifeKeeper for Linux on Azure Shared Disk

Step by step Guide



図 4.4-2 Red Hat 仮想マシンイメージ選択画面

4. 基本の設定

踏み台ノードは、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-shared-disk
仮想マシン名	step-node-
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)

表を見ながら赤枠部分を設定値に変更します。

LifeKeeper for Linux on Azure Shared Disk

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](#)

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](#)

VM アーキテクチャ 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

サイズ * [すべてのサイズを表示](#)

休止状態を有効にする (プレビュー)

i To enable Hibernation, you must register your subscription. [詳細情報](#)

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 *

[確認および作成](#)

[< 前へ](#) [次: ディスク >](#)

図 4.4-3 仮想マシンの基本設定

5. ディスクの設定

LifeKeeper for Linux on Azure Shared Disk

Step by step Guide

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

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 4.4-4 ディスクの変更

6. ネットワークの設定

踏み台サーバー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-shared-disk
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

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。

8. タグの設定

作成者が分かるようにタグを設定します。

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踏み台サーバーの手順に加えて、Azure 共有ディスクの設定を行います。

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-shared-disk
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

3. ディスクの設定

OS ディスクの種類の変更 及び Azure 共有ディスクの追加を行います。

4. 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。今回、テスト環境のためコスト面を考慮して Standard SSD に変更しますが、実際にはご自身の設定をご利用ください。

5. Azure 共有ディスクの追加

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
共有ディスクを有効にする	はい
最大共有数	2

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。

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。

3. 共有ディスクの有効化

共有ディスクを有効にする を押下して Azure 共有ディスクを有効化します。

以上の手順が終了すると、次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	16 GiB Standard SSD LRS サイズを変更します
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 checked="" type="radio"/> はい <input type="radio"/> いいえ
最大共有数 ①	<input type="text" value="2"/>
VM と共にディスクを削除	<input type="checkbox"/>

図 4.4-5 ディスクの確認画面

6. 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-shared-disk
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

7. 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。

8. タグの設定

タグを設定します。

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。

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。
ディスクの設定では、ノード A で作成した共有ディスクを追加します。

基本 タブの設定

項目	設定値
リソースグループ	LKL-shared-disk
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-shared-disk
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

1. 既存ディスクの接続

ノード A で作成した共有ディスクをノード B に接続します。

画面下部の 既存のディスクの接続 を押下してください。

作成したディスクの追加を行います、完了した画面は次のようになります。



図 4.4-6 clstr-node-B への共有ディスクの追加

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。
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 4.4-7 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。
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。
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。

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
step-node	10.3.1.50
clstr-node-A	10.3.1.11
clstr-node-B	10.3.1.12

③ 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。

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。

[フィードバック](#) [ネットワーク インターフェイスの接続](#) [ネットワーク インターフェイスのデタッチ](#)

ネットワーク インターフェイスの接続

既存のネットワーク インターフェイスの接続

接続可能なネットワーク インターフェイスがありません

ネットワーク インターフェイスの作成と接続

OK

キャンセル

図 4.4-8 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

4. 他ノードへの NIC の追加

上記の手順でノード 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。

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。

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。

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。

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-shared-disk
名前	LKL-shared-disk-LB
SKU	Standard
種類	内部

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 4.5-1 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-shared-disk-LB
仮想ネットワーク	Vnet-LKL-shared-disk
サブネット	ZoneAll (10.3.1.0/24)
割り当て	静的
IP アドレス	10.3.1.100

名前 * ✓

仮想ネットワーク

サブネット * ▼

割り当て 動的 静的

IP アドレス * ✓

可用性ゾーン * ⓘ ▼

図 4.5-2 フロントエンド IP 構成の変更画面

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。

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 4.5-3 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①

バックエンドプールの構成

NIC

IP アドレス

IP 構成

仮想マシンと仮想マシン スケール セットに関連付けられている IP 構成は、ロードバランサーと同じ場所にあり、同じ仮想ネットワーク内にある必要があります。

図 4.5-4 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。

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 4.5-5 負荷分散規則の追加

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-shared--LB-PostgreSQL
フロントエンド IP アドレス	LKL-shared-LB (10.3.1.100)
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

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-shared-probe
プロトコル	TCP
ポート	12345
間隔	5

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。

LifeKeeper for Linux on Azure Shared Disk Step by step Guide

名前 *	<input type="text" value="LKL-shared-LB-PostgreSQL"/>
IPバージョン *	<input checked="" type="radio"/> IPv4 <input type="radio"/> IPv6
フロントエンド IP アドレス * ⓘ	<input type="text" value="LKL-shared-LB (10.3.1.200)"/> ▼
バックエンドプール * ⓘ	<input type="text" value="LKL-LB-backpool"/> ▼
高可用性ポート ⓘ	<input type="checkbox"/>
プロトコル	<input checked="" type="radio"/> TCP <input type="radio"/> UDP
ポート *	<input type="text" value="5432"/>
バックエンドポート * ⓘ	<input type="text" value="5432"/>
正常性プローブ * ⓘ	<input type="text" value="(新規) LKL-shared-probe (TCP:12345)"/> ▼ 新規作成
セッション永続化 ⓘ	<input type="text" value="なし"/> ▼
アイドルタイムアウト(分) * ⓘ	<input type="text" value="4"/>
TCP リセットを有効にする	<input type="checkbox"/>
フローティング IP を有効にする ⓘ	<input checked="" type="checkbox"/>

図 4.5-6 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。

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踏み台ノードの Public IP アドレスを確認します。



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。

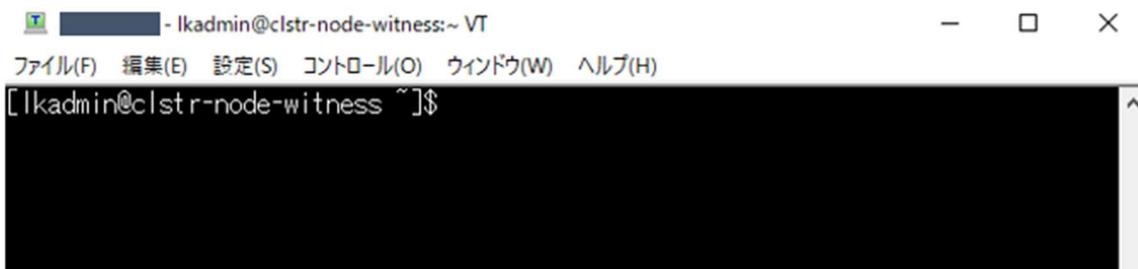


図 5.1-2 ssh ログイン時のプロンプト

5.2. 接続確認と X Server の有効化

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。

踏み台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
```

パスワードの入力では、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 ログインができたなら、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 X Server の有効化

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。

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、踏み台サーバーからクラスタースタートへの ssh 時に-X オプションを付けることで、クラスタースタート上の 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、以降クラスタースタートの操作のために 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。ノード A とノード B 上で、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 step-node
```

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

今回、OS イメージをローカルリポジトリとして設定する際に、インストールメディアを用意するため明示的に/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 -l /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。

踏み台ノードへの転送 (SCP を利用する場合)

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

踏み台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
```

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. 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 v9.8.0 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。

踏み台ノードへの転送 (SCP を利用する場合)

```
$ scp LKL_V980_092623.iso lkadmin@<Public_IP>:~
```

踏み台ノードからクラスターノードへの転送

```
$ scp LKL_V980_092623.iso lkadmin@10.3.1.11:~
```

```
$ scp LKL_V980_092623.iso lkadmin@10.3.1.12:~
```

7.2. 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_V980_092623.iso /mnt/iso/  
# mount /mnt/iso/sps_980.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。

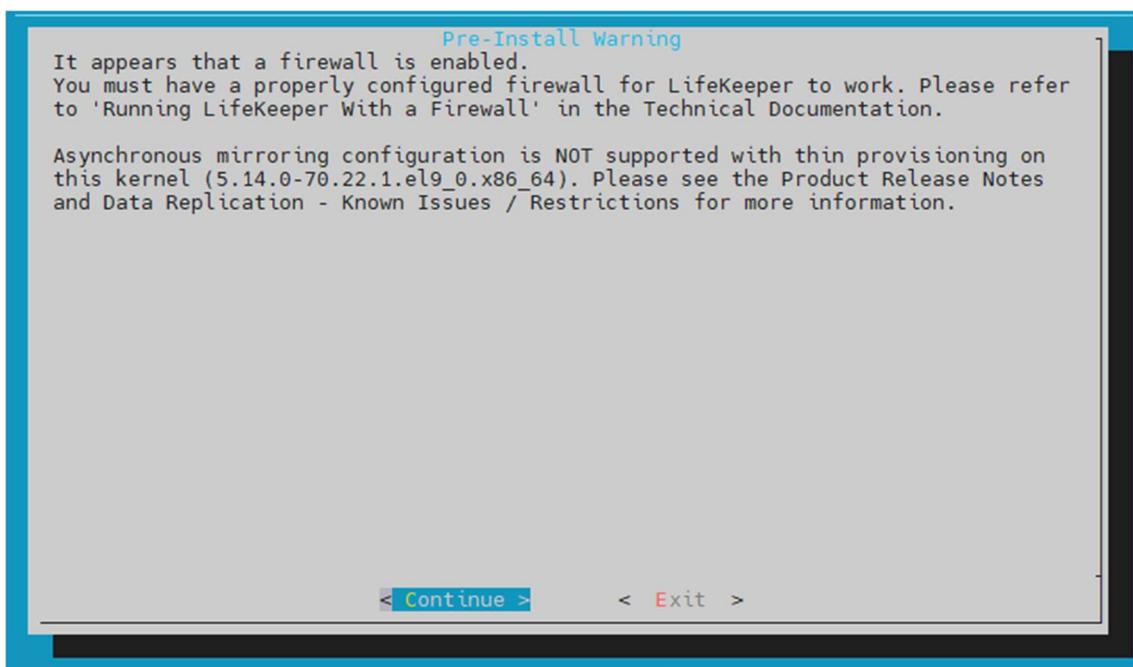


図 7.2-1 setup 実行画面

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。

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。

Default インストールに加えて、LB Health Check Kit, Scsi3, PostgreSQL Recovery Kit を選択する必要があります。

Recovery Kit Selection Menu に移動します。

インストール選択画面から LB Health Check Kit, Scsi3, PostgreSQL Recovery Kit を選択して追加インストールする kit の設定を行います。

LifeKeeper for Linux on Azure Shared Disk

Step by step Guide

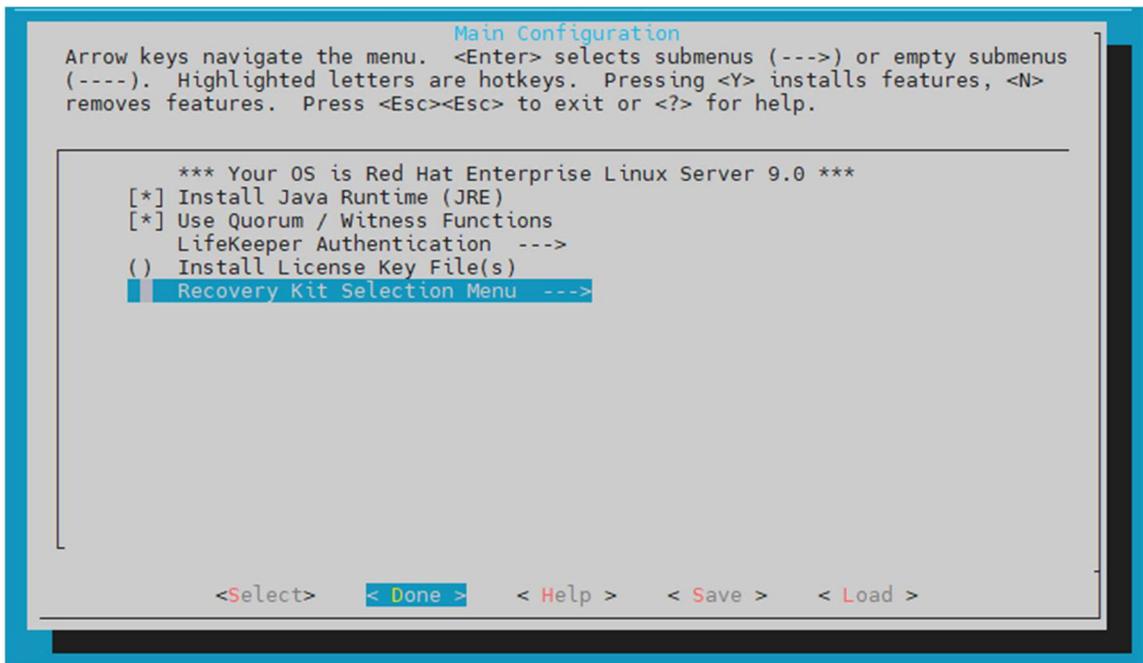


図 7.2-2 Recovery Kit Selection Menu を選択

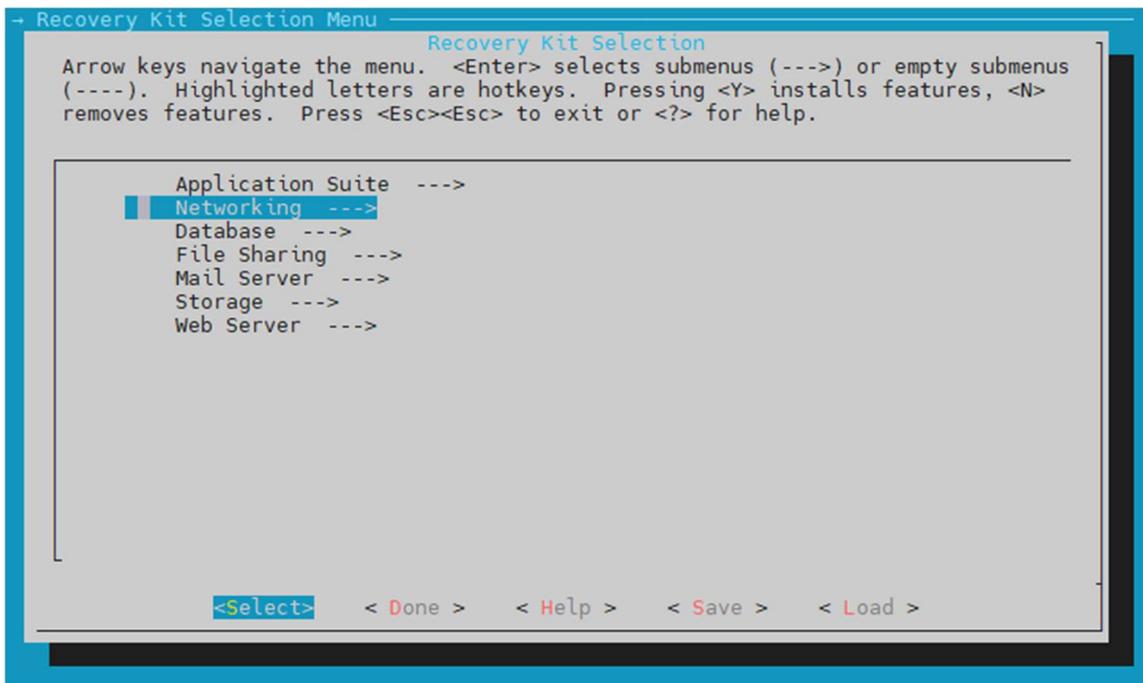


図 7.2-3 リカバリーキットのインストール選択画面

追加する kit の選択後、メインメニューの画面に戻り Done を選択し、インストールを開始します。

LifeKeeper for Linux on Azure Shared Disk Step by step Guide

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。

また、以下のコマンドを実行することで次回の 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

ノード A 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。

ログインに成功すると次のように GUI が表示されます。

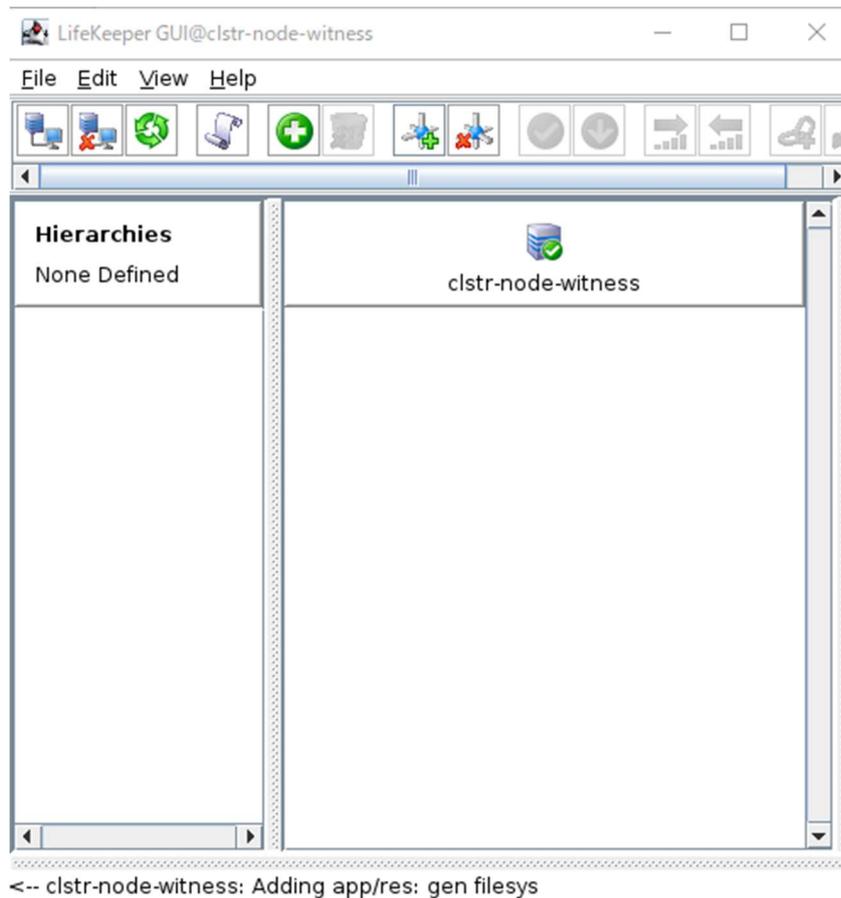


図 8.1-1 ログイン後の LifeKeeper GUI

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。

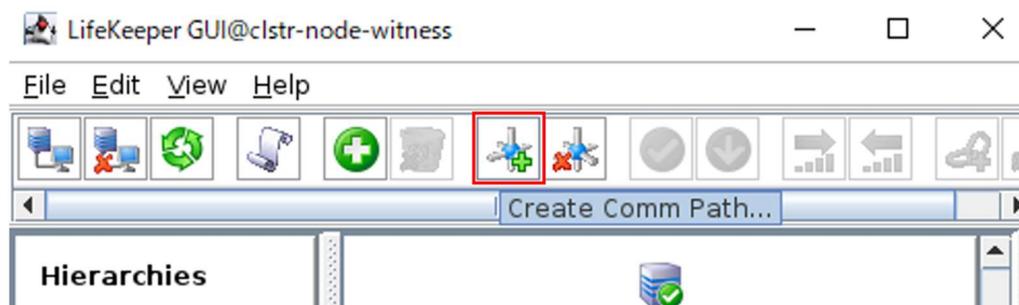


図 8.1-2 コミュニケーション作成ボタン

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-A
RemoteSeerver	clstr-node-B と入力し、Add
Device Type	TCP
Local IP Address(es)	10.3.1.11
Remote IP Address	10.3.1.12
Priority	1

正常に追加された場合、下記のコミュニケーションパスも追加で作成します。

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、ファイル共有リソースで保護し、PostgreSQL へのアクセスは仮想 IP アドレス 10.3.1.100 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。

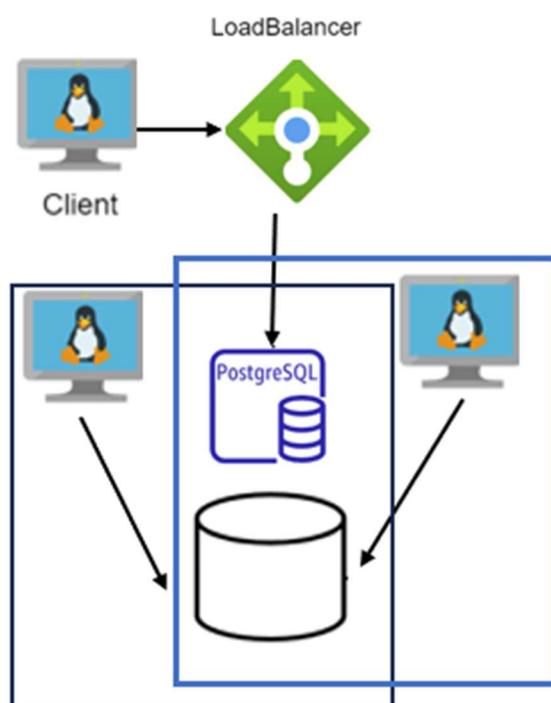


図 9-1 サービスのイメージ図

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、ファイル共有リソースは稼働系のディスクへの書き込みが行われた場合、共有ディスクを使用しているため、待機系に接続されたディスクにも書き込まれます。

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。

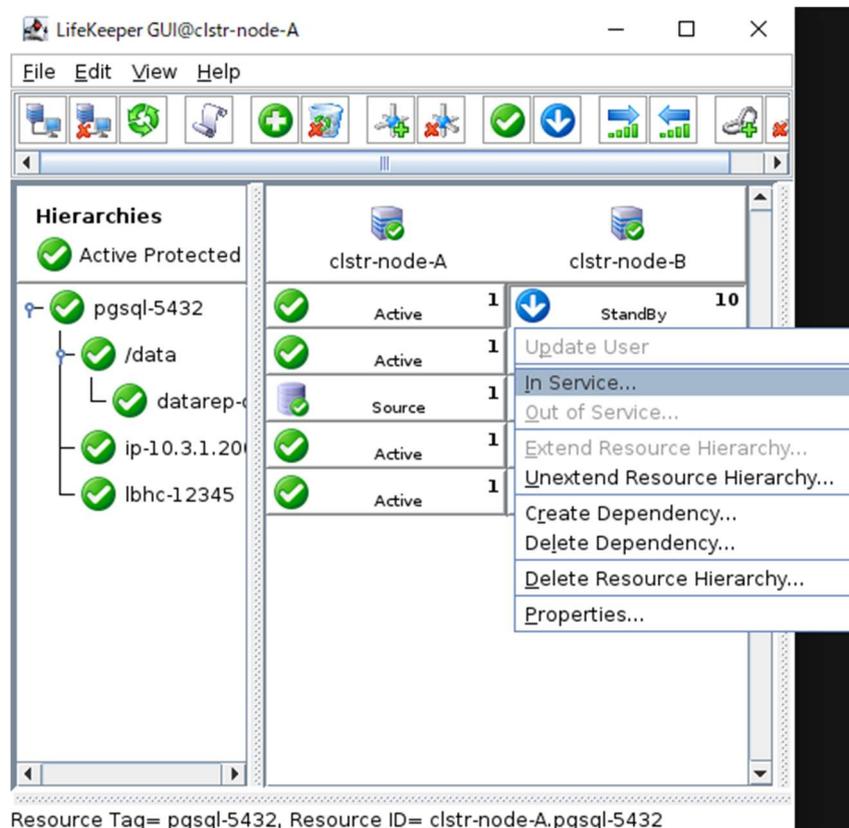


図 9.1-1 リソースのコンテキストメニュー

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 on Azure Shared Disk 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。

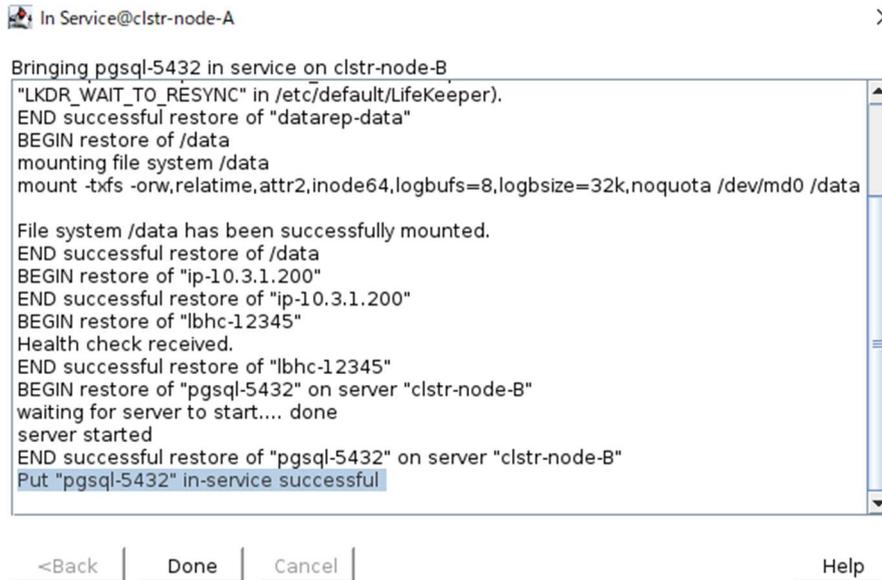


図 9.1-2 スイッチオーバーの終了

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。

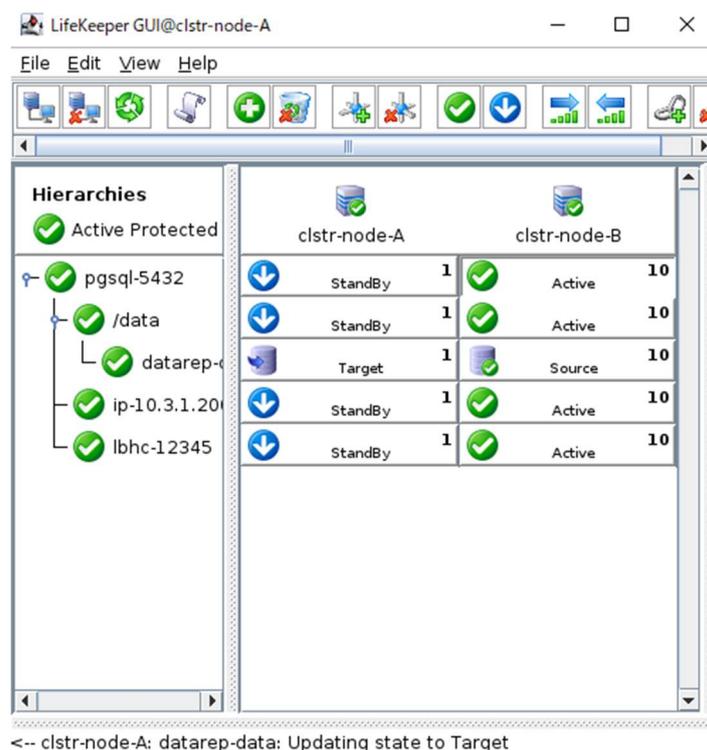


図 9.1-3 スイッチオーバー終了後の LifeKeeper GUI

以上でスイッチオーバーテストは終了となります。

9.2. ファイル共有リソースの作成

Azure 仮想マシンを作成した際に追加したディスクをファイル共有リソースの保護ファイルとして設定します。

① ディスクのパーティション分割とフォーマット

ノード A にて以下のコマンドを使用して追加したデータディスクを確認します。

```
# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0                               7:0    0  7.9G  1 loop /media/cdrom
sda                                  8:0    0   64G  0 disk
├─sda1                              8:1    0  200M  0 part /boot/efi
├─sda2                              8:2    0  500M  0 part /boot
├─sda3                              8:3    0    1M  0 part
└─sda4                              8:4    0  63.3G  0 part
   ├─rootvg-tmplv 253:0    0    2G  0 lvm  /tmp
   ├─rootvg-usrlv 253:1    0   10G  0 lvm  /usr
   ├─rootvg-homelv 253:2    0   17G  0 lvm  /home
   ├─rootvg-varlv 253:3    0    8G  0 lvm  /var
   └─rootvg-rootlv 253:4    0    2G  0 lvm  /
sdb                                  8:16   0    4G  0 disk
└─sdb1                              8:17   0    4G  0 part /mnt
sdc                                  8:32   0   16G  0 disk
```

追加したディスクは 16GiB のため、sdc についてパーティション分割とフォーマットを行います。まず、パーティション分割の作成をします。

```
# parted /dev/sdc
(parted) mklabel gpt
(parted) mkpart
Partition name? []? 空欄で enter
File system type? 空欄で enter
Start? 0%
End? 100%
```

LifeKeeper for Linux on Azure Shared Disk Step by step Guide

もう一度 lsblk コマンドを使用して sdc1 が作成されていることを確認します。

```
sdc          8:32  0  16G  0 disk
└─sdc1      8:33  0  16G  0 part
```

ノード B にてパーティションテーブルの更新をします。

```
# parted /dev/sdc
(parted) p
(parted) q
```

ノード A に戻り、ディスクをフォーマット、マウントを行います。

```
# mkfs.xfs /dev/sdc1 -f
# mount /dev/sdc1 /media/test
```

② 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。

③ Recovery Kit として File System を選択

④ 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	File System
New Mount Point	/media/test
Root tag	/media/test

Next を押下して Pre Extend ウィザードへ進みます。

⑤ 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。

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。

Next をクリックして Extend ウィザードに進めます。

⑥ 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	/media/test
Root Tag	/media/test

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。

Finish を押して、ファイル共有リソースの作成は以上になります。

⑦ スイッチオーバーテスト

正常にリソースが作成された後、/media/test 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。

```
# yum install postgresql-server
```

また、接続の確認を行う踏み台ノードでは、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。

```
# yum 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。

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ファイル共有リソースが保護しているファイル上に作成し、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 /media/test/pgsql/data  
# chown -R postgres:postgres /media/test/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 /media/test/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。

下記ファイルをエディターで開いて変更してください。

1. /media/test/pgsql/data/pg_hba.conf の設定

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. /media/test/pgsql/data/postgresql.conf の設定

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 /media/test/pgsql/data -o "-p 5432" -w
```

⑦ 起動確認と接続(稼働系と踏み台ノード)

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、踏み台ノード(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
```

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

ファイル共有リソースの作成時同様に、Create Resource Hierarchy を選択し、リソース作成ウィザードを起動します。

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	/media/test/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。

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。

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

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。

② Recovery Kit として 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
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。

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。

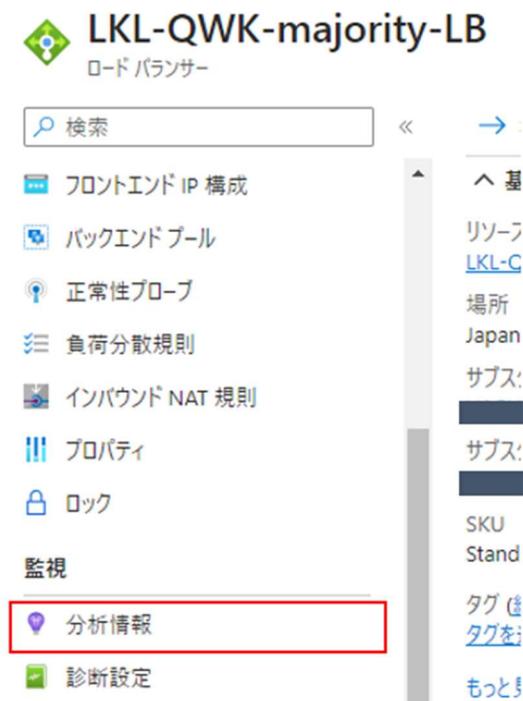


図 9.4-3 Azure ポータル ロードバランサーの概要画面

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。
Show Metrics Pane を選択、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

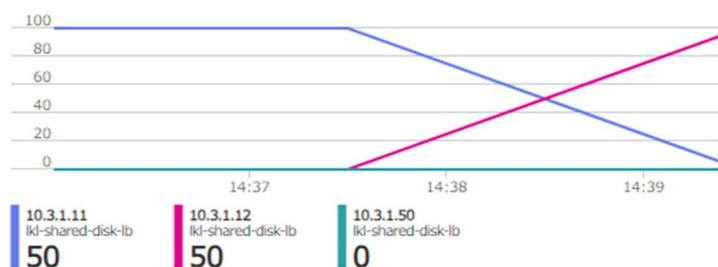


図 9.4-4 正常性プローブの確認

9.5. IP リソースの作成

- ① Create Resource Hierarchy を選択
- ② Recovery Kit として 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	10.3.1.100
Netmask	255.255.255.255
Network Interface	eth0
IP Resource Tag	ip-10.3.1.100

- 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	10.3.1.100
Netmask	255.255.255.255
Network Interface	eth0
IP Resource Tag	ip-10.3.1.100

- 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。

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。

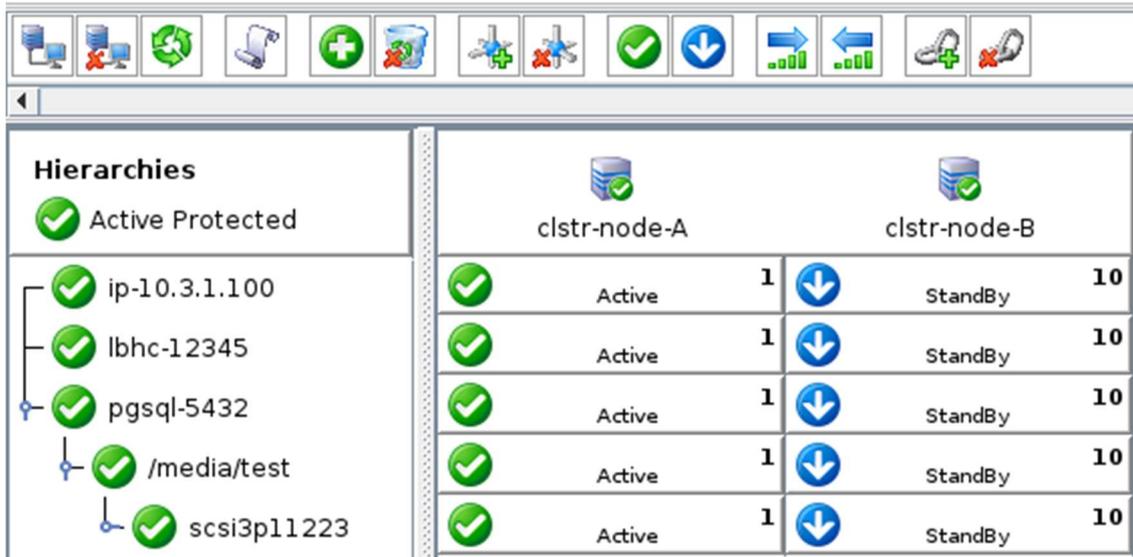


図 9.6-1 すべてのリソース作成後の LifeKeeper GUI

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ファイル共有リソースのほかに、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。

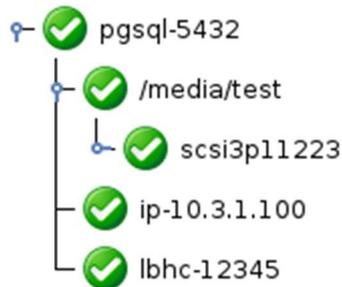


図 9.6-2 階層関係作成後のリソース構成図

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。

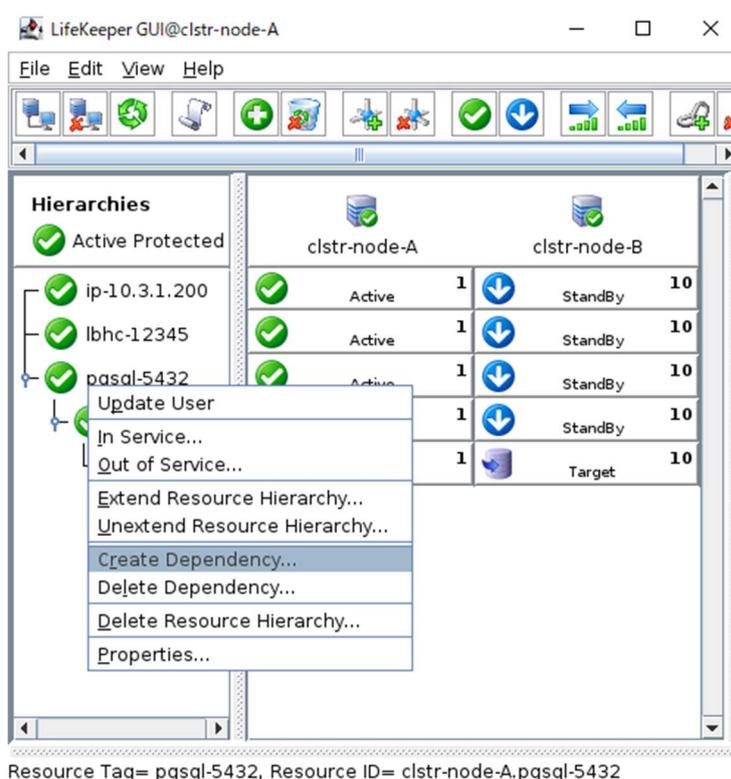


図 9.6-3 PostgreSQL リソースのコンテキストメニュー

② 子リソースを選択

子リソース(依存元)に IP リソース(ip-10.3.1.100)を選択します。

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

③ LB Health Check リソースの依存関係の作成

① から②の手順を再度繰り返し、LB Health Check リソースにも依存関係を作成します。

正常に作成されると依存関係の定義は終了です。

④ 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。

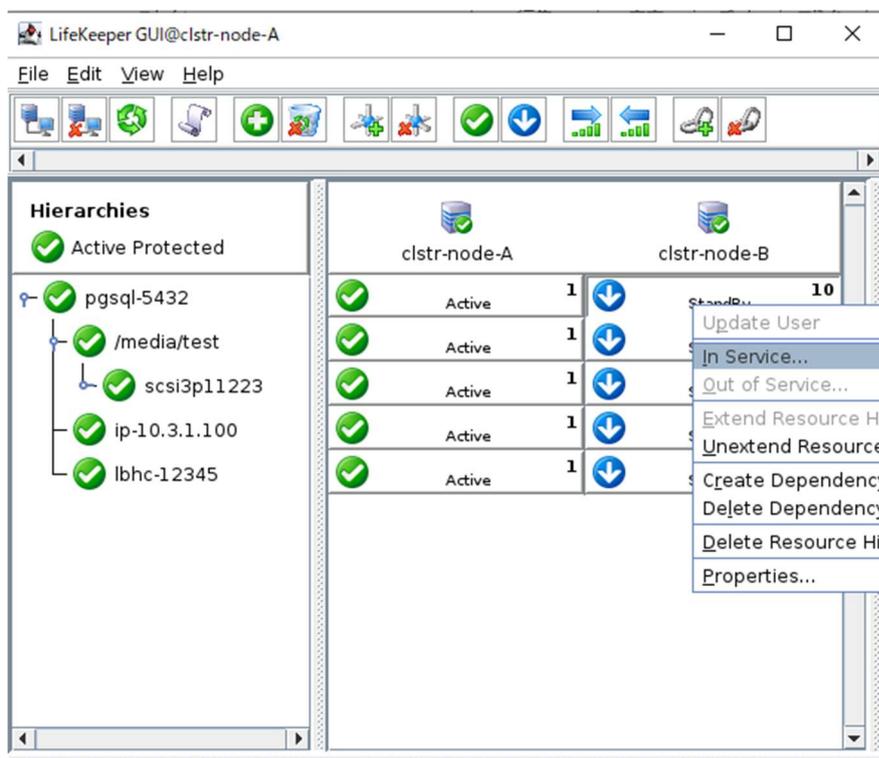


図 9.6-4 PostgreSQL リソースのコンテキストメニュー

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 10.3.1.100 -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. 動作検証テスト

今回の構成ではファイル共有システムを利用した I/O フェンシングという機能にてスプリットブレインの発生を防止します。これによって待機系によって共有領域にアクセスしても Reservation Conflict の発生からアクセスが拒否されます。

その他にも 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>

以下、動作検証テスト方法です。

① コミュニケーションパス障害

本テストのシナリオは以下の通りです。

コミュニケーションパス断線 (サービスの提供が不可能)

この場合、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 でリザーブダッシュが行われ、サービスの起動が行われます。
3. ノード A が強制的に再起動を行います。

② コミュニケーションパス障害テスト

実際にノード A のネットワークに障害を起こし、I/O フェンシングが行われることを確認します。

1. すべてのリソースの稼働をノード A に移行しておきます。
2. ノード A 上で以下のコマンドを実行します。

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

- I/O フェンシングの機能によってリザーブの奪取が行われるため、ノード B 上でサービスの起動が行われます。

GUI 上でノード B にてリソースが Active になっていれば成功です。

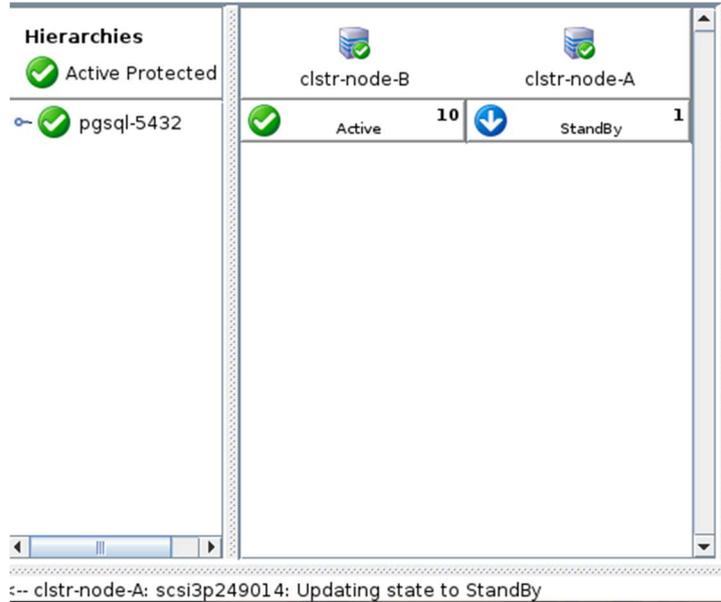


図 10-1 リザーブ奪取後の GUI 画面

- ノード A の ssh 接続が切れてノード A が再起動されていることを確認します。正しく再起動されていれば完了です。

11. 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>

12. 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>