

# SIOS Protection Suite for Linux

## Microsoft Azure クイックスタートガイド

---

第 2 版



## 目次

1 本書の内容 .....	4
将来における互換性 .....	4
制限事項.....	5
用語について .....	5
2 Microsoft Azure の概要 .....	6
3 構成情報 .....	7
サーバー構成 .....	7
ソフトウェア構成 .....	8
監視対象.....	8
ネットワーク構成 .....	9
4 Azure 上の LifeKeeper 特有の設定について .....	10
5 仮想マシンの構築、OS の起動 .....	13
リソースグループの作成 .....	13
仮想ネットワークの作成 .....	14
仮想マシンの作成 .....	16
ロードバランサーの作成 .....	34
OS の設定 .....	40
6 LifeKeeper による HA クラスターの構築 .....	49
LifeKeeper のインストールと起動 .....	49
コミュニケーションパスの作成 .....	50
Quorum/Witness の設定 .....	51
Broadcast Ping の無効化 .....	52
IP リソースの作成.....	52
Data Replication リソース階層の作成 .....	55
Oracle リソース階層の作成 .....	61
接続確認.....	84
7 Availability Zone(高可用性ゾーン) .....	88
7.1 動作確認内容.....	88
7.2 Azure の構成 .....	91
7.3 クライアントリダイレクトの動作確認方法 .....	94

8 参考資料 .....	95
9 謝辞.....	96
10 お問い合わせ.....	97
11 免責事項 .....	98

#### 改訂履歴

2020年4月28日	初版	
2020年8月28日	第2版	ILB への設定に関連する箇所に正常性プローブの設定を追加しました。

## 1 本書の内容

---

本書は、SIOS Protection Suite for Linux（以下、LifeKeeper）を Microsoft Azure（以下、Azure）リソースマネージャー環境においてご利用頂く際の参考として頂くために弊社で実際に行った作業手順をベースとしてクイックスタートガイドとしてまとめたものです。

本ガイドでは、Azure VM 上のクラスター環境の例として、以下の構成を作成する手順をご案内します。

- LifeKeeper クラスター（2 ノードクラスター） ※マルチ NIC 構成
- DataKeeper レプリケーションによるデータミラー型の共有データ領域
- IP リソース、Oracle リソース



**免責条項:** 本ガイドでは Microsoft Azure 環境における一般的な SIOS Protection Suite for Linux 設定の概略を述べています。そのため、本ガイドで使用している設定値は、説明に必要な項目を抜粋して記載しています。また、記載のない項目は基本的にデフォルト値または任意の値を前提としています。

実際には、より多くの Azure インフラストラクチャーオプションを利用した構成が可能です。本ガイドに記載の設定値は、実際の要件に合わせて、適宜置き換えてご利用ください。

## 将来における互換性

---

本文書に記載されております内容は、2020 年 8 月時点の LifeKeeper for Linux、および Microsoft Azure の仕様に基づいています。将来における Azure 及び LifeKeeper の仕様変更について保証するものではありません。必要に応じて最新のドキュメンテーションを参照し、適切な設定を行ってください。

## 制限事項

---

- マルチ NIC 構成は、「Predictable Network Interface Name」を利用可能な RHEL7、CentOS 7 でのみサポートします。
- RHEL5、CentOS 5、RHEL6、CentOS 6 ではシングル NIC 構成のみをサポートし、マルチ NIC 構成はサポートしていません。
- アクティブ/アクティブ構成はサポートしていません。
- 複数のポートを使用するサービスもしくは複数のサービスを提供する構成をサポートしていません。

## 用語について

---

本書で使用する用語について説明します。

- 内部負荷分散 (Internal Load Balancing または ILB)  
Azure クラウドサービスまたは地域が限定された仮想ネットワークの内側に存在する仮想マシン間に負荷分散を実現する機能です。
- [IP リソースが保護する VIP]  
LifeKeeper で作成した仮想 IP アドレスです。
- [ILB が設定する VIP]  
Azure で作成した Internal Load Balancing の仮想 IP アドレスです。
- [Azure のプライベート IP アドレス]  
Azure の Virtual Network で使用するプライベートアドレスです。

## 2 Microsoft Azure の概要

---

Azure は、Microsoft が提供するパブリッククラウドサービスです。ハードウェア、ネットワーク、ディスクなどのストレージ、Windows Server や Linux などのサーバーOS、Web サーバーや RDBMS などのミドルウェア、グループウェアやサーバー・アプリケーション、そして .Net Framework などのアプリケーション実行環境を、ユーザーがオンデマンドでネットワークを経由して利用可能な IT の利用形態です。

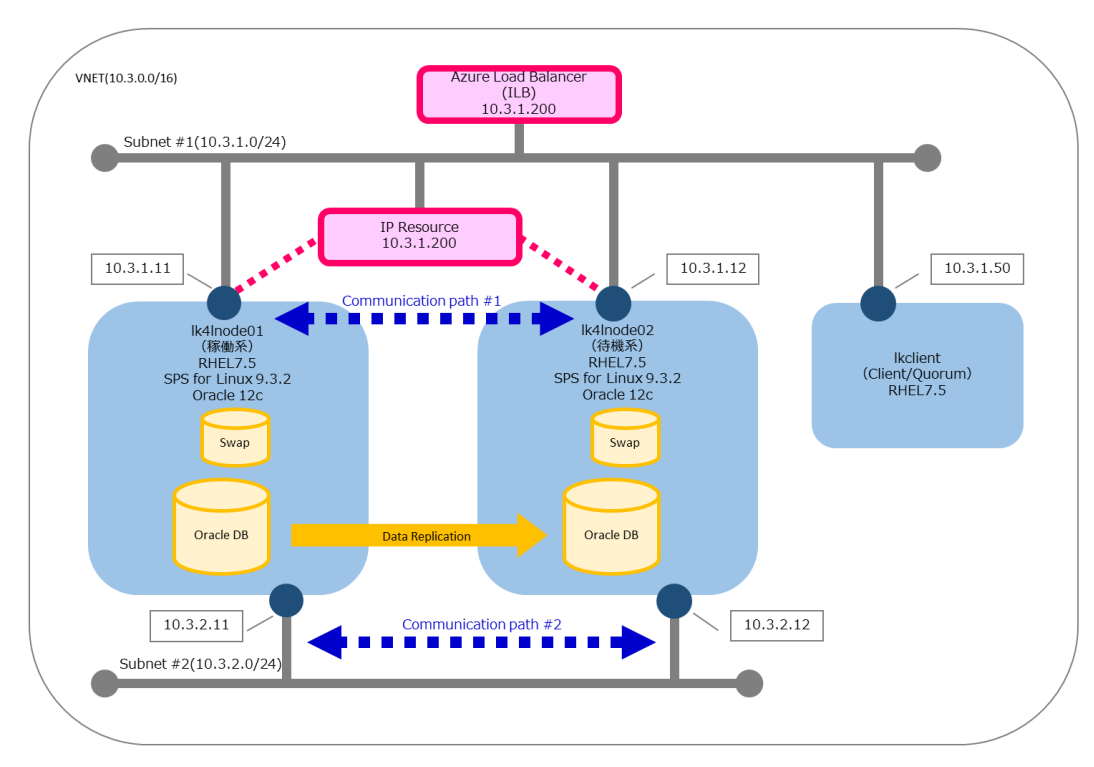
Microsoft が管理する世界中のデータセンターに、インターネット経由でサーバー・アプリケーションを簡単に作成し、展開、管理が行えます。サーバーやインフラを所有することなく、必要な時に必要な量のメモリやディスクといったコンピューティングリソースを利用することが出来ます。料金体系はリソース毎に分かれており、時間単位の従量課金での利用となります。

Azure では、Web ブラウザベースの管理ポータルを提供しており、ユーザーは直感的に操作する事が可能です。

Azure では、仮想マシン(VM)、Web サイト、モバイルサービス、クラウドサービスの4つの形態でサービスを提供していますが、本ドキュメントでは、仮想マシンに対して LifeKeeper を導入し、可用性を高める手法について紹介いたします。

### 3 構成情報

本構成では LifeKeeper を使って、以下の構成のアクティブ/スタンバイクラスターを構築します。マルチ NIC 構成の Azure VM に LifeKeeper を導入します。



全体構成図

### サーバー構成

仮想マシンサイズ	クラスタースタートノードは Standard_A1 (1 コア、1.75 GB メモリ) Client/Witness Server は Basic_A0 (0.25 コア、0.75GB メモリ) ※注 1
----------	---

データディスク	30GiB (Oracle DB 用) 10GiB (Swap 領域用) (sdb は未使用 ※注 2) ※注 1
[IP リソースが保護する VIP]	10.3.1.200
[ILB に設定する IP]	10.3.1.200
ILB が転送するポート	1521
ILB の負荷分散先ホスト およびポート	lk4lnode01/lk4lnode02 の各 1521 ポート
[Azure のプライベート IP アドレス]	クラスターノード (稼働系) 10.3.1.11 / 10.3.2.11 クラスターノード (待機系) 10.3.1.12 / 10.3.2.12 クライアント兼 Witness サーバー 10.3.1.50

※注 1 : Oracle のインストール要件 (メモリは 1 GB、Oracle データベース・インストールディスクサイズ、Swap エリアサイズ) に合うよう、インスタンスサイズおよび仮想マシンのディスクを用意してください。

※注 2 : Azure が初期構成でマウントして起動する /mnt/resource のデータディスクは、再起動毎にディスク ID が変更されます。その為、SIOS Protection Suite for Linux の共有ディスクとして利用するディスクには使用できません。そのため、本文書では別途ディスクの増設を行っています。

## ソフトウェア構成

OS	RedHat Enterprise Linux 7.5 64bit
LifeKeeper	SIOS Protection Suite for Linux 9.3.2
Oracle	Oracle Database 12c Enterprise Edition

## 監視対象

監視対象	IP リソースが保護する VIP / ファイルシステム / DataReplication / Oracle DB / Oracle Listener
------	--

## ネットワーク構成

---

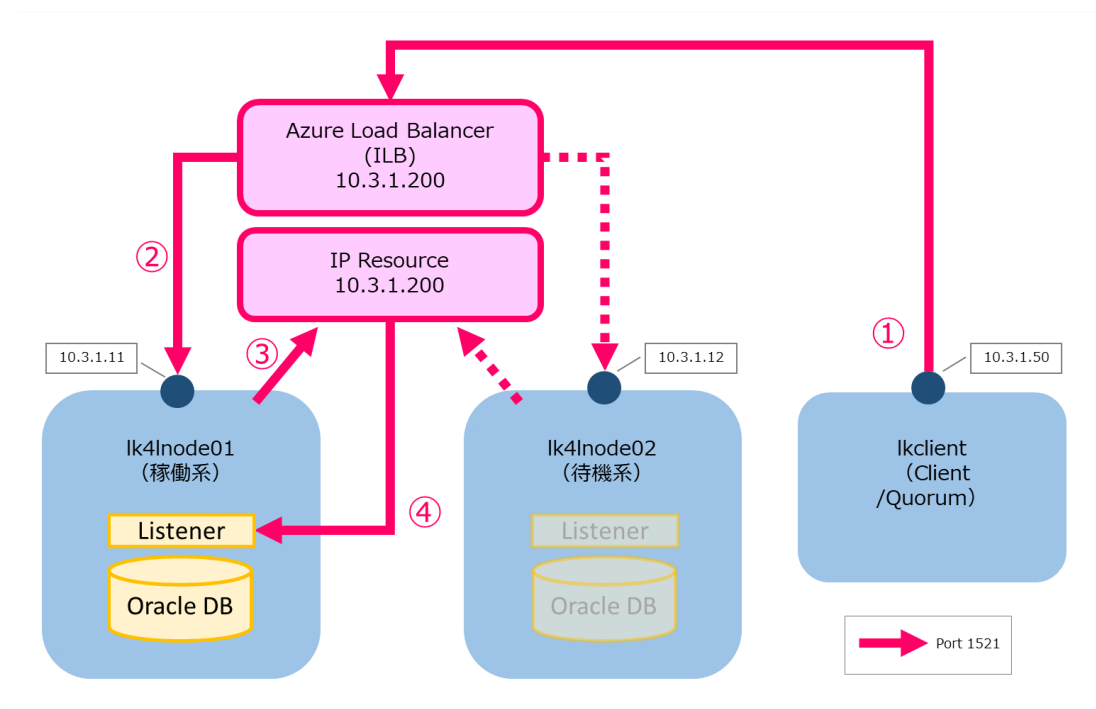
Azure では、仮想マシン (VM) 間での通信を行えるようにする為、“仮想ネットワーク” (VNET) を作成します。VNET は、サブネットを指定する事で、このサブネット内で VM 間での通信が行えるようにしています。

Azure で VM にネットワークを構成する場合、あらかじめ VNET を作成し、VM にこの VNET を指定するのが一般的です。VNET を使用すると、VM に仮想プライベートネットワーク (VPN) を提供できます。オプションとして、VPN をオンプレミス環境に接続して、ハイブリッドソリューションやクロスプレミスソリューションの実現もできます。VNET は管理ポータルサイトや Azure PowerShell を通して、DNS および IP アドレス範囲の構成を含むネットワークトポロジを制御できます。

## 4 Azure 上の LifeKeeper 特有の設定について

### 1. インターナルロードバランサー (ILB) の導入

Azure では[IP リソースが保護する VIP]を Azure の VNET 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 VIP]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、以下の様に ILB の導入を行い、[ILB が設定する VIP]をネットワーク通信経路として設定します。



① Oracle Clientが Oracle Listener へ接続する為、10.3.1.200(port1521)への接続を ILB 経由で開始します。

② ILB に負荷分散先 (backend pool) に 10.3.1.11 (稼働系ノード : Ik4Node01) と 10.3.1.12 (待機系ノード : Ik4Node02) が登録されています。ILB の正常性プローブに Oracle Listener の 1521 ポートを指定することで、ILB は 10.3.1.11 (稼働系ノード) と 10.3.1.12 (待機系ノード) の 1521 ポートのヘルスチェックを行います。Oracle Listener の 1521 ポートが open になっているのは

10.3.1.11（稼働系ノード）であるため、ILB が受け取った 10.3.1.200(port1521)宛のパケットは 10.3.1.11（稼働系ノード）へ転送されます。

③ LifeKeeper の仕様上 Oracle Listener リソースは[IP リソースが保護する VIP]を指定する必要があります。その為、受信用に IP リソースを作成します。LifeKeeper の IP リソース（10.3.1.200）が In-Service になっているのは稼働系のみとなる為、実際には 稼働系（10.3.1.11）のみリクエストを受け取ります。

④ 結果として Oracle Client からの接続要求は稼働系の Oracle Listener（10.3.1.200 : port1521）を受け取ります。

## 2. アドレス変換

LifeKeeper では、IP リソースの監視処理に外部ホストへの PING を使用しますが、Azure の仕様上、仮想 IP を送信元とする PING 処理は行えません。その為、以下のようにパケットの送信元情報を変更する設定を行うことで、PING 処理を有効にし、外部ホスト間とのネットワークの障害を検出可能にします。



① IP リソースから IP リソースで保護する VIP をソースとして指定した PING を開始します。

② PING の送信元アドレスを、10.3.1.200(IP リソースで保護する VIP)から 10.3.1.11(Azure のプライベート IP アドレス)に置き換えます。

③ ソース置き換え後、PING の送信先にパケットを送信します。[Azure のプライベート IP アドレス]に置き換えられる事で、ネットワークが正常であれば応答が返ります。

## 3. Quorum/Witness の導入

Azure 環境上では、クラスターの共有ディスクの代替として DataKeeper によるレプリケーションディスクを使用します。DataKeeper を使用したクラスターでは、コミュニケーションパスが全断するようなネットワークの障害が発生した際、その仕組み上スプリットブレイン状態になる可能性があります。これを回避するため、Quorum/Witness 機能による I/O Fencing を設定してください。動作確認は TCP Remote モードによる正常動作を確認しています。下記 URL 中のダウンロードリンクにある PDF ファイル“4.1\_2) TCP Remote モード”をご参照ください。

LifeKeeper for Linux Quorum Check/Witness Server 機能利用ガイド

[http://sios.jp/products/lkdk/product/pdf/lifekeeper\\_quorum-witness.pdf](http://sios.jp/products/lkdk/product/pdf/lifekeeper_quorum-witness.pdf)

また、以下のテクニカルドキュメンテーションをご参照ください。

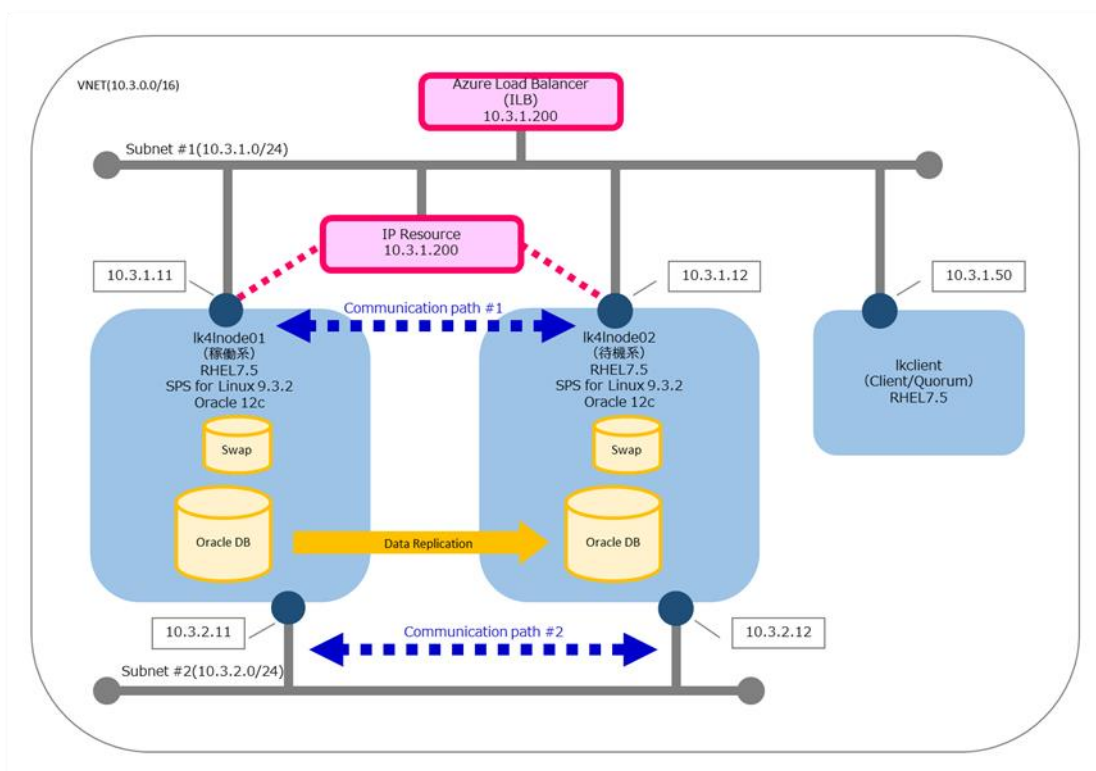
<http://docs.us.sios.com/spslinux/9.5.0/ja/topic/quorum-witness>

## 5 仮想マシンの構築、OS の起動

Azure 上に仮想マシンを作成します。

前述の通り、Azure で LifeKeeper を使用する場合、Azure 環境に特有な設定（以下 3 点）を行う必要があります。

- 内部負荷分散（Internal Load Balancing または ILB）を使用し、LifeKeeper で保護する Oracle リソースへの接続を有効にする。
- 仮想 IP アドレスを使用した通信のため、アドレス変換設定を行う。
- スプリットブレインを回避するため Quorum の設定を行う。



全体構成図（再掲）

## リソースグループの作成

1. Azure Portal > [Resource groups] > [Add]を選択し、本環境で使用するリソースグループを作成します。必要な値を入力し、[Review + Create]をクリックします。

Dashboard > Resource groups > Create a resource group

## Create a resource group

Basics Tags Review + Create

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. [Learn more](#)

PROJECT DETAILS

\* Subscription ⓘ

Resource group ⓘ

RESOURCE DETAILS

\* Region ⓘ

Japan East

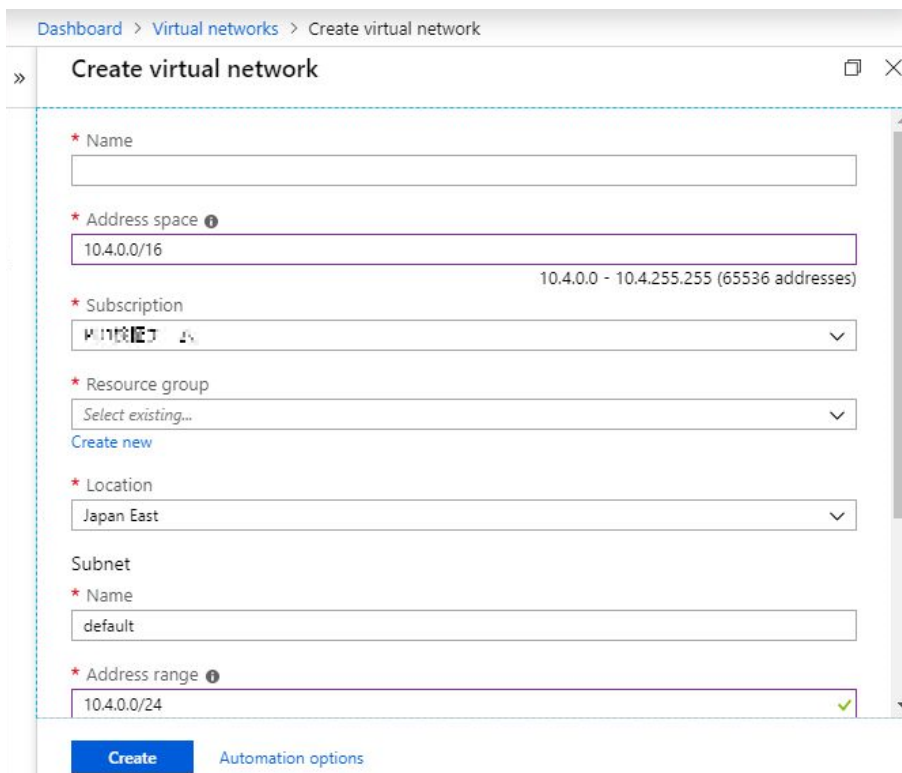
Review + Create Next: Tags

2. 内容を確認し、[Create]をクリックします。

## 仮想ネットワークの作成

本環境で使用する仮想ネットワークを作成します。今回は、アドレススペース1つ、サブネット2つを作成します。このうちひとつのサブネットはサービス用、もうひとつのサブネットは DataKeeper レプリケーション用になります。

1. Azure Portal > [Virtual networks] > [Add] を選択し、本環境で使用する仮想ネットワークと、同時にサブネットを1つ作成します。



2. 必要な値を入力し、[Create]をクリックします。

今回は以下の値を使用します。

項目	入力もしくは選択する値	
Name	lk4l-vnet	
Address space	10.3.0.0/16	
Subnet		
Name	lk4l-nw01	
Address range	10.3.1.0/24	
Firewall	Disabled(default)	

3. Azure Portal > [Virtual networks] > [(仮想ネットワーク名)] > [Subnets] > [Add] を選択し、仮想ネットワークに対して2つめのサブネットを追加します。

4. 必要な値を入力し、[OK]をクリックします。

今回は以下の値を使用します。

項目	入力もしくは選択する値	
Name	lk4l-nw02	
Address range	10.3.2.0/24	

## 仮想マシンの作成

本環境で使用する仮想マシンを作成します。

作成する仮想マシンは以下の3台です。

- クラスタースタートアップ ノード (稼働系)
- クラスタースタートアップ ノード (待機系)
- クライアント兼 Witness サーバー

## クラスターノード（稼働系）の作成

1. Azure Portal > [Virtual machines] > [Add]を選択します。

はじめに、基本設定を入力します。

Dashboard > Virtual machines > Create a virtual machine

### Create a virtual machine

Basics | Disks | Networking | Management | Advanced | Tags | Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image.  
Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.  
Looking for classic VMs? [Create VM from Azure Marketplace](#)

**PROJECT DETAILS**

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

\* Subscription

\* Resource group  [Create new](#)

**INSTANCE DETAILS**

\* Virtual machine name

[Review + create](#) [Previous](#) [Next: Disks >](#)

今回は以下の値を使用します。

Basics		
項目	入力もしくは選択する値	
INSTANCE DETAILS		
Virtual machine name	lk4lnode01	
Availability options	Availability set	※注 1
	(create new)lk4lcluster	
Image	Red Hat Enterprise Linux 7.5	※注 2
Size	Standard A1	※注 3 ※注 4
ADMINISTRATOR ACCOUNT		
Authentication type	Password	

Username	lkadmin	
Password	XXXXXXXX	
INBOUND PORT RULES		
Public inbound ports	Allow selected ports	
Select inbound ports	SSH	

※注 1 : Availability Set は、Azure 上の LifeKeeper クラスターの前提です。必ず設定してください。

※注 2 : 導入する LifeKeeper がサポートする OS バージョンを選択してください。LifeKeeper がサポートする OS バージョンは、各バージョンのサポートマトリクス (<http://jpdocs.us.sios.com/>) を参照ください。

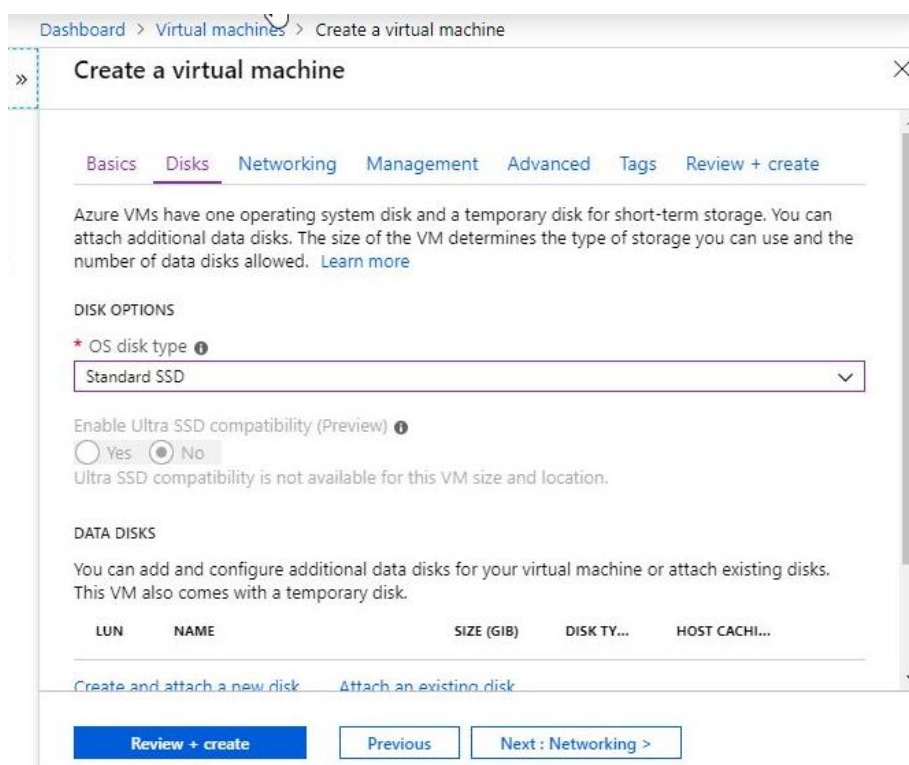
※注 3 : 必ず Availability Set が利用できるインスタンスサイズを選択してください。一部のインスタンスサイズでは Availability Set が利用できません。

※注 4 : 必ず負荷分散規則が利用できるインスタンスサイズを選択してください。一部のインスタンスサイズでは、負荷分散規則が利用できません。

2. 必要な値を入力したら、[Next:Disks] をクリックします。

3. 次にディスク設定を入力します。

今回の環境では、OS ディスクに加えて Oracle DB 用のディスク (30GiB) と Swap 領域用のディスク(10GiB)が必要なため、[Create and attach a new disk]をクリックして追加のディスクをアサインします。



今回は以下の値を使用します。

Disks		
項目	入力もしくは選択する値	
DISK OPTIONS		
OS Disk type	Standard SSD	
DATA DISKS(1st)		
LUN	0	
NAME	(create and attach a new disk) lklnode01_DataDisk_0	
Size(GiB)	30	
DISK TYPE	Standard SSD	
HOST CACHING	None	※注 1
DATA DISKS(2nd)		
LUN	1	
NAME	(create and attach a new disk) lklnode01_DataDisk_1	任意の入力値

Size(GiB)	10	
DISK TYPE	Standard SSD	
HOST CACHING	None	※注 1

※注 1 : 本環境ではデータベース用途のため、キャッシュを無効化しましたが、要件に応じて適切に設定してください。

4. 必要な値を入力したら、[Next:Networking] をクリックします。

5. 次にネットワーク設定を入力します。

The screenshot shows the 'Create a virtual machine' configuration page in the Azure portal. Under the 'NETWORK INTERFACE' section, the following settings are visible:

- Virtual network:** lk4l-vnet
- Subnet:** lk4l-nw01 (10.3.1.0/24)
- Public IP:** None
- NIC network security group:** Basic (selected)

At the bottom of the configuration area, there are three buttons: 'Review + create' (highlighted in blue), 'Previous', and 'Next: Management >'.

このとき、前の手順で作成した1つめのサブネットを、仮想マシンに対して設定します。

この設定により、ネットワークインターフェースがひとつアタッチされます。

(2つ目のネットワークインターフェースは、仮想マシン作成後に設定します。)

今回は以下の値を使用します。

Networking		
項目	入力もしくは選択する値	
Virtual network	lk4l-vnet	

Subnet	lk4l-nw01(10.3.1.0/24)	※注 1
Public IP	(create new) Name:lk4lnode01pubip SKU:Basic Assignment:Static	※注 2
NIC network security group	Basic	
Public inbound ports	Allow selected ports	
selected inbound ports	SSH	

※注 1 : 前の手順で作成した 1 つめのサブネットを、仮想マシンに対して設定します。

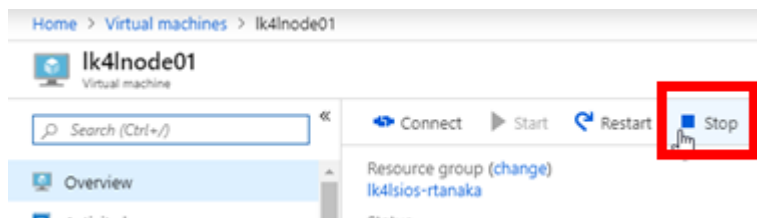
※注 2 : 本手順ではインターネット経由でアクセスするため、Public IP を設定しています。

6. 必要に応じて管理設定、詳細設定、タグ設定を行い、最後に[Review + create]をクリックします。

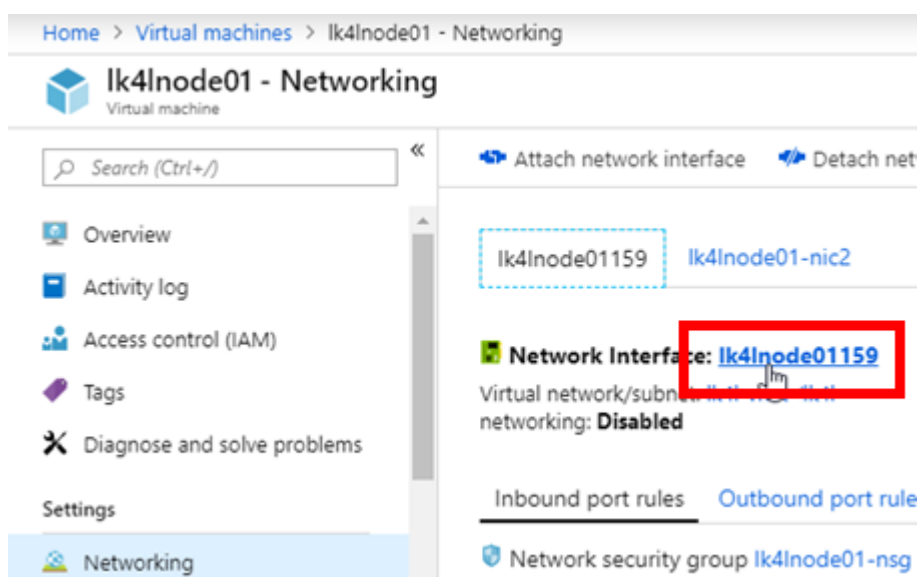
7. 内容を確認し、[Create]をクリックします。

上記の構成でデプロイすると、Redhat Enterprise Linux のインストールが完了した状態で起動します。

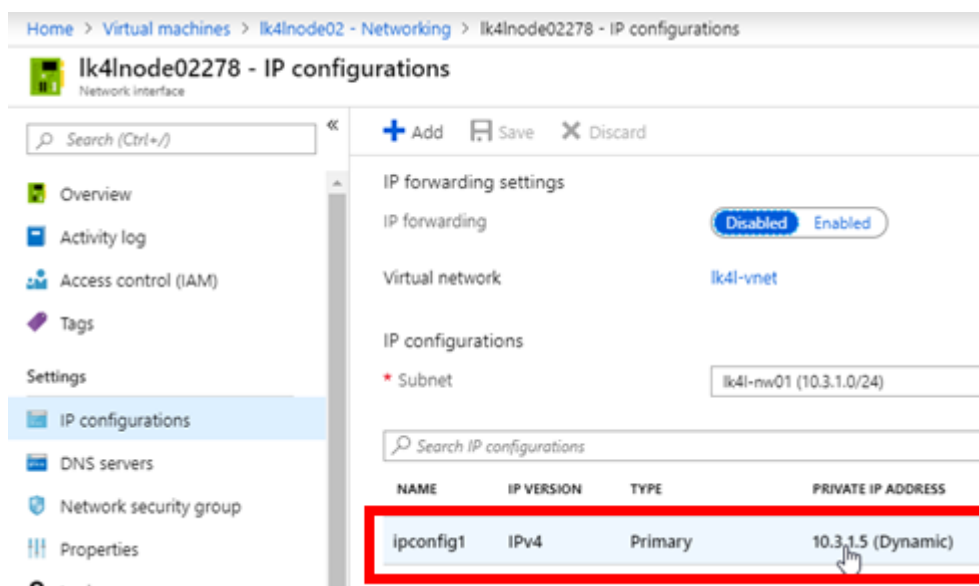
8. このあとネットワーク設定変更を行うため、一旦仮想マシンを停止します。



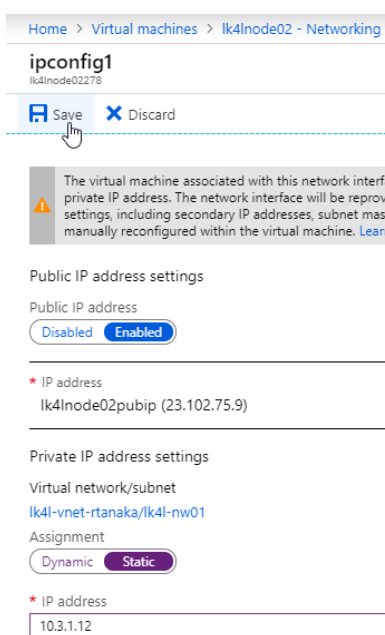
9. 1 つめのネットワークインターフェースを静的 IP アドレスに変更します。Azure Portal > [Virtual machines] > [(仮想マシン名)] > [Networking] > [ ( 1 つめのインターフェース名) ] をクリックします。



10. [IP configurations] > [ipconfig1]をクリックします。この時点では、IP アドレスの設定が動的(Dynamic)になっています。



11. IP アドレスを静的に設定変更します。

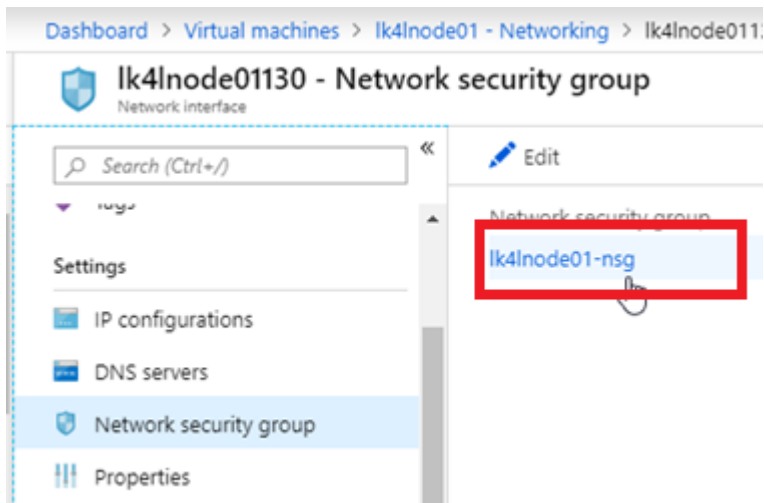


今回は以下の値を使用します。

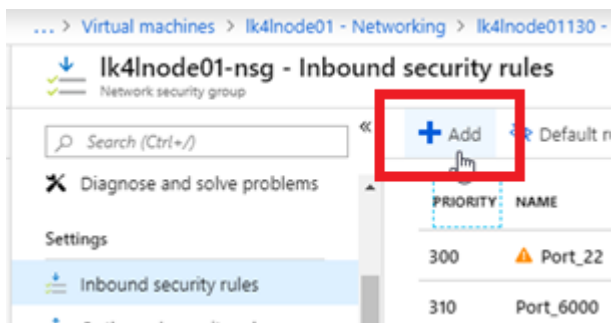
項目	入力もしくは選択する値	
Assignment	Static	
IP address	10.3.1.11	

12. [Save]をクリックして設定を保存します。

13. Oracle リスナーサービスで、TCP Port 1521 を使用して通信できるようにするため、ネットワークセキュリティグループを編集します。Azure Portal > [Virtual machines] > [(仮想マシン名)] > [Networking] > [ (1つめのインターフェース名) ] > [Network security group]>[(セキュリティグループ名)]をクリックします。



14. [Inbound security rules]>[Add]をクリックします。



15. インバウンドルールを追加します。

**Add inbound security rule**  
lk4lnode01-nsg

Basic

\* Source ?  
Any

\* Source port ranges ?  
\*

\* Destination ?  
Any

\* Destination port ranges ?  
8080

\* Protocol  
Any TCP UDP

\* Action  
Allow Deny

\* Priority ?  
350

\* Name

Add

以下の値を使用します。

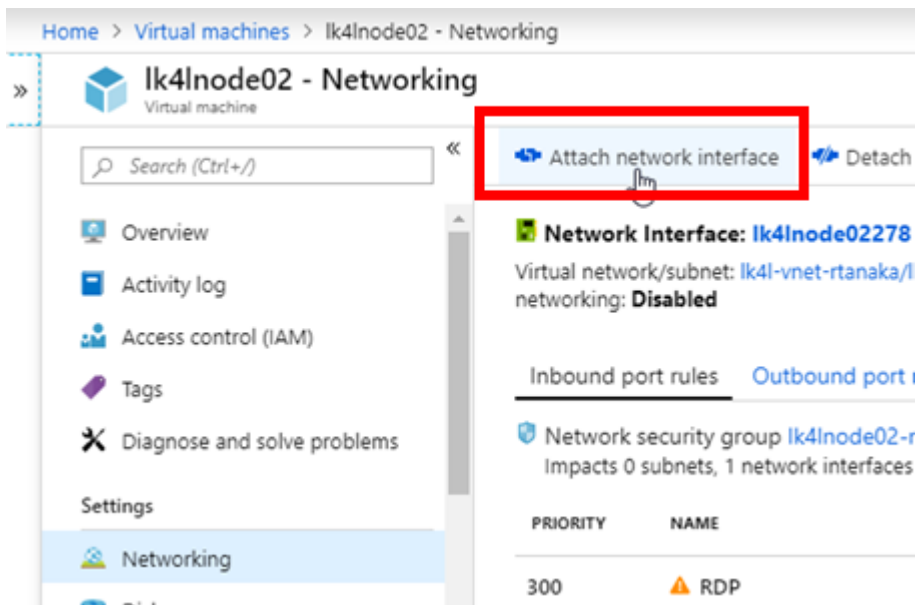
項目	入力もしくは選択する値	
Source	Any	
Source port ranges	*	
Destination	Any	
Destination port ranges	1521	※注 1
Protocol	TCP	
Action	Allow	
Name	Port_1521	

※Oracle リスナーが port1521 を使用するため、該当のポートへの通信を許可します。

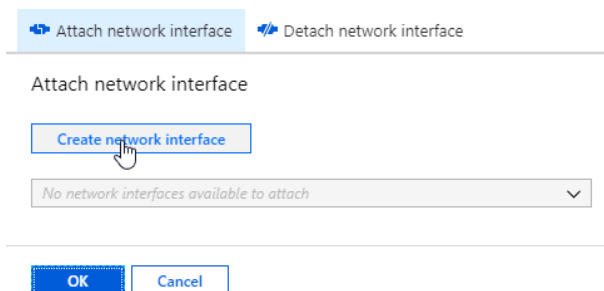
16. [Add]をクリックし、設定を保存します。

17. 次に、2つ目のネットワークインターフェースを追加します。Azure Portal > [Virtual machines] > [(仮想マシン名)] > [Networking] > [Attach network

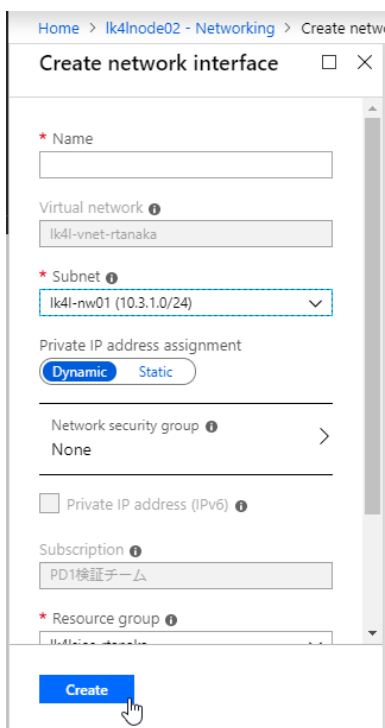
interface] をクリックします。



18. Create network interface をクリックします。



19. 必要な値を入力したら、[Create] をクリックします。



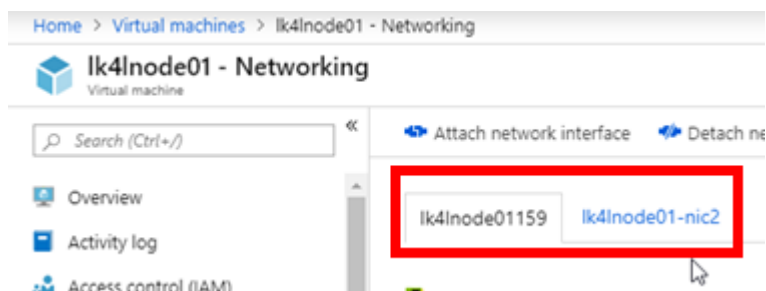
以下の値を使用します。

項目	入力もしくは選択する値	
Name	lk4lnode02-nic2	
Subnet	10.3.2.0/24	※注 1
Private IP address assignment	Static	
Private IP address	10.3.2.11	
Network security group	None	

※注 1 : 前の手順で作成した 2 つめのサブネットを、仮想マシンに対して設定します。

20. [OK] をクリックします。

21. インターフェイスが 2 つアタッチされていることを確認します。



22. 仮想マシンを起動します。

## クラスターノード（待機系）の作成

1. 同様の手順で、クラスターノード（待機系）に使用する仮想マシンを作成します。基本設定は、以下の値を使用します。

Basics		
項目	入力もしくは選択する値	
INSTANCE DETAILS		
Virtual machine name	lk4lnode02	

その他の設定はすべて稼働系と同じにしてください。

ディスク設定は、以下の値を使用します。

Disks		
項目	入力もしくは選択する値	
DATA DISKS(1st)		
LUN	0	
NAME	(create and attach a new disk) lk4lnode02_DataDisk_0	
DATA DISKS(2nd)		
LUN	1	
NAME	(create and attach a new disk) lk4lnode02_DataDisk_1	任意の入力値

その他の設定はすべて稼働系と同じにしてください。

ネットワーク設定は、以下の値を使用します。

Networking		
項目	入力もしくは選択する値	
Public IP	(create new) Name:lk4lnode02pubip SKU:Basic Assignment:Static	※注 1

※注 1：本手順ではインターネット経由でアクセスするため、Public IP を設定しています。

その他の設定はすべて稼働系と同じにしてください。

- 必要に応じて管理設定、詳細設定、タグ設定を行い、最後に[Review + create]をクリックします。
- 内容を確認し、[Create]をクリックします。
- このあとネットワーク設定変更を行うため、一旦仮想マシンを停止します。
- 1 つめのネットワークインターフェースを静的 IP アドレスに変更します。Azure Portal > [Virtual machines] > [(仮想マシン名)] > [Networking] > [ (1 つめのインターフェース名) ] をクリックします。
- [IP configurations] > [ipconfig1]をクリックします。この時点では、IP アドレスの設定が動的(Dynamic)になっています。

7. IP アドレスを静的に設定変更します。

以下の値を使用します。

項目	入力もしくは選択する値	
Assignment	Static	
IP address	10.3.1.12	

8. [Save]をクリックして設定を保存します。

9. 次に、2つ目のネットワークインターフェースを追加します。Azure Portal > [Virtual machines] > [(仮想マシン名)] > [Networking] > [Attach network interface] をクリックします。

10. Create network interface をクリックします。

11. 必要な値を入力したら、[Create] をクリックします。

以下の値を使用します。

項目	入力もしくは選択する値	
Name	lk4lnode02-nic2	
Subnet	10.3.2.0/24	※注 1
Private IP address assignment	Static	
Private IP address	10.3.2.12	
Network security group	None	

※注 1 : 前の手順で作成した 2 つめのサブネットを、仮想マシンに対して設定します。

12. [OK] をクリックします。

13. インターフェイスが2つアタッチされていることを確認します。

14. 仮想マシンを起動します。

## クライアント兼 Witness サーバーの作成

1. 上記と同様の手順で、クライアント兼 Witness サーバーに使用する仮想マシンを作成します。

基本設定として、主に以下の値を使用します。

Basics		
項目	入力もしくは選択する値	
INSTANCE DETAILS		
Virtual machine name	lk4lclient	
Availability options	No Infrastructure redundancy required	※注 1
Image	Red Hat Enterprise Linux 7.5	
Size	Basic A0	
INBOUND PORT RULES		
Public inbound ports	Allow selected ports	
Select inbound ports	SSH	

※注 1 : クラスタード以外は Availability Set の設定は必須ではありませんが、今回のようにクライアントが Quorum に利用される場合、可用性を考慮して Availability Set に含めることも検討してください。

必要な値を入力したら、[Next:Disks >] をクリックします。

2. 次にディスク設定を入力します。

今回はディスクの追加はしません。 [Nest:Networking >]をクリックします。

3. 次にネットワーク設定を入力します。

以下の値を使用します。

Networking		
項目	入力もしくは選択する値	
Virtual network	lk4l-vnet	
Subnet	lk4l-nw01(10.3.1.0/24)	※注 1
Public IP	(create new) Name:lk4lclientpubip SKU:Basic Assignment:Static	※注 2
NIC network security group	Basic	
Public inbound ports	Allow selected ports	
selected inbound ports	SSH	

※注 1 : 前の手順で作成した 1 つめのサブネットを、仮想マシンに対して設定します。

※注 2 : 本手順ではインターネット経由でアクセスするため、Public IP を設定しています。

4. 必要に応じて管理設定、詳細設定、タグ設定を行い、最後に[Review + create]をクリックします。

5. 内容を確認し、[Create]をクリックします。

6. 仮想マシンが起動したら、1 つめのネットワークインターフェースを静的 IP アドレスに変更します。Azure Portal > [Virtual machines] > [(仮想マシン名)]

>[Networking] > [ (1つめのインターフェース名) ] をクリックします。

7. [IP configurations] > [ipconfig1]をクリックします。この時点では、IP アドレスの設定が動的(Dynamic)になっています。

8. IP アドレスを静的に設定変更します

以下の値を使用します。

項目	入力例	
Assignment	Static	
IP address	10.3.1.50	

9. [Save]をクリックして設定を保存します。

10. 以上で、三台の仮想マシンが作成されました。

The screenshot shows the Azure portal interface for 'Virtual machines'. The breadcrumb path is 'Home > Virtual machines'. The page title is 'Virtual machines' with the subtitle '既定のディレクトリ'. Below the title, there are navigation options: '+ Add', 'Reservations', 'Edit columns', 'Refresh', and 'Assign tags'. The 'Subscriptions' section shows 'lk4l' selected. Below that, there are filters for 'All resource groups' and 'All types'. A table lists 3 items:

<input type="checkbox"/>	NAME ↑↓	TYPE ↑↓	STATUS	RESOURCE
<input type="checkbox"/>	lk4lclient	Virtual machine	Running	lk4lclient
<input type="checkbox"/>	lk4lnode01	Virtual machine	Running	lk4lnode01
<input type="checkbox"/>	lk4lnode02	Virtual machine	Running	lk4lnode02

The table content is highlighted with a red border in the original image.

## ロードバランサーの作成

本環境で使用するロードバランサーを作成します。

Azure 上では、そのままでは LifeKeeper で作成した仮想 IP による通信が行えません。仮想 IP を実際に機能させるためには、ロードバランサーが必要です。

Azure のロードバランサーには外部ロードバランサーと内部ロードバランサーが存在しますが、今回の用途では、内部ロードバランサーを使用します。また、この内部ロードバランサーの IP アドレスの値は、後に設定する仮想 IP アドレスの値としても使用され、保護対象サービスのアクセスポイントとなります。

1. Azure Portal > [Load balancers] > [Add] を選択し、本環境で使用するロードバランサーを作成します。必要な値を入力し、[Review + Create]をクリックします。

2. 以下の値を使用します。

項目	入力もしくは選択する値	
INSTANCE DETAILS		
Name	lk4lsiosilb	
Type	Internal	※注 1
SKU	Basic	
PUBLIC IP ADDRESS		
Virtual network	lk4l-vnet	
Subnet	lk4l-nw01(10.3.1.0/24)	
IP address assignment	Static	
Private IP address	10.3.1.200	※注 2

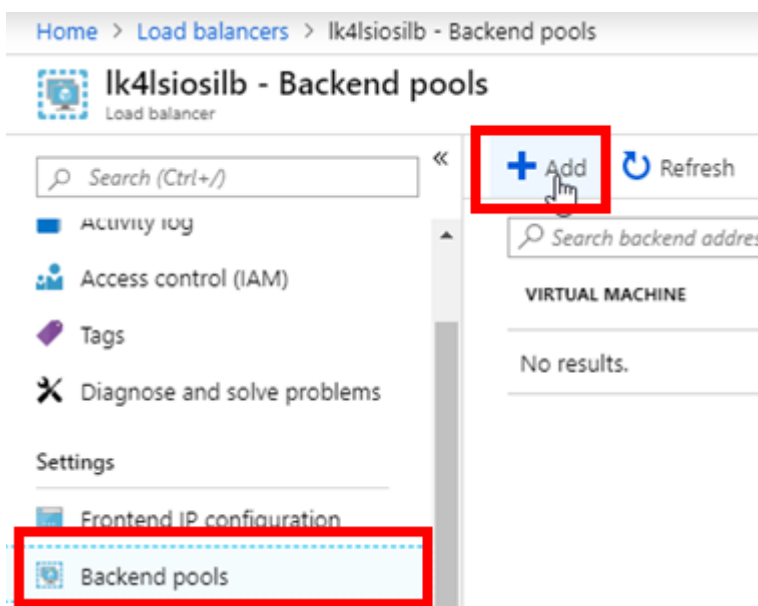
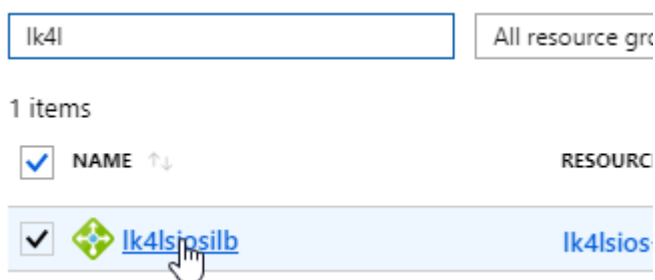
※注 1 : 内部ロードバランサーを作成するため、必ず「Internal」を選択します。

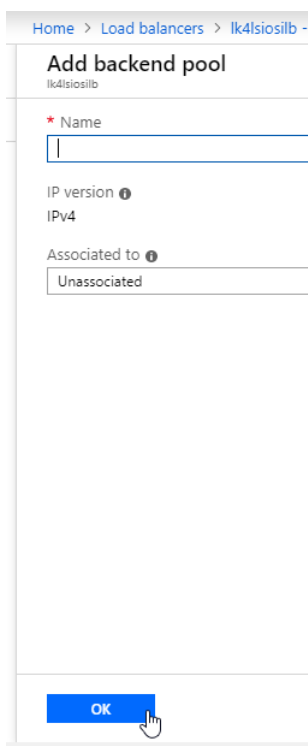
※注 2 : この値が、後に設定する仮想 IP アドレスの値としても使用され、保護対象サー

ビスのアクセスポイントとなります。

3. 内容を確認し、[Create]をクリックします。

4. 次に、作成済のクラスター用の仮想マシン 2 台を内部ロードバランサーのバックエンドプールに追加します。Azure Portal > [Load balancers] > [(ロードバランサー名)] > [Backend pools] > [Add]を選択します。





以下の値を使用します。

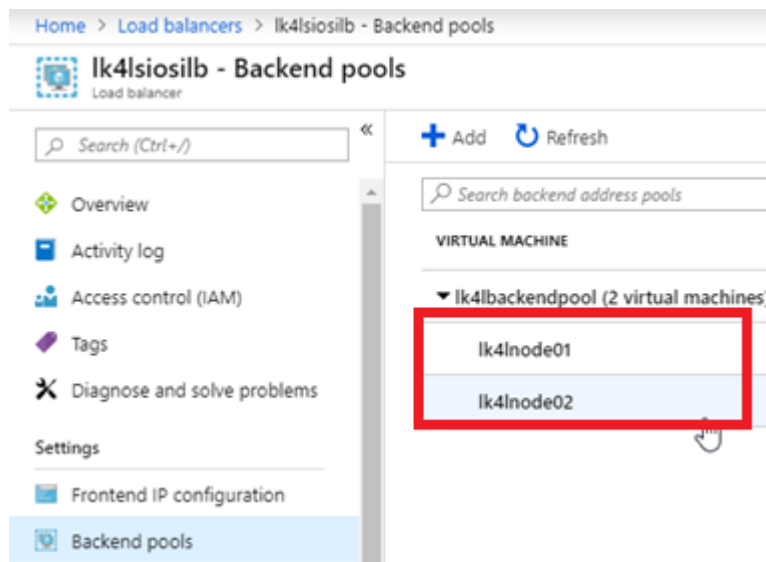
項目	入力例	
Name	lk4lbackendpool	
Associated to	Availability set	
Availability set	lk4lcluster	※注 1
Target # 1		
Target virtual machine	lk4lnode01	
Network IP configuration	ipconfig1(10.3.1.11)	※注 2
Target # 2		
Target virtual machine	lk4lnode02	
Network IP configuration	ipconfig1(10.3.1.12)	※注 2

※注 1 : 作成済の Availability Set を指定します。

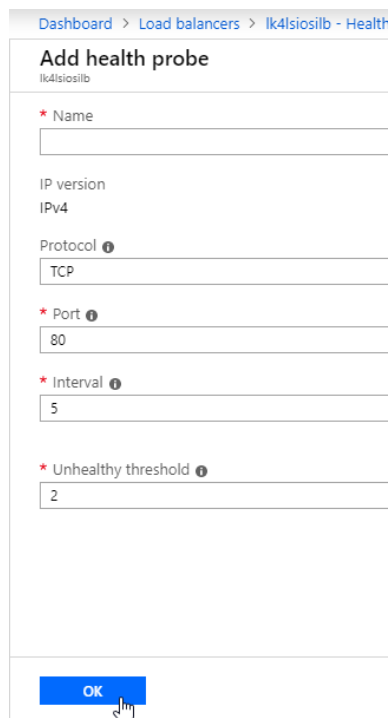
※注 2 : バックエンドプールを、作成済の Availability set 内の仮想マシンがもつプライマリーインターフェースに紐づけます。

必要な値を入力したら、[OK]をクリックします。

5. バックエンドプールが作成されたことを確認します。



6. 続けて、プローブを設定します。Azure Portal > [Load balancers] > [(ロードバランサー名)] > [Health Probes] > [Add] をクリックします。



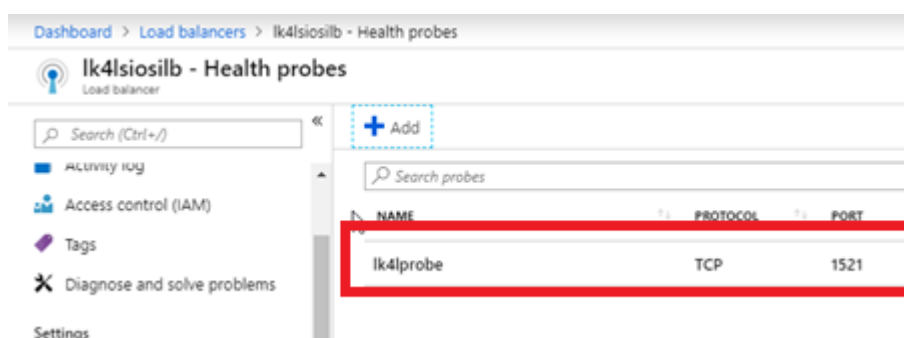
以下の値を使用します。

項目	入力例	
Name	lk4lprobeport	
Protocol	TCP	※注 1
Port	1521	※注 1

※注 1 : プロブ用には、Oracle リスナーが使用するプロトコルおよびポートを指定します。

7. 必要な値を入力したら、[OK]をクリックします。

8. プロブが作成されたことを確認します。



9. 続けて、負荷分散規則を設定します。Azure Portal > [Load balancers] > [(ロードバランサー名)] > [Load balancing rules] > [Add] をクリックします。

Dashboard > Load balancers > lk4lsiosilb - Load b

### Add load balancing rule

lk4lsiosilb

Protocol  
 TCP  UDP

\* Port

\* Backend port ⓘ

Backend pool ⓘ

Health probe ⓘ

Session persistence ⓘ

Idle timeout (minutes) ⓘ

Floating IP (direct server return) ⓘ  
 Disabled  Enabled

以下の値を使用します。

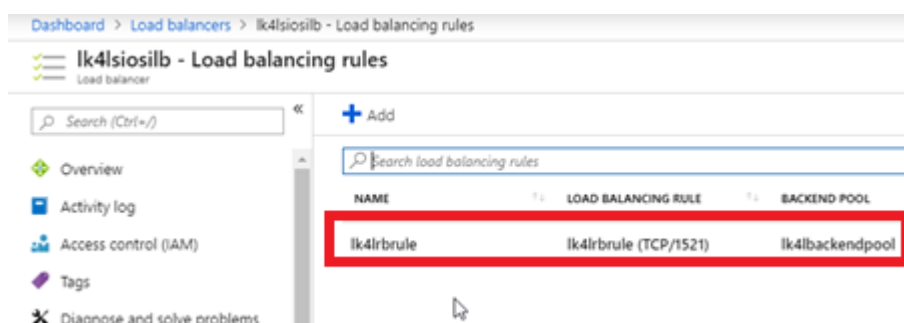
項目	入力例	
Name	lk4lrbrule	
Protocol	TCP	
Port	1521	※注 1
Backend port	1521	※注 1
Backend pool	lk4lbackendpool	
Health probe	lk4lprobe	
Floating IP address	Enabled	※注 2

※注 1 : 負荷分散用のポートには、Oracle リスナーが使用するポートを指定します。

※注 2 : クライアントからクラスターへ向けた通信は、IP リソース(VIP) 宛のパケット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。

10. 必要な値を入力したら、[OK]をクリックします。

11. 負荷分散規則が作成されていることを確認します。



以上でロードバランサーの設定は完了です。

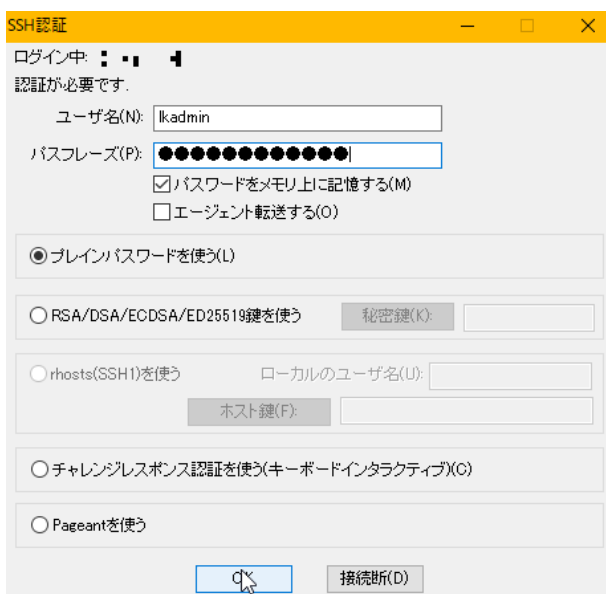
## OS の設定

作成した 3 台の仮想マシンに対して OS 設定を行い、LifeKeeper がインストールできるようにします。

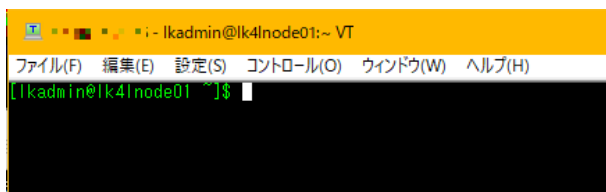
### 仮想マシンへのログイン

両クラスターノードおよびクライアントで、以下の設定を行います。

1. Azure Portal > [Virtual machines] > [Overview]>[Connect]を選択し、接続に必要な情報を確認します。
2. ローカル PC でターミナルソフトウェア(TeraTerm 等)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 アクセスします。



3. 仮想マシンにログインできました。



4. 次に、以降 root が使用できるよう設定します。root 権限に変更します。

5. \$ sudo su -

6. root パスワードを設定します。

7. # passwd

8. root ログインを許可します。

```
# cp /etc/ssh/sshd_config /etc/ssh/sshd_config.org
# vi /etc/ssh/sshd_config
<略>
PermitRootLogin yes //コメントアウトを外す
<略>
#systemctl restart sshd
```

ターミナルソフトウェアで別ウィンドウを起動し、root でログインできることを確認します。(今まで使用していたターミナルウィンドウは、念のため閉じずに実施してください。)

ログインに成功したら、これ以降の作業は、すべて root で実施します。

## NIC アドレスの固定化

Azure インフラストラクチャー更新に伴って 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。両クラスターノードで、以下の設定を行います。

1. 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。

```
#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::20d:3aff:fe50:42bf prefixlen 64 scopeid 0x20<link>
    ether 00:0d:3a:50:42:bf txqueuelen 1000 (Ethernet)
<略>
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::e3c9:ec16:62a:bbd0 prefixlen 64 scopeid 0x20<link>
    ether 00:0d:3a:51:fc:01 txqueuelen 1000 (Ethernet)
<略>
```

2. 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。

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

3. 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。

```
#vi /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
<略>
HWADDR=00:0d:3a:50:42:bf //追記

#vi /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth1 //変更
ONBOOT=yes
<略>
HWADDR=00:0d:3a:51:fc:01 //追記
```

## SELinux

SELinux を無効化します。両クラスターノードで、以下の設定を行います。

1. 以下のファイルを開きます。

```
# vi /etc/selinux/config
```

2. 以下のパラメータを、“enforcing”から“disabled”に変更して保存します。

```
SELINUX=disabled
```

3. 設定を有効にするため、仮想マシンを再起動します。
4. SELinux が無効化されていることを確認します。

```
# getenforce
Disabled
```

## ファイアーウォール

LifeKeeper は特定のポートを使用するため、下記の資料を参考に、OS のファイアーウォールの設定を変更します。

## ファイアウォールを使用した状態での LifeKeeper の実行

<http://docs.us.sios.com/spslinux/9.5.0/ja/topic/running-lifekeeper-with-a-firewall>

今回はファイアウォールを無効化します。両クラスターノードおよびクライアントで、以下の設定を行います。

1. 以下のコマンドを実行します。

```
# systemctl stop firewalld.service
# systemctl disable firewalld.service
```

## アドレス変換

ここでは、以下2つの設定を同時に実施します。

- VIP への通信が行えるよう、ILB が負荷分散した 1521 ポート宛の packets について、ディスティネーションアドレスを[IP リソースが保護する VIP]宛に変換します。
- IP リソースの監視処理が正常に行えるよう、ICMP パケットのソースアドレスを[Azure のプライベート IP アドレス]に変換します。

両クラスターノードで、以下の設定を行います。

1. IP tables パッケージをインストールします。

```
# yum -y install iptables-services
```

2. IP tables を有効にします。

```
#systemctl start iptables.service
#systemctl enable iptables.service
```

3. IP tables が有効であることを確認します。

```
#systemctl list-unit-files | grep iptables
iptables.service                                enabled
```

4. IP tables の設定を行います。

稼働系ノード (lk4lnode01) では以下のコマンドを実行します。

```
# iptables -t nat -A PREROUTING -p tcp --dport 1521 -j DNAT --to-destination 10.3.1.200:1521
# iptables -t nat -A POSTROUTING -p icmp -s 10.3.1.200 -j SNAT --to-source 10.3.1.11
# service iptables save
```

待機系ノード (lk4lnode02) では以下のコマンドを実行します。

```
# iptables -t nat -A PREROUTING -p tcp --dport 1521 -j DNAT --to-destination 10.3.1.200:1521
# iptables -t nat -A POSTROUTING -p icmp -s 10.3.1.200 -j SNAT --to-source 10.3.1.12
# service iptables save
```

5. iptables の設定が追加されていることを確認します。以下は稼働系の例です。

```
#cat /etc/sysconfig/iptables
<略>
:POSTROUTING ACCEPT [16:960]
-A PREROUTING -p tcp -m tcp --dport 1521 -j DNAT --to-destination 10.3.1.200:1521
-A POSTROUTING -s 10.3.1.200/32 -p icmp -j SNAT --to-source 10.3.1.11
<略>
```

6. 設定を有効にするため、仮想マシンを再起動します。

## GUI 接続設定

---

Azure 提供の仮想マシンは、デフォルトでは GUI 環境がインストールされていません。LifeKeeper GUI を操作できるよう、クラスターノード上に X11 環境、クライアント上に GUI 環境を設定します。

両クラスターノードで、以下の設定を行います。

1. GUI パッケージを導入します。以下のコマンドを実行します。

```
# yum -y groupinstall "X11"
```

2. パッケージの導入が完了したら、以下のコマンドを実行します。

```
# systemctl set-default graphical.target
```

3. 次に、サブネット内通信の許可設定をします。以下のファイルを編集し、エントリーを[REJECT]の前に追加します。

```
# vi /etc/sysconfig/iptables
<略>
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [4710:957955]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -s 10.3.1.0/24 -p tcp -j ACCEPT //追記
-A INPUT -s 10.3.2.0/24 -p tcp -j ACCEPT //追記
-A INPUT -j REJECT --reject-with icmp-host-prohibited
<略>
```

4. 設定を有効にするため、仮想マシンを再起動します。

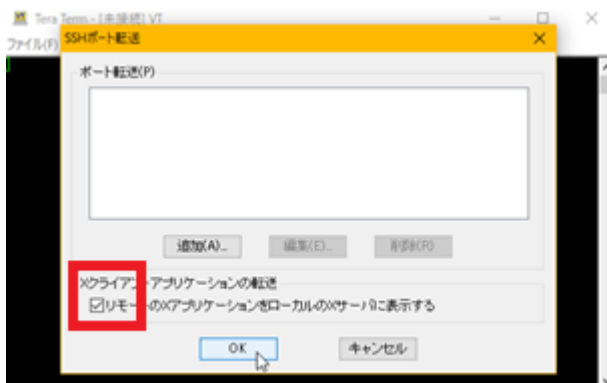
```
# reboot
```

5. 続けて、ローカル PC 側の設定を行います。ローカル PC に X Server ソフトウェア (Xming 等) をインストールし、起動します。

6. 次に、ローカル PC のターミナルソフトウェア(TeraTerm 等)に、X11

forwarding の設定を行います。

設定方法はご利用のターミナルソフトウェアによって異なります。以下は、Teraterm での設定例です。



## 名前解決

両クラスターノードおよびクライアントで、以下の設定を行います。

1. /etc/hosts に、クラスターノード（稼働系）、クラスターノード（待機系）クライアント兼 Witness サーバー 3 台の情報を登録します。

```
#vi /etc/hosts
10.3.1.11 lk4lnode01 //追記
10.3.1.12 lk4lnode02 //追記
10.3.1.50 lk4lclient //追記
```

2. それぞれが、相互にホスト名で通信できることを確認します。

```
[root@lk4lnode01 ~]# ping lk4lnode02 -c 4
[root@lk4lnode01 ~]# ping lk4lclient -c 4
```

```
[root@lk4lnode02 ~]# ping lk4lnode01 -c 4
[root@lk4lnode02 ~]# ping lk4lclient -c 4
```

```
[root@lk4lclient ~]# ping lk4lnode01 -c 4
```

```
[root@lk4lclient ~]# ping lk4lnode02 -c 4
```

## カーネルバージョン確認する

---

確認の為、RHEL のカーネルバージョンを調べます。

```
# uname -r
3.10.0-862.11.6.el7.x86_64
# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 7.5 (Maipo)
```

LifeKeeper をサポートする各ディストリビューションについては、以下のリンク先文書を参照してください。

SIOS Protection Suite for Linux サポートマトリックス

<http://docs.us.sios.com/spslinux/9.5.0/ja/topic/sios-protection-for-linux-support-matrix>

以上で LifeKeeper 導入のための事前準備は完了です。

## 6 LifeKeeper による HA クラスターの構築

仮想マシンの OS に LifeKeeper をインストールし、HA クラスターを構築します。

### LifeKeeper のインストールと起動

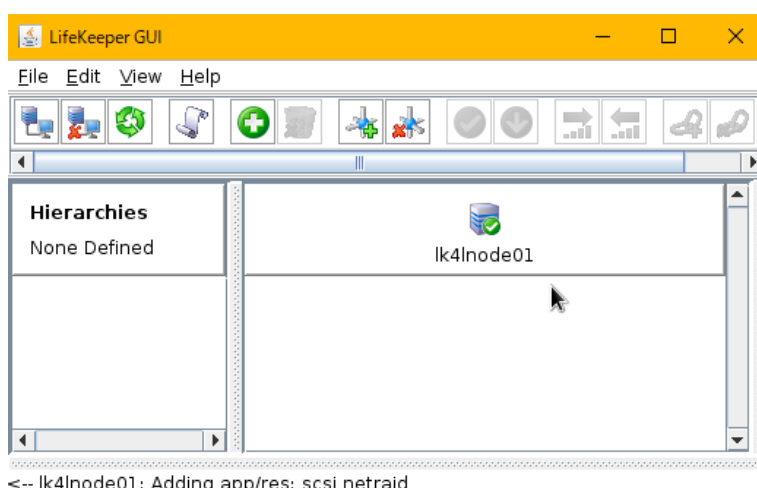
1. 事前に、scp コマンド等を使用して仮想マシン上に LifeKeeper インストールイメージおよびライセンスファイルをアップロードしておきます。
2. 両クラスターノードに LifeKeeper をインストールします。オンラインマニュアル等をご参照ください。
  - 「Use Quorum / Witness Functions」を選択します。
  - 「LifeKeeper Authentication」では、root を入力します。
  - 「Recovery Kit Selection Menu」では、「LifeKeeper Oracle RDBMS ecovery Kit」と「DataKeeper for Linux」を選択します。
  - 「LifeKeeper Startup After Install」を選択します。
5. インストール完了後、シェルの環境変数に LifeKeeper コマンドパスを追加します。

```
# vi /root/.bash_profile
PATH=$PATH:/opt/LifeKeeper/bin
MANPATH=$MANPATH:/opt/LifeKeeper/man
export PATH MANPATH
# source /root/.bash_profile
```

3. LifeKeeper GUI を起動します。

```
# lkGUIapp &
```

LifeKeeper GUI が起動しました。



## コミュニケーションパスの作成

LifeKeeper は設定した全てのコミュニケーションパスを通じ、ハートビートの応答確認を行うことによってノードの死活監視を実施しています。ハートビート通信において所定の回数応答が得られなかった場合、そのコミュニケーションパスの接続が切れていると判断しステータスを DEAD とします。全てのコミュニケーションパスのステータスが DEAD となった場合、ノード障害と判定してフェイルオーバーを開始します。

LifeKeeper ではサーバー間に 2 経路以上のコミュニケーションパスを推奨しているため、今回は以下のように構成します。

種別	プライオリティ	稼働系	待機系
TCP	1	10.3.1.11	10.3.1.12
TCP	2	10.3.2.11	10.3.2.12

コミュニケーションパスの作成方法は、以下のリンク先文書をご参照ください。

コミュニケーションパスの作成

<http://docs.us.sios.com/spslinux/9.5.0/ja/topic/creating-a-communication-path>

## Quorum/Witness の設定

スプリットブレインの発生を抑制するため、クライアント上に Quorum/Witness を設定します。今回は Quorum モードとして TCP Remote モード、Witness モードは none を使用します。

1. Quorum モード、Witness モードを設定します。両クラスターノードで、以下のファイルを編集します。

```
#vi /etc/default/LifeKeeper
<略>
QUORUM_MODE=tcp_remote //編集
<略>
QUORUM_HOSTS=10.3.1.50 //編集
<略>
QUORUM_TIMEOUT_SECS=20 //必要に応じてタイムアウトを編集
<略>
WITNESS_MODE=none //編集
<略>
```

2. LifeKeeper を再起動します。

```
#lkstop
#lkstart
```

Quorum/Witness

<http://docs.us.sios.com/spslinux/9.5.0/ja/topic/quorum-witness>

Quorum パラメーター一覧

[Linux]Quorum/Witness Server Kit のご利用について

<https://lkdkuserportal.sios.jp/hc/ja/articles/360037363892>

## Broadcast Ping の無効化

Azure の仮想マシン環境では、IP リソース作成直後 Broadcast Ping への応答が得られません。その為、Broadcast Ping による死活監視を両クラスターノードで無効にします。

以下のファイルを編集します。

```
# vi /etc/default/LifeKeeper
<略>
NOBCASTPING=1 //初期値 0 から変更
```

## IP リソースの作成

LifeKeeper GUI で IP リソースを作成します。

1. LifeKeeper GUI 管理画面より Create Resource Hierarchy を選択します。



2. Create Resource ウィザードに沿って設定します。今回は以下の値を使用します。

項目	入力もしくは選択する値	
Select Recovery Kit	IP	
Switchback Type	Intelligent	
IP Resource	10.3.1.200	※注 1
Netmask	255.255.255.0	
Network Interface	eth0	
IP Resource Tag	ip-10.3.1.200	

※注 1 : 設定済の Azure ロードバランサーのフロントエンド IP と同じ値を使用します。

Extend 時には、以下の値を使用します。

項目	入力もしくは選択する値	
Target Server	lk4lnode02	
Switchback Type	Intelligent	
Template Priority	1	
Target Priority	10	
IP Resource	10.3.1.200	※注 1
Netmask	255.255.255.0	
Network Interface	eth0	
IP Resource Tag	ip-10.3.1.200	

※注 1 : 設定済の Azure ロードバランサーのフロントエンド IP と同じ値を使用します。

3. IP リソースが作成されました。

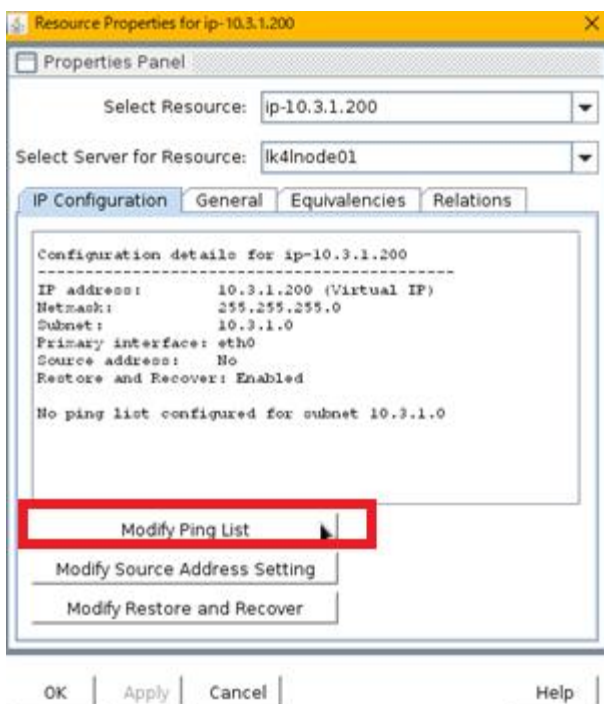
Hierarchies	lk4lnode01	lk4lnode02
Active Protected		
ip-10.3.1.200	Active 1	StandBy 10

IP リソース階層の作成

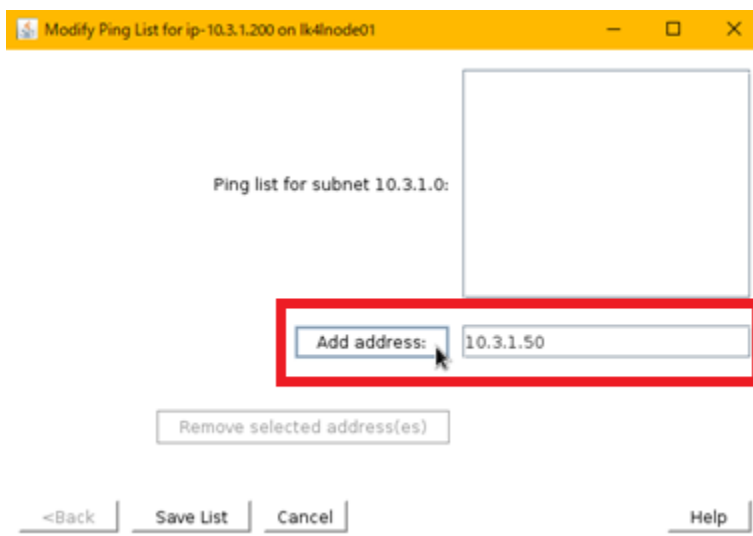
<http://docs.us.sios.com/spslinux/9.5.0/ja/topic/creating-an-ip-resource-hierarchy>

4. 今回の環境では、Broadcast Ping による監視を無効にしているため、Unicast Ping を利用して死活監視を行います。以下に設定例を記載します。

5. IP リソースのプロパティ画面を表示し、Modify Ping List をクリックします。



6. Ping List に設定する IP アドレスを入力し、Add address をクリックします。



7. Save List をクリックし、設定値を保存します。

8. 設定した Ping List のアドレスから ping 応答が得られることを確認します。

```
# ping -I 10.3.1.200 10.3.1.50
```

## Data Replication リソース階層の作成

ここでは Data Replication リソースの作成方法を解説します。

### ディスクの設定

まず、Data Replication リソースで使用するディスクの設定をします。

1. クラスターの両ノードで、oracle データベースのマウントポイント用ディレクトリを作成します。

```
# mkdir /mnt/ORA
```

2. 両ノードで、ディスク(インスタンス作成時に指定したディスク)がアタッチされていることを確認します。

```
# parted -l
(略)
/dev/sdc: unrecognised disk label
Model: Msft Virtual Disk (scsi)
Disk /dev/sdc: 33.3GB
(略)
Model: Msft Virtual Disk (scsi)
Disk /dev/sdd: 11.8GB
(略)
```

3. 両ノードで、Oracle 用ディスクのパーティションを作成します。

```
# parted /dev/sdc
GNU Parted 3.1
Using /dev/sdc
..... (略)
```

4. 作成したパーティションを確認します。

```
# parted -l
(略)
Model: Msft Virtual Disk (scsi)
Disk /dev/sdc: 32.2GB
```

```

Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
  1      1049kB 32.2GB 32.2GB                ora

Model: Msft Virtual Disk (scsi)
Disk /dev/sdd: 10.7GB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Disk Flags:

Number  Start   End     Size    File system  Flags
  1      0.00B 10.7GB 10.7GB  linux-swap(v1)
    
```

5. 両クラスターノードで、Oracle 用ファイルシステムのマウントポイントを作成します。

```
6. # mkdir /mnt/ORA
```

7. 次に、DataKeeper がデバイスを認識できるように設定します。  
 Azure 上で DataKeeper for Linux を使用してデータレプリケーションを行う場合、標準の設定ではストレージの識別に使用する ID を取得できないため、GUID Partition (GPT) を作成し一意の ID をパーティションに割り当てるか、LVM を使用する必要があります。詳細については、オンラインマニュアルの[トラブルシューティング](#)を参照してください。

ディスク設定は以上です。

## Data Replication リソース階層の作成

次に、LifeKeeper GUI で Data Replication リソースを作成します。

1. プライマリーノードで以下のコマンドで LifeKeeper GUI を起動します。

```
# /opt/LifeKeeper/bin/lkGUIapp &
```

2. [Create Resource Hierarchy]のアイコンをクリックしてリソース作成を開始します。



3. Create Resource ウィザードが表示されます。プルダウンメニューで Recovery Kit に対し[Data Replication]を選択して、[Next]をクリックします。



4. Create Resource ウィザードで、以下の値を入力します。

項目	入力もしくは選択する値	
Switchback Type	Intelligent	
Server	lk4lnode01	
Hierarchy Type	Replicate New Filesystem	
Source Disk	/dev/sdc1 ※注 1	
New Mount Point	/mnt/ORA	※注 2
New Filesystem Type	xfs	
Data Replication Resource Tag	datarep-ORA	

File System Resource Tag	/mnt/ORA	
Bitmap File	/opt/LifeKeeper/bitmap_mnt_ORA	※注 3
Enable Asynchronous Replication?	no	

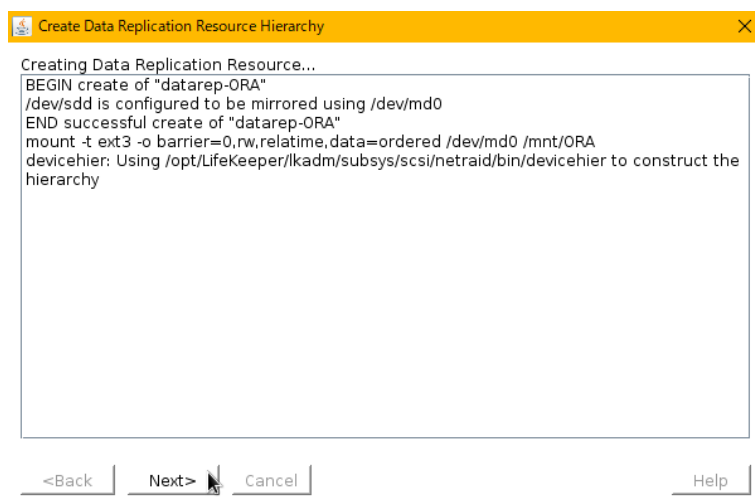
※注 1 : この後「ATTENTION! <デバイス名> is not shareable with any other server」の表示が出力されますが、レプリケーションを作成する場合は必ず出力されるため、無視して問題ありません。

※注 2 : 作成済のマウントポイントを指定します。

※注 3 : 今回はデフォルトのパスを設定していますが、パフォーマンス向上のため、より高速な専用領域を指定することも検討してください。

5. Data Replication リソースの作成が開始されます。

6. 「End of successful Create of...」と表示されれば成功です。[Next]をクリックして Pre-Extend ウィザードへ遷移します。

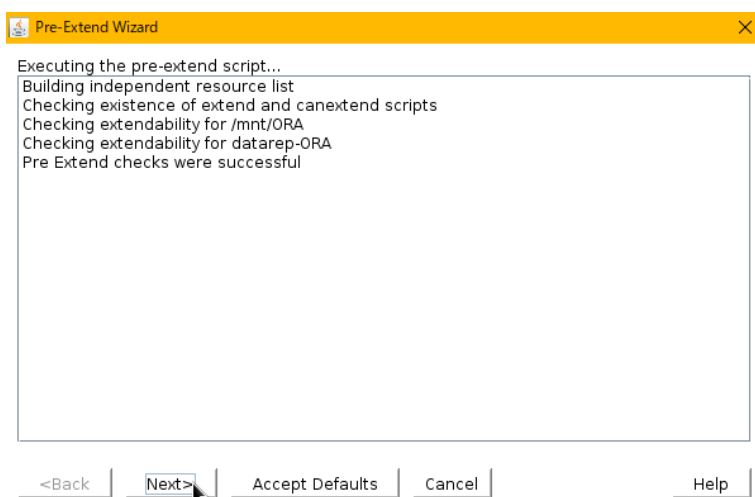


7. 以下の値を入力します。

項目	入力もしくは選択する値
Target Server	lk4lnode02
Switchback Type	Intelligent

Template Priority	1
Target Priority	10

8. Pre-Extend が開始されます。「Pre Extend checks were successful」と表示されれば成功です。[Next]をクリックして次に進みます。



9. Extend Resource ウィザードで、以下の値を入力します。

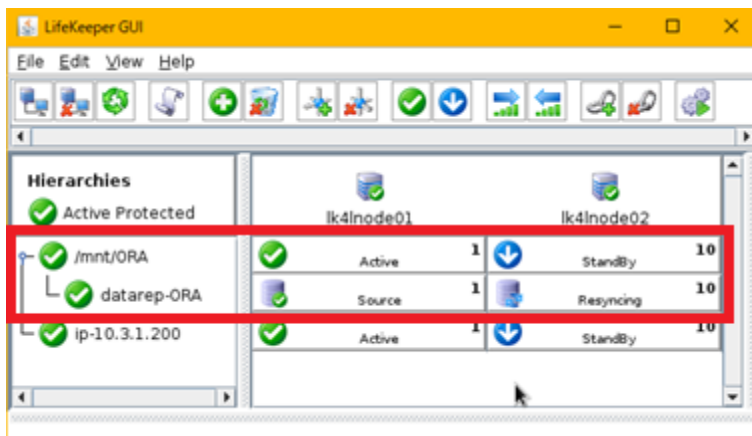
項目	入力もしくは選択する値
Mount Point	/mnt/ORA
Root Tag	/mnt/ORA
Target Disk	/dev/sdc1
Data Replication Resource Tag	datarep-ORA
Bitmap File	/opt/LifeKeeper/bitmap_mnt_ORA
Replication Path	10.3.2.11/10.3.2.12 ※注 1

※注 1 : 2 つめのサブネット上のコミュニケーションパスを指定します。

10. Extend が開始されます。「Hierarchy successful extended」と表示されれば成功です。[Finish]をクリックします。

11. [Done]を押下し、ウィザードを終了します。

Data Replication リソースが作成されました。



※Swap 領域に対しては Data Replication リソースを作成しません。

参考 : SIOS Protection Suite for Linux テクニカルドキュメンテーション

DataKeeper リソース階層の作成

<http://docs.us.sios.com/spslinux/9.5.0/ja/topic/creating-a-datakeeper-resource-hierarchy>

## Oracle リソース階層の作成

### Oracle インストール前の準備

Oracle Database 12c をインストールのための環境設定を行います。Oracle インストールの前提条件については、Oracle 社のドキュメントをご参照ください。

両クラスターノードで、以下の設定を行います

1. swap 領域を作成します。

Oracle をインストールするには 3 GB 以上の Swap 領域が必要です。

インスタンスにアサイン済の 10GB のディスク (/dev/sdd) を使用して Swap エリアを構成します。

```
# mkswap /dev/sdd
# swapon /dev/sdd
# free
      total        used        free      shared  buff/cache   available
Mem:    1706084    189248    1348300         9184     168536     1300564
Swap:   12582904         0    12582904
# blkid | grep /dev/sdd
/dev/sdd: UUID="268e56bd-c64f-4798-9197-e411a497ba11" TYPE="swap" //UUIDを控える
# vi /etc/fstab
# /etc/fstab
# Created by anaconda on Fri Mar 23 17:41:14 2018
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=50a9826b-3a50-44d0-ad12-28f2056e9927 /      xfs     defaults      0 0
UUID=268e56bd-c64f-4798-9197-e411a497ba11 swap swap defaults 0 0 ←追加
```

2. RHEL のオプションレポジトリを登録します。

必須パッケージ「compat-libstdc++-33」を含むレポジトリを追加します。

```
# yum-config-manager --enable rhui-REGION-rhel-server-extras rhui-REGION-rhel-server-optional
```

### 3. 必須パッケージをインストールします。

```
# yum -y install binutils compat-libcap1 compat-libstdc++-33 compat-  
libstdc++-33.i686 gcc gcc-c++ glibc glibc.i686 glibc-devel glibc-devel.i686  
ksh libgcc libgcc.i686 libstdc++ libstdc++.i686 libstdc++-devel libstdc++-  
devel.i686 libaio libaio.i686 libaio-devel libaio-devel.i686 libXext  
libXext.i686 libX11 libX11.i686 libxcb libxcb.i686 libXi libXi.i686 make  
sysstat
```

### 4. カーネルパラメータを編集します。

注意：記号、空白はワード、エクセルの仕様で勝手に異なる文字に変更されてコピーペーストでコマンド実行すると、エラーになる場合があります。また、末尾に空白を絶対につけないでください。変数が数値ではなく文字列に認識されてしまい、コマンドが失敗する場合があります。

```
# MEMTOTAL=$(free -b | sed -n '2p' | awk '{print $2}')  
# SHMMAX=$(expr $MEMTOTAL / 2)  
# SHMMNI=4096  
# PAGESIZE=$(getconf PAGE_SIZE)  
# cat >> /etc/sysctl.conf << EOF  
>fs.aio-max-nr = 1048576  
>fs.file-max = 6815744  
>kernel.shmmax = $SHMMAX  
>kernel.shmall = `expr ¥( $SHMMAX / $PAGESIZE ¥) ¥* ¥( $SHMMNI / 16  
¥)`  
>kernel.shmmni = $SHMMNI  
>kernel.sem = 250 32000 100 128  
>net.ipv4.ip_local_port_range = 9000 65500  
>net.core.rmem_default = 262144  
>net.core.rmem_max = 4194304  
>net.core.wmem_default = 262144  
>net.core.wmem_max = 1048576  
>EOF
```

5. カーネルに反映させます

```
# sysctl -p
```

6. Oracle 専用のユーザー/グループを作成し、システムの環境設定をします。

```
# i=54321; for group in oinstall dba backupdba oper dgdba kmdba; do
groupadd -g $i $group; i=`expr $i + 1` ;done
# useradd -u 1200 -g oinstall -G dba,oper,backupdba,dgdba,kmdba -d
/home/oracle oracle
```

7. oracle ユーザーのパスワードを設定します

```
# passwd oracle
Changing password for user oracle.
New password:
Retype new password:
Passwd: all authentication tokens updated successfully.
```

8. /mnt/ORA が DK により保護されておりプライマリーノードでマウントされていることを確認します。(この作業はプライマリーノードのみ行います)

```
# df
Filesystem      1K-blocks    Used Available Use% Mounted on
<<中略>>
/dev/md0        31864752  7396580  22842880  25% /mnt/ORA
<<後略>>

#lcdstatus -e
lk4lnode02  LSNR.LISTENER  dblistener-11079  ISP      1  lk4lnode01
lk4lnode02  /mnt/ORA        /mnt/ORA          ISP      1  lk4lnode01
lk4lnode02  datarep-ORA    /dev/sdc          ISP      1  lk4lnode01
lk4lnode02  ip-10.3.1.200  IP-10.3.1.200    ISP      1  lk4lnode01
<<後略>>
```

9. ディレクトリ作成とアクセス権、オーナーを変更します。(この作業はプライマ

リーノードのみ行います)

```
# mkdir -p /mnt/ORA/app/oracle
# chown -R oracle:oinstall /mnt/ORA/app
# chmod -R 775 /mnt/ORA
```

10. /etc/pam.d/login 14 行目あたりに追記します。

```
# vi /etc/pam.d/login
session required pam_selinux.so open
session required pam_namespace.so
session required pam_limits.so //追記
session optional pam_keyinit.so force revoke
```

11. /etc/security/limits.conf の最終行に追記します。

```
# vi /etc/security/limits.conf
oracle soft nproc 2047 //追記
oracle hard nproc 16384 //追記
oracle soft nofile 1024 //追記
oracle hard nofile 65536 //追記
oracle soft stack 10240 //追記
oracle hard stack 32768 //追記
```

12. Oracle 用の環境設定をしておきます。作業は前の手順で作成した「oracle」ユーザーで行います。

```
# su oracle
$ vi ~/.bash_profile
//以下を最終行に追記
umask 022
export ORACLE_BASE=/mnt/ORA/app/oracle
export ORACLE_HOME=/mnt/ORA/app/oracle/product/12.1.0/dbhome_1
export PATH=$PATH:$ORACLE_HOME/bin
export ORACLE_SID= <SID の名前>
export NLS_LANG=Japanese_Japan.AL32UTF8

$source ~/.bash_profile
```

ここで設定した ORACLE\_SID は、後の手順で使用するため控えておきます。今回は

"lkoracle"とします。

root の xauth の設定を oracle ユーザーにコピーします。(この作業はプライマリーノードのみ行います)。

```
# xauth list
# su oracle
$ xauth add <DisplayName> <ProtocolName> <Hexkey>
```

13. oracle ユーザーの xauth の出力結果が root と同じであることを確認します。

```
$ xauth list
```

14. インストール作業用ディレクトリ作成 (この作業はプライマリーノードのみ行います)

```
$ mkdir /home/oracle/tmp
```

以上でインストールするための準備は完了です。

## Oracle DB のインストール

ここからの作業はクラスターノード(プライマリー)で行います。Oracle のインストールについては、Oracle 社のドキュメントをご参照ください。

1. Oracle 12c のインストールパッケージをダウンロードし、/home/oracle 以下の適当なディレクトリに解凍します。

2. su oracle で oracle ユーザーになります。

3. 解凍したディレクトリ配下に database というディレクトリが作成されています。そのディレクトリは以下にあるインストーラーを起動します。

```
$/runInstaller &
```

4. Configure Security Update

Oracle に登録済の Email とサポートパスワード入力して[Next]。ここは必須ではありません。

5. Installation option

Install Database Software only を選択して[Next]

6. Grid Installation options

Single Instance Database installation を選択して[Next]

7. Product Language

好みの言語を選択して[Next]

8. Database Edition

Enterprise Edition を選択して[Next]

9. Installation Location

Oracle base = /mnt/ORA/app/oracle

Software Location = /mnt/ORA/app/oracle/product/12.1.0/dbhome\_1

であればそのまま[Next]

10. Create Inventory

デフォルトのまま[Next]

11. Product list Checks

デフォルトのまま[Next]

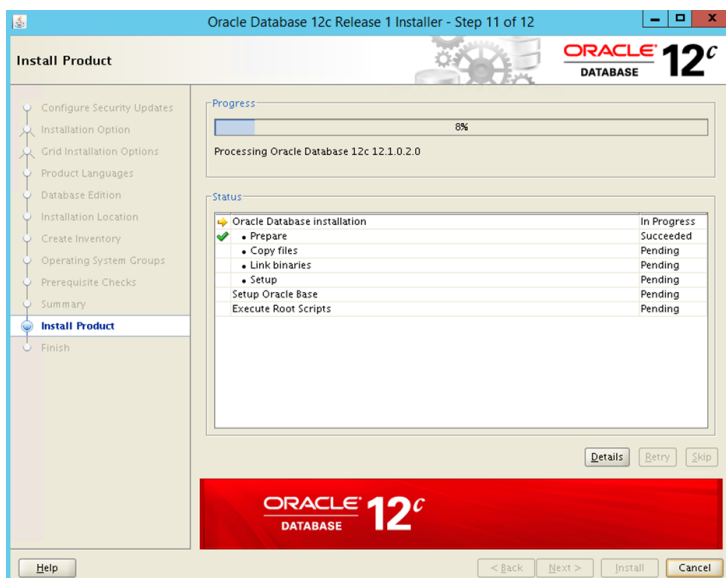
12. Operating System Group

デフォルトのまま[Next]

### 13. Summary

問題なければそのまま[Install]を実行します。

### 14. インストールが開始します。



15. Execute Configuration scripts のウィンドウがオープンします。指示に従い root でスクリプトを実行します。

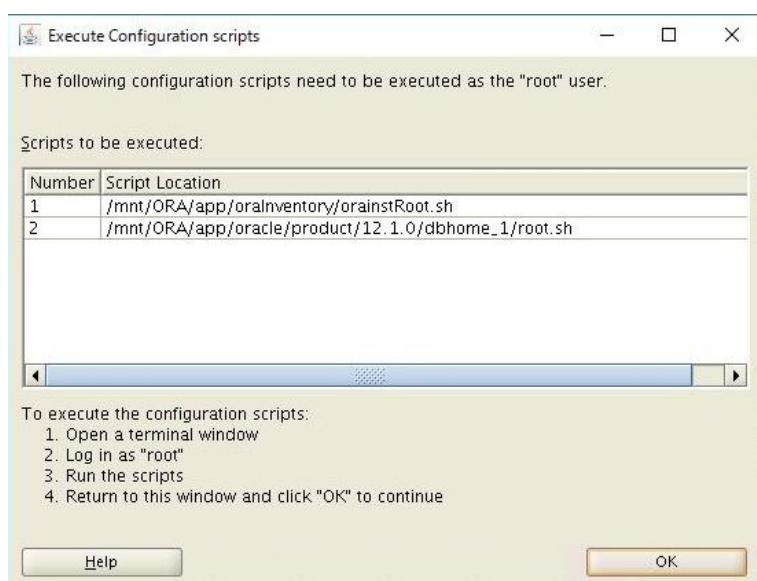
```
# /mnt/ORA/app/oraInventory/orainstRoot.sh
Changing permissions of /mnt/ORA/app/oraInventory.
Adding read,write permissions for group.
Removing read,write,execute permissions for world.

Changing groupname of /mnt/ORA/app/oraInventory to oinstall.
The execution of the script is complete.
# /mnt/ORA/app/oracle/product/12.1.0/dbhome_1/root.sh
Performing root user operation.

The following environment variables are set as:
  ORACLE_OWNER= oracle
  ORACLE_HOME= /mnt/ORA/app/oracle/product/12.1.0/dbhome_1

Enter the full pathname of the local bin directory: [/usr/local/bin]:
Copying dbhome to /usr/local/bin ...
Copying oraenv to /usr/local/bin ...
Copying coraenv to /usr/local/bin ...

Creating /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root script.
Now product-specific root actions will be performed.
```



スクリプト実行完了したら、[OK]ボタンをクリックします

16. [close]を押してインストール完了します。

## リスナー設定

リスナーの設定をします。クラスターノード(プライマリー)で、以下の設定を行います。

1. oracle ユーザーで以下のコマンドを実行します。

```
$ /mnt/ORA/app/oracle/product/12.1.0/dbhome_1/bin/netca &
```

2. Listener Configuration を選択して[Next]をクリックします。



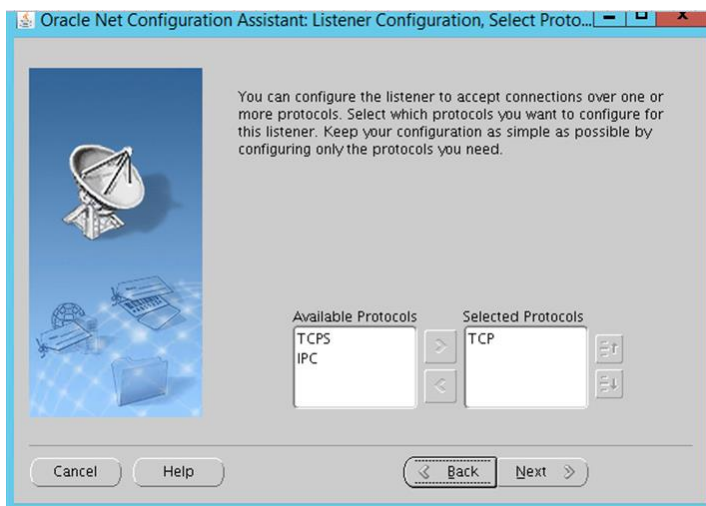
3. Add を選択して[Next]をクリックします。



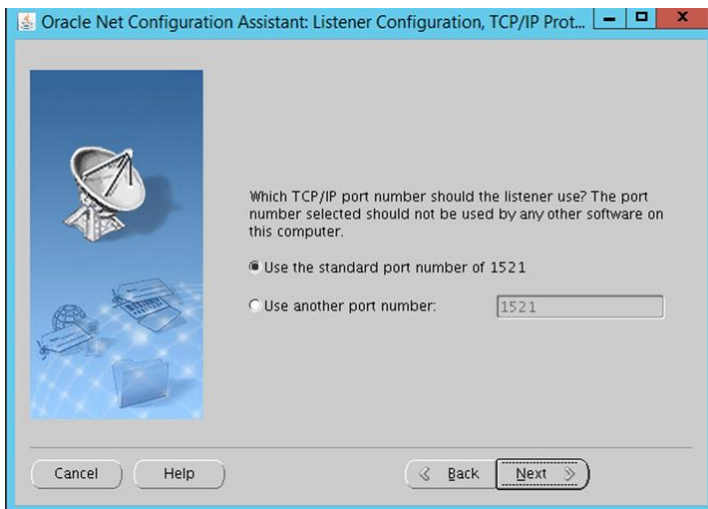
4. Listener の名前を指定して[Next]をクリックします。



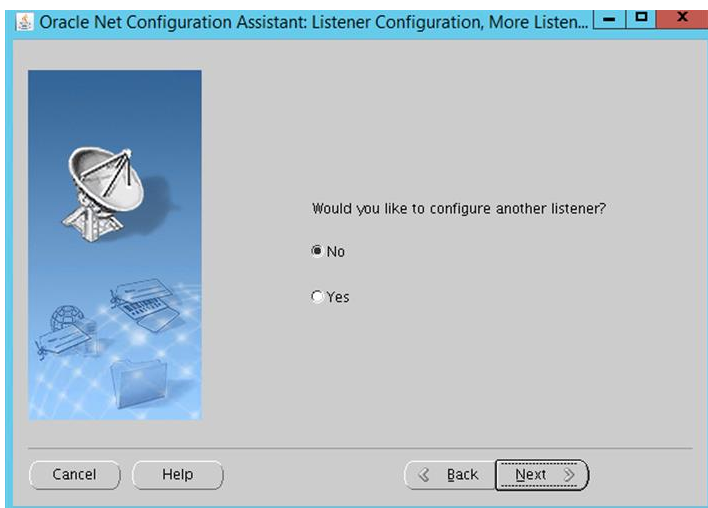
5. このまま[Next]をクリックします。



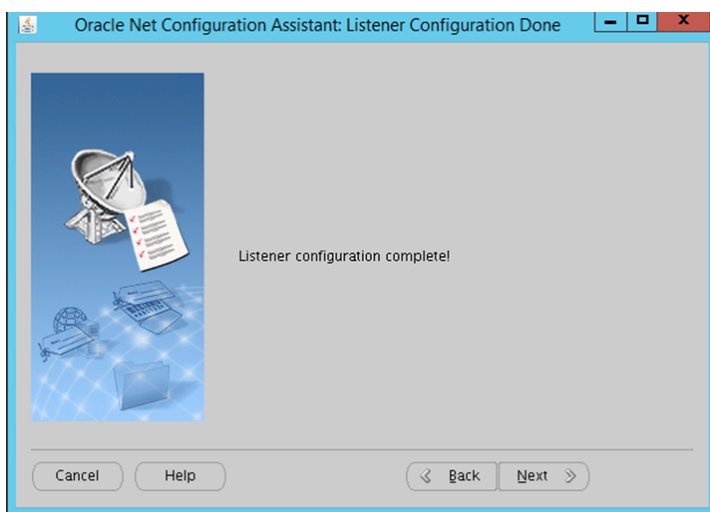
6. ポートはデフォルトの 1521 を選択して[Next]をクリックします。



7. Listener 設定終了します。[Next]をクリックします。



8. Listener 設定が完了しました。ダイアログ右上の[X]をクリックして完了します。



以上で Listener 設定は完了です。

## DB の作成

---

クラスターノード(プライマリー)で、以下の設定を行います。

1. oracle ユーザーで以下のコマンドを実行して下さい

```
$ /mnt/ORA/app/oracle/product/12.1.0/dbhome_1/bin/dbca &
```

2. Database operation

Create Database を選択します。[Next]をクリックします。

3. Creation Mode

Advanced Mode を選択します。[Next]をクリックします。

4. Database Template

General Purpose or Transaction Processing を選択して[Next]をクリックします。

#### 5. Database Identification

データベースの名前と SID を設定します。今回は以下の通り設定します。

Global Database Name = lkoracle

SID = lkoracle

#### 7. Management Options

デフォルトのままにします。[Next]をクリックします。

#### 8. Database Credentials

Use the Same Administrative Password for All Account を選択して Password を登録します。

#### 9. Network Configuration

すでに Listener を作成しているので、デフォルトのままにします。[Next]をクリックします。

#### 10. Storage Locations

デフォルトのままにします。[Next]をクリックします。

#### 11. Database Options

デフォルトのままにします。[Next]をクリックします。

#### 12. Initialize Parameters

デフォルトのままにします。[Next]をクリックします。

#### 13. Creation Options

デフォルトのままにします。[Next]をクリックします。

#### 14. Summary

設定に問題なければ、[Finish]をクリックしてデータベース作成をスタートします。

15. データベースの作成が完了しました。

16. 続いて、`/etc/oratab` を修正します。なお、本設定は後の手順で元に戻します。

```
# vi /etc/oratab
oracle:/mnt/ORA/app/oracle/product/12.1.0/dbhome_1:Y ←NをYに変更
```

17. `/etc/oratab` をプライマリーノードからセカンダリーノードにコピーします

これで Oracle DB のインストールは完了です。

※セカンダリーノードへの Oracle DB インストールは不要です。

## Oracle の設定

リスナーリソース作成前に、設定ファイルの修正が必要です。これは、リスナーに対して仮想 IP アドレスでアクセスするためです。

クラスターノード(プライマリー)で、以下の設定を行います。

1. `$ORACLE_HOME/network/admin/listener.ora` を修正します。

```
$vi $ORACLE_HOME/network/admin/listener.ora
<<中略>>
  SID_LIST_LISTENER =
    (SID_LIST =
      (SID_DESC =
        (SID_NAME = LKORACLE)
      )
    )
  )
  LISTENER =
    (DESCRIPTION_LIST =
      (DESCRIPTION =
        (ADDRESS = (PROTOCOL = TCP)(HOST = 10.3.1.200)(PORT = 1521))
      )
    )
  )
```

2. \$ORACLE\_HOME/network/admin/tnsnames.ora を修正します。

```
$vi $ORACLE_HOME/network/admin/tnsnames.ora
LISTENER_LKORACLE =
  (ADDRESS = (PROTOCOL = TCP)(HOST = 10.3.1.200)(PORT = 1521))

LKORACLE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 10.3.1.200)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = lkoracle)
    )
  )
```

3. リスナー経由のログインで使用する、パスワードファイルを作成します。パスワードファイルの記述形式については、Oracle 社の公式ドキュメントを参照してください。

```
$orapwd file=orapwlkoracle force=y ignorecase=y password=XXXXXXXX
```

4. リスナー経由で、データベースに接続確認を行います。

```
$ sqlplus sys/XXXXXXXX@10.3.1.200:1521/lkoracle as sysdba
SQL*Plus: Release 12.1.0.2.0 Production on 木 5月 23 07:52:41 2019

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
に接続されました。
SQL> select instance_name from v$instance ;

INSTANCE_NAME
-----
lkoracle
```

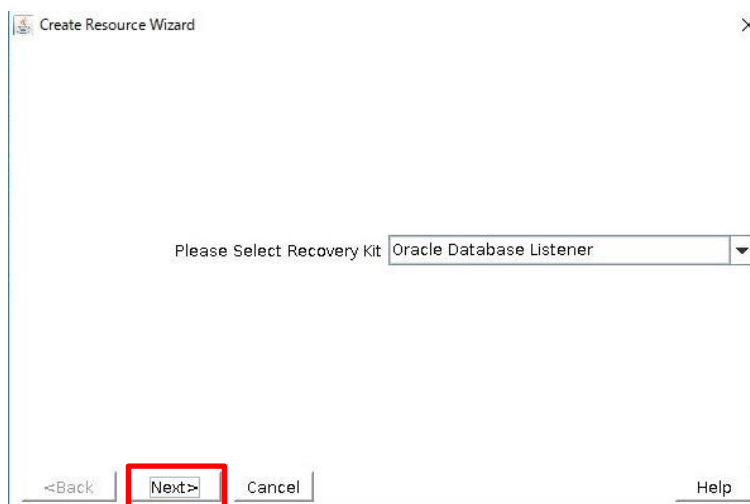
## Oracle database listener リソース階層の作成

最初に Oracle Database Listener リソースを作成します。

1. [Create Resource Hierarchy]のアイコンをクリックしてリソース作成を開始します。



2. Create Resource ウィザードが表示されます。プルダウンメニューで Recovery Kit に対し[Oracle Database Listener]を選択して、[Next]をクリックします。



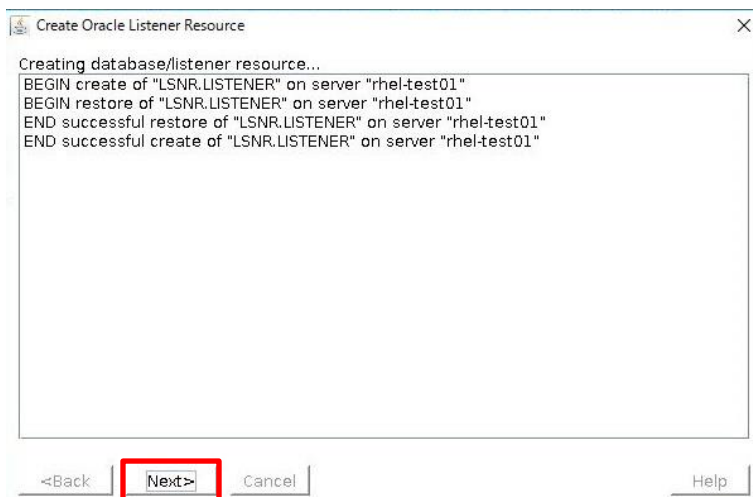
3. Create Resource ウィザードで、以下の値を入力します。

番号	項目	入力もしくは選択する値
1	Switchback Type	Intelligent
2	Server	lk4lnode01
3	Listner Configuration File Path	/mnt/ORA/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
4	Listener Name(s)	LISTENER
5	Listener Executable(s)	/mnt/ORA/app/oracle/product/12.

		1.0/dbhome_1/bin
6	Listener Protection Level	Full Control(Start,Stop,Monitor,& Recover)
7	Listener Recovery Level	Standard,(On)
8	IP Address Name	ip-10.3.1.200
9	Listener Tag	LSNR.LISTENER

4. Oracle Database Listener リソースの作成が開始されます。

5. 「End of successful Create of...」 と表示されれば成功です。[Next]をクリックして Pre-Extend ウィザードへ遷移します。



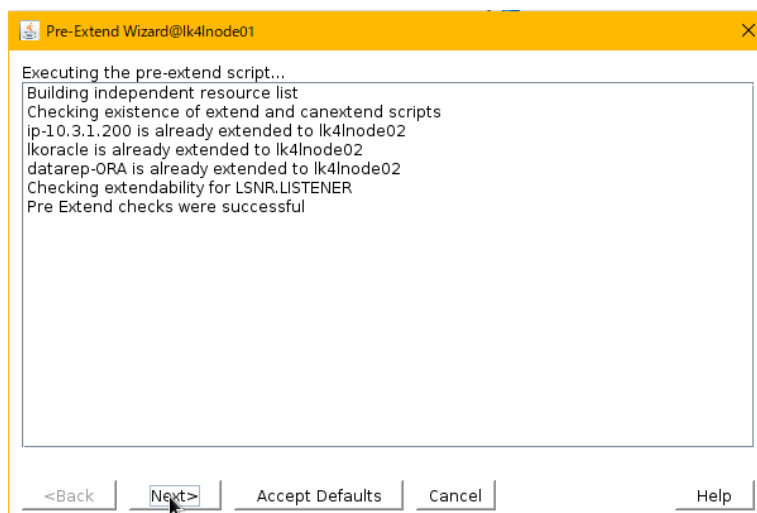
6. 以下の値を入力します。

番号	項目	入力もしくは選択する値
1	Target Server	lk4lnode02
2	Switchback Type	Intelligent
3	Template Priority	1
4	Target Priority	10

7. Pre-Extend が開始されます。「Pre Extend checks were successful」 と表示されれば成功です。[Next]をクリックして次に進みます。

8. Pre-Extend が開始されます。

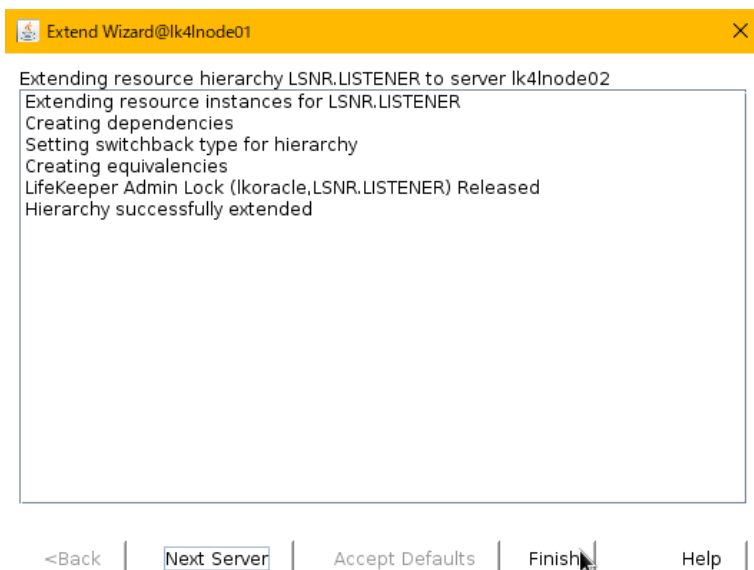
9. 「Pre Extend checks were successful」と表示されれば成功です。[Next]をクリックして進めてください。



10. Extend Oracle Listener Resource ウィザードで、以下の値を入力します。

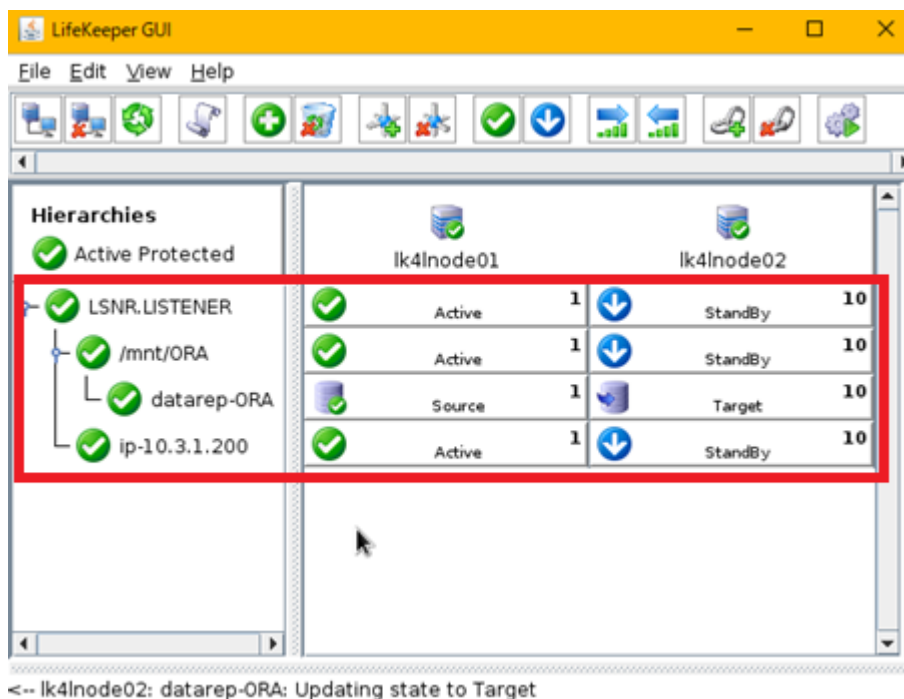
番号	項目	入力もしくは選択する値
1	Enter the Listener Configuration File Path	/mnt/ORA/app/oracle/product/12.1.0/dbhome_1/network/admin/listener.ora
2	Enter the Path to the Listener Executable(s)	/mnt/ORA/app/oracle/product/12.1.0/dbhome_1/bin
3	Listener Tag	LSNR.LISTENER

11. Extend が開始されます。「Hierarchy successful extended」と表示されれば成功です。[Finish]をクリックします。



12. [Done]を押下し、ウィザードを終了します。

13. Oracle Database Listener リソースが作成されました。



参考 : Oracle Recovery Kit 管理ガイド リソース階層の作成

<http://docs.us.sios.com/spslinux/9.5.0/ja/topic/creating-an-oracle-resource-hierarchy>

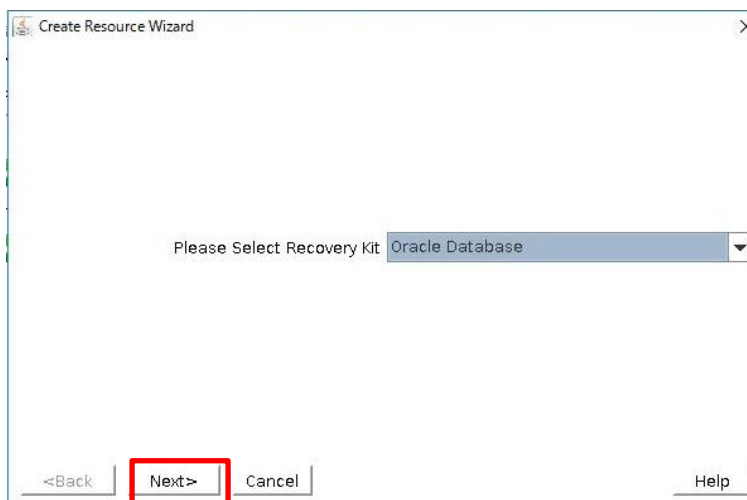
## Oracle リソース階層の作成

次に Oracle database のリソースを作成します。

1. [Create Resource Hierarchy]のアイコンをクリックしてリソース作成を開始します。



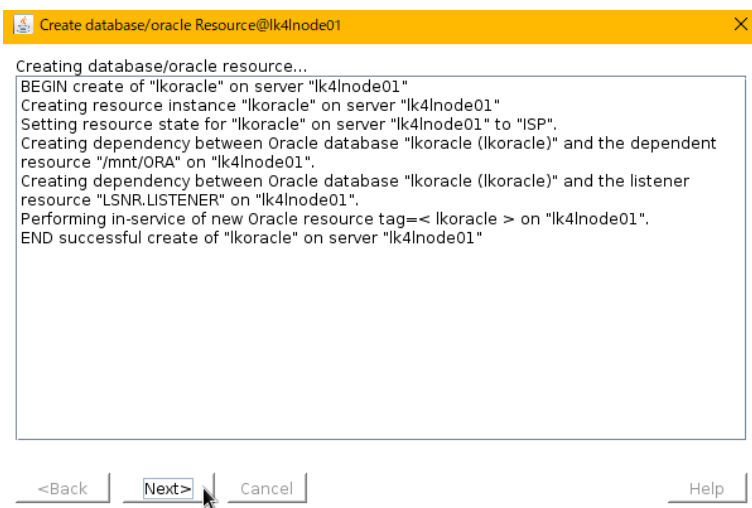
2. Create Resource ウィザードが表示されます。プルダウンメニューで Recovery Kit に対し[Oracle Database]を選択して、[Next]をクリックします。



3. Create Resource ウィザードで、以下の値を入力します。

番号	項目	入力もしくは選択する値
1	Switchback Type	Intelligent
2	Server	lk4lnode01
3	Oracle__SID for Database	lkoracle
4	User Name	oracle
5	Password	<<Oracle DB 作成時に指定したパスワード>>
6	Select the Oracle Listener	LSNR.LISTENER
7	Database Tag	lkoracle

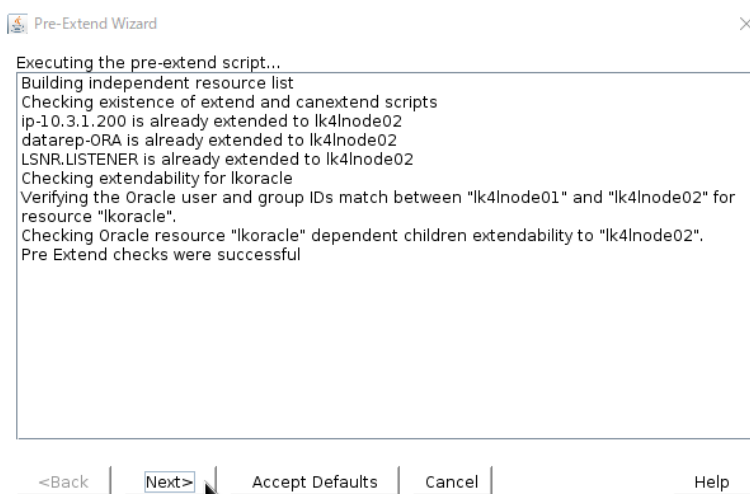
4. Oracle Database リソースの作成が開始されます。
5. 「End of successful Create of...」と表示されれば成功です。[Next]をクリックして Pre-Extend ウィザードへ遷移します。



6. 以下の値を入力します。

番号	項目	入力もしくは選択する値
1	Target Server	lk4lnode02
2	Switchback Type	Intelligent
3	Template Priority	1
4	Target Priority	10

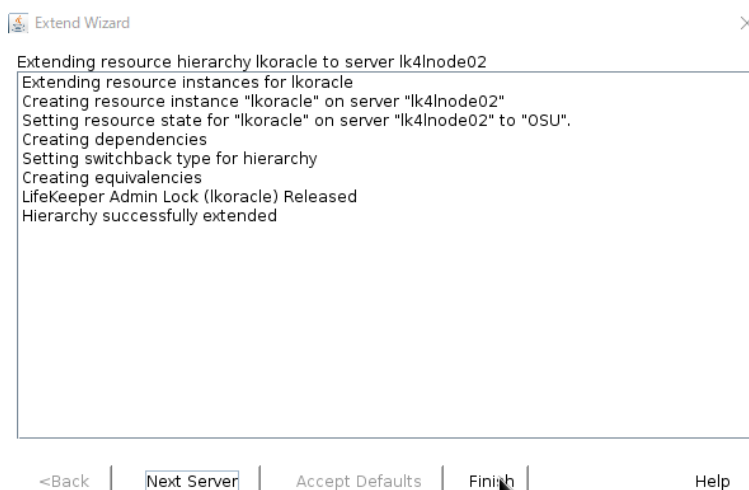
7. Pre-Extend が開始されます。「Pre Extend checks were successful」と表示されれば成功です。[Next]をクリックして次に進みます。



8. Extend database/oracle Resource ウィザードが表示されたら、[Extend]をクリックします。



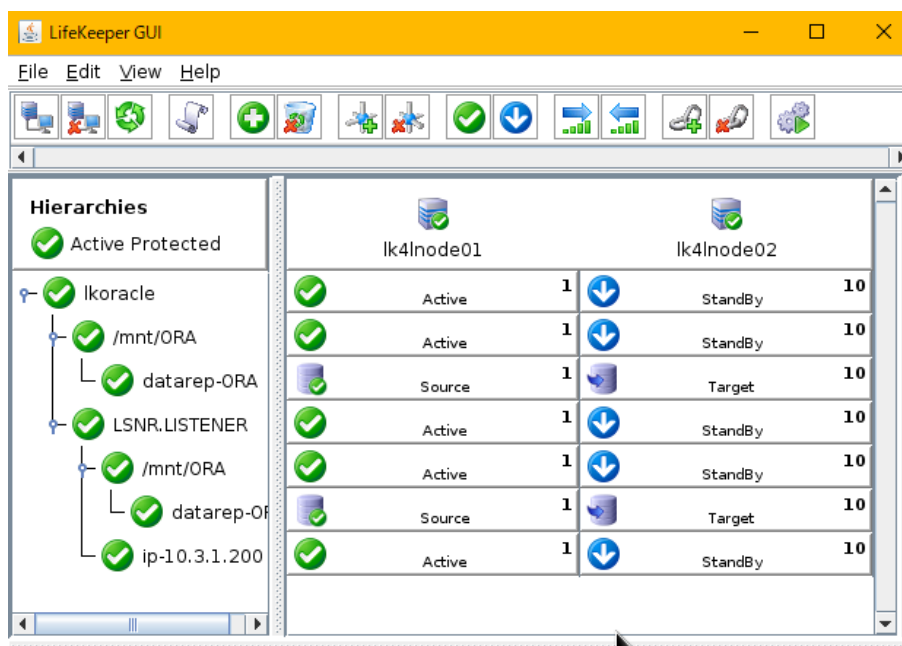
9. Extend が開始されます。「Hierarchy successful extended」と表示されれば成功です。[Finish]をクリックします。



10. [Done]を押下し、ウィザードを終了します。

11. Oracle Database リソースが作成されました。

完成したリソースツリーです。



12. /etc/oratab の修正を元に戻します。

```
# vi /etc/oratab
oracle:/mnt/ORA/app/oracle/product/12.1.0/dbhome_1:N ←YをNに変更
```

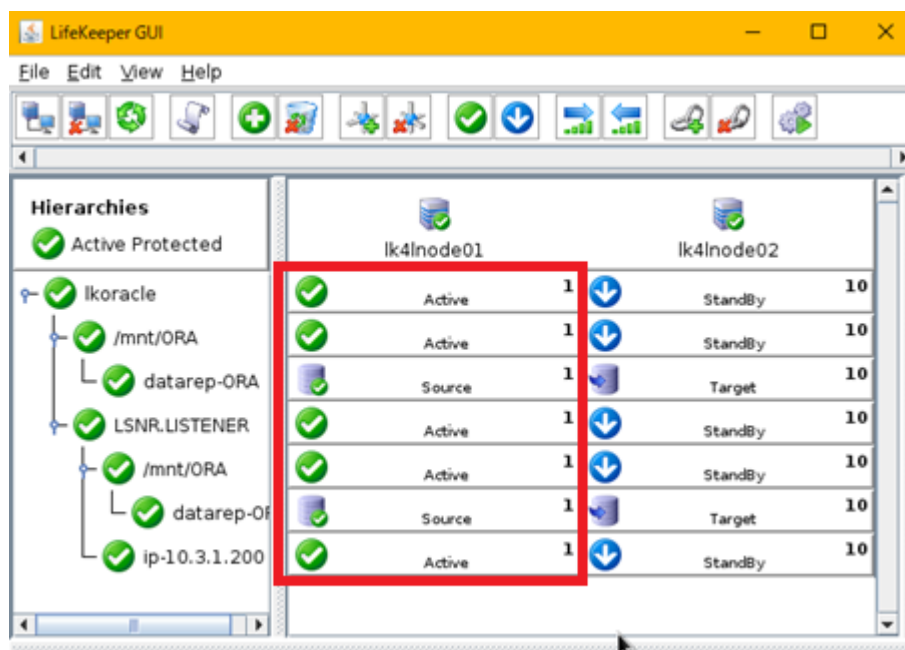
参考 : Oracle Recovery Kit 管理ガイド リソース階層の作成

<http://docs.us.sios.com/spslinux/9.5.0/ja/topic/creating-an-oracle-resource-hierarchy>

## 接続確認

最後にリモートのクライアントより、データベースへの接続確認を行います。

1. クライアント上に Oracle クライアントソフトウェアを導入します。
2. まず、プライマリーノード上でリソースがアクティブであることを確認します。



3. リモートのクライアントから、リスナー経由でデータベースに接続します。(接続には仮想 IP アドレスを使用します。)

```
# hostname
lk4lclient
# sqlplus sys/XXXXXXXX@10.3.1.200:1521/lkoracle as sysdba
```

```
SQL*Plus: Release 19.0.0.0.0 - Production on Sat May 25 17:21:32 2019
```

Version 19.3.0.0.0

Copyright (c) 1982, 2019, Oracle. All rights reserved.

Connected to:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options

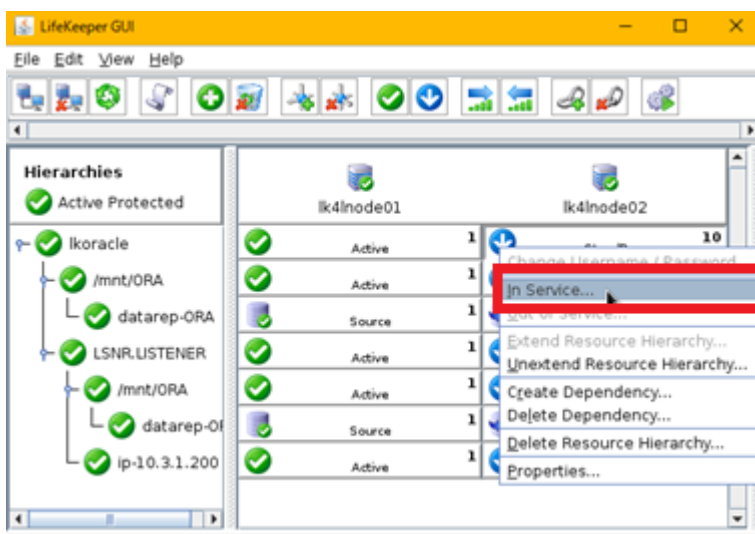
```
SQL> select instance_name from v$instance ;
```

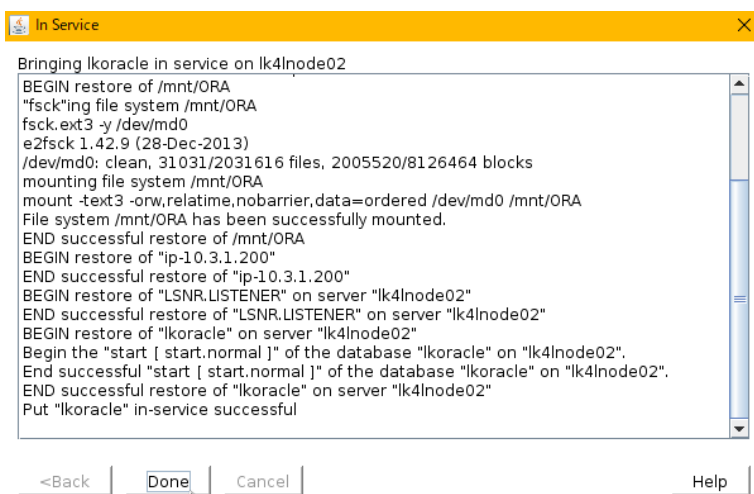
```
INSTANCE_NAME
```

```
-----
```

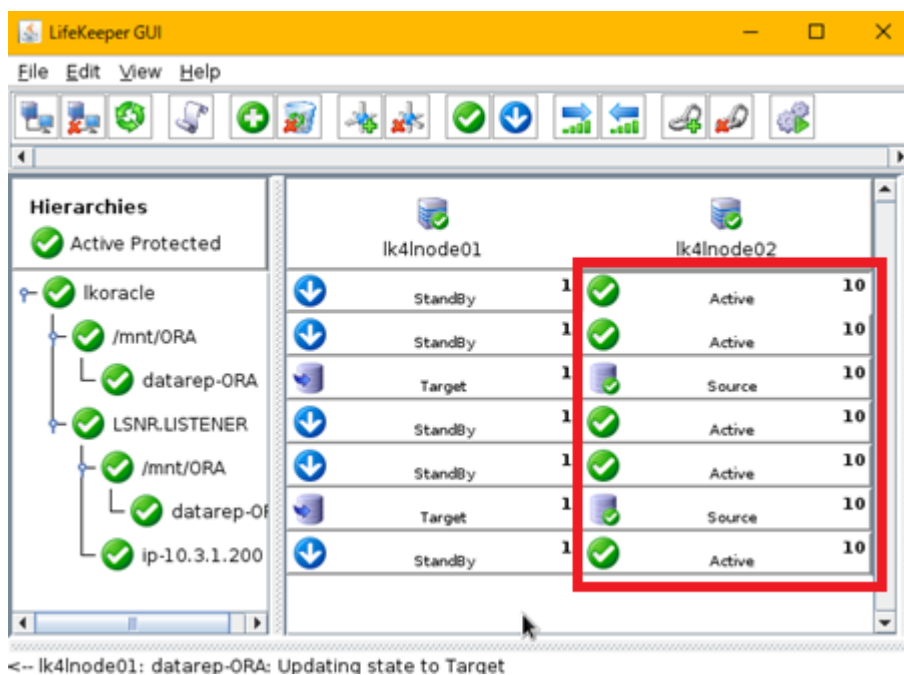
```
lkoracle
```

4. LifeKeeper GUI で Oracle Database リソースのスイッチオーバーを実施します。  
Oracle Database リソースを右クリック > [In Service]を選択します。





5. スイッチオーバーが完了しました。



6. リモートのクライアントから以下の通りリスナー経由で接続します。(クラスターホスト名でアクセスします)

```

# hostname
lk4lclient
# sqlplus sys/XXXXXXXX@10.3.1.200:1521/lkoracle as sysdba

SQL*Plus: Release 19.0.0.0.0 - Production on Sat May 25 17:21:32 2019
Version 19.3.0.0.0
    
```

```
Copyright (c) 1982, 2019, Oracle. All rights reserved.
```

```
Connected to:
```

```
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
```

```
With the Partitioning, OLAP, Advanced Analytics and Real Application Testing options
```

```
SQL> select instance_name from v$instance ;
```

```
INSTANCE_NAME
```

```
-----
```

```
lkoracle
```

7. 切り替え前後において、同じ IP アドレス（仮想 IP アドレス）で Oracle に接続  
できることが確認できました。

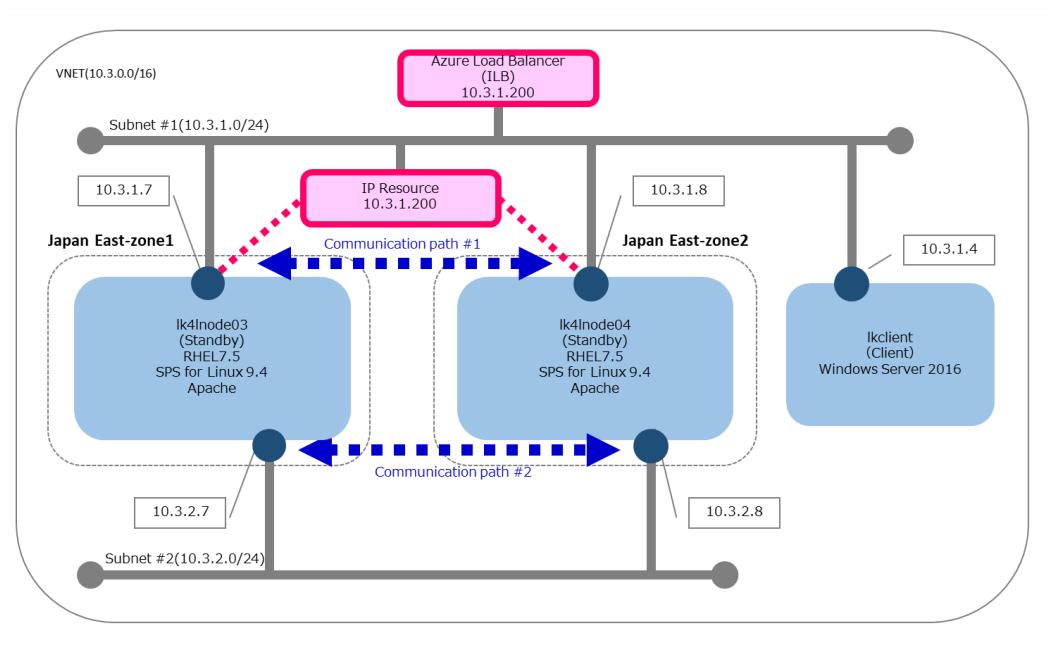
## 7 Availability Zone(高可用性ゾーン)

LifeKeeper を利用する場合、Azure 上のクラスター用仮想サーバー (VM) は、これまで説明をしてきた「Availability Set (高可用性セット) : 同一 DC 内の別サーバーによる HA クラスタ構成」のほかに「Availability Zone(高可用性ゾーン) : 同一リージョン内の異なる DC 間での HA クラスタ構成」もサポート構成となります。

Microsoft Azure Cloud 上で Availability Zone(高可用性ゾーン) 構成 (以下 Multi-AZ) を LifeKeeper for Linux でサポートするにあたり、ノード切り替えに伴い ILB でのリダイレクトが Multi-AZ 環境でも正常に動作するかどうか動作確認を行いました。

### 7.1 動作確認内容

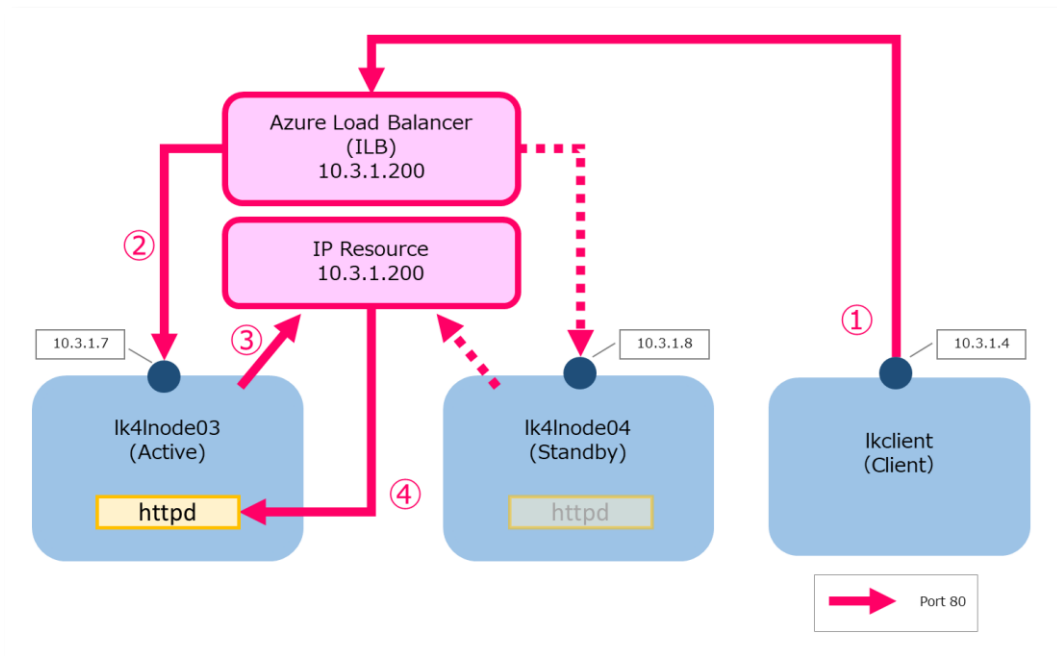
#### 1. 構成



保護対象アプリケーションは Apache で、クライアント PC から Web ブラウザ等による接続確認を行いました。

稼働系サーバー (Ik4Node03) と待機系サーバー (Ik4Node04) はそれぞれ異なる zone に属しています。クライアント (Ikclient) からの http リクエストは 80 番でプ

ローブしている ILB 経由でアクティブなサーバーの仮想 IP (10.3.1.200) にリダイレクトされます。2. ILB によるクライアントリダイレクトの動作概要



ILB によるクライアントリダイレクトの動作概要

クライアント PC からの接続動作の概要は以下の通りです。

- ① Client が httpd へ接続する為、10.3.1.200(port80)への接続を ILB 経由で開始します。
- ② ILB に負荷分散先 (backend pool) に 10.3.1.7 (稼働系 : lk4lnode03)と 10.3.1.8 (待機系 : lk4lnode04) が登録されています。ILB の正常性プローブに Apache の 80 ポートを指定することで、ILB は 10.3.1.7 (稼働系ノード) と 10.3.1.8 (待機系ノード) の 80 ポートのヘルスチェックを行います。Apache の 80 ポートが open になっているのは 10.3.1.7 (稼働系ノード) なので、ILB が受け取った 10.3.1.200(port80)宛のパケットは 10.3.1.7 (稼働系ノード) へ転送されます。
- ③ LifeKeeper の仕様上 Apache リソースは[IP リソースが保護する VIP]を指定する必要があります。その為、受信に IP リソース (10.3.1.200) を作成します。LifeKeeper の IP リソース (10.3.1.200) が In-Service になっているのは稼働系のみとなる為、実際には 稼働系 (10.3.1.7) のみリクエストを受け取ります。

- ④ 結果として Client からの接続要求は稼働系の httpd (10.3.1.200 : port80) が受け取ります。

## 7.2 Azure の構成

Availability Zone を使用するには VM の作成時に以下のように設定します。

- ①[Region]で「Availability zone」をサポートするリージョンを選択します。※ 1
- ②[Availability options]項目で「Availability zone」を選択します。
- ③[Availability zone]項目で、zone を[ 1 , 2 , 3 ]のいずれかを選択します。

稼働系と待機系はそれぞれ異なるゾーン（稼働系(Zone 1)、待機系 (Zone 2)となるようにします。

※ 1 例えば [(Asia Pacific) Japan East]は「Availability zone」をサポートするリージョンです。 [Japan West]のような Availability Set しか利用できないリージョンを選択すると[Availability options]項目で「Availability zone」が選択できません。

保護対象アプリケーションは Apache なので ILB 設定でプローブするポートは 80 番を設定します。このほかの設定は Availability Set を使用した構成と同じです。

Availability zone を使用した構成の環境は以下の通りです。

### 1. ネットワーク、ロードバランサー

#### [Network]

VNet : 10.3.0.0/16

Subnet :

Subnet1 : 10.3.1.0/24

Subnet2 : 10.3.2.0/24

#### [Load Balancer]

Location : Japan East

SKU : Standard

Availability Zone : Zone redundant

FrontendIP:

Type : Private

Subnet : 10.3.1.0/24

IP Address : 10.3.1.200(Static)

BackendPool :

lk4lnode03  
lk4lnode04  
Health Probe :  
    Protocol : TCP  
    Port : 22  
Load Balancing Rule :  
    Port : 80  
    Backend Port : 80  
Inbound NAT rules :  
    0 rules  
Session persistence : None  
Floating IP(Direct Server Return) : Enable

## 2. 仮想サーバー

### [Active Node]

Name : lk4lnode03  
Operating system : Linux(RHEL 7.5)  
Size : Standard A1 v2 (1 vcpu, 2 GiB memory)  
Network Interface :  
    Private IP : 10.3.1.7  
    Public IP : 52.156.45.126  
Installed Software :  
    httpd : httpd-2.4.6-80.el7\_5.1.x86\_64  
    iptables : iptables-1.4.21-24.1.el7\_5.x86\_64  
    LifeKeeper : SPS for Linux v9.5.0 Build 108  
IP tables :

```
...  
-A PREROUTING -p tcp -m tcp --dport 80 -j DNAT --to-destination 10.3.1.200:80  
-A POSTROUTING -s 10.3.1.200/32 -p icmp -j SNAT --to-source 10.3.1.8  
...  
:OUTPUT ACCEPT [4188:806503]  
-A INPUT -s 10.3.0.0/16 -p tcp -j ACCEPT  
...
```

httpd.conf :

```
...  
Listen 10.3.1.200:80...  
...  
ServerName 10.3.1.200:80  
...
```

### [Standby Node]

Name : lk4lnode04

Network Interface :

Private IP : 10.3.1.8

Public IP : None

IP tables :

```
...
-A PREROUTING -p tcp -m tcp --dport 80 -j DNAT --to-destination 10.3.1.200:80
-A POSTROUTING -s 10.3.1.200/32 -p icmp -j SNAT --to-source 10.3.1.7
...
:OUTPUT ACCEPT [4188:806503]
-A INPUT -s 10.3.0.0/16 -p tcp -j ACCEPT
...
```

そのほか、Active Node と同じ。

### [Client Node]

Name : lkclient

Operating System : Windows (Windows Server 2012 Datacenter)

Size : Basic A1 (1 vcpus, 1.75 GiB memory)

Network Interface :

Private IP : 10.3.1.4

Public IP : 104.41.171.59

Installed Software :

Google Chrome : Version 80.0.3987.149 (Official Build) (64-bit)

Xming : 6.9.0.31

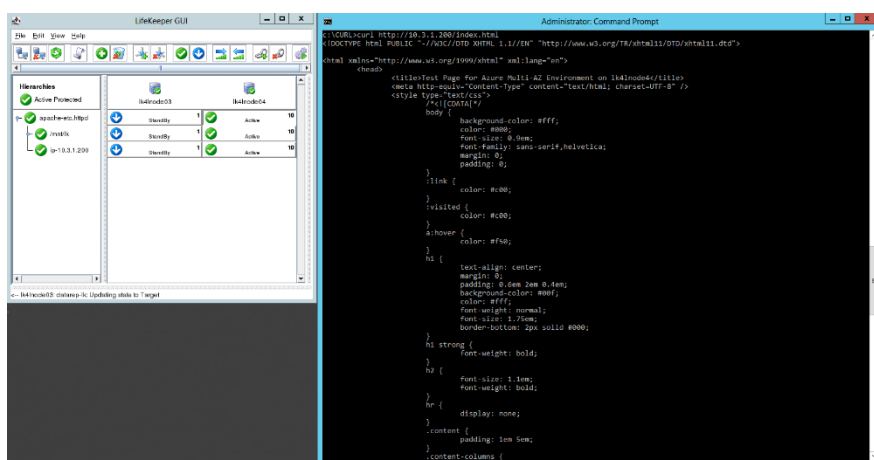
Tera Team : 4.102

WinSCP : 5.17.2

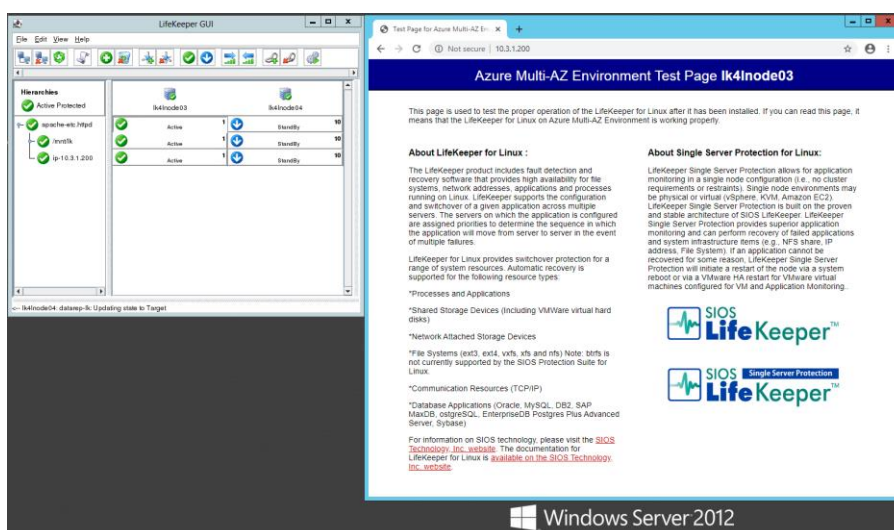
## 7.3 クライアントリダイレクトの動作確認方法

lkclient から、以下 2 つの方法でアクセスを確認し、いずれの方法でも仮想 IP への接続が行えることを確認しました。

1. WindowsServer2012 (クライアント) のターミナルから curl (コマンドラインベースの http アクセスツール) で <http://10.3.1.200/> にアクセスし稼働系の index.html が表示されることを確認しました。  
スイッチオーバーを行いに伴い待機系の index.html が表示されることを確認しました。



2. クライアントの Web ブラウザ上で <http://10.3.1.200/> を入力し Web ページにアクセスし index.html が表示されることを確認しました。(\*確認の都度キャッシュをクリアする)  
スイッチオーバーに伴いスイッチオーバー先の index.html が表示されることを確認しました。



## 8 参考資料

---

サイオステクノロジー :

[SIOS Technical Documentation](#)

[SIOS Protection Suite for Linux テクニカルドキュメンテーション](#)

[Oracle Recovery Kit 管理ガイド](#)

[LifeKeeper/DataKeeper ユーザーポータル](#)

社外リンク :

Microsoft Azure ホームページ

<http://azure.microsoft.com/ja-jp/>

Microsoft Azure Linux 仮想マシンのドキュメント

<https://docs.microsoft.com/ja-jp/azure/virtual-machines/linux/>

Microsoft Azure Japan Team Blog (ブログ)

<http://blogs.msdn.com/b/windowsazurej/>

High Availability for a file share using WSFC, ILB and 3rd-party Software SIOS Datakeeper (ILB の設定で参考資料として利用)

<http://azure.microsoft.com/blog/2014/11/11/high-availability-for-a-file-share-using-wsfc-ilb-and-3rd-party-software-sios-datakeeper/>

## 9 謝辞

---

本ドキュメント作成に際して、日本マイクロソフト様より Microsoft Azure の利用をご承認いただき、また環境構築や設定、テストケースの作成に際して多大なるご支援をいただきました。御礼を申し上げます。

**Microsoft Azure ホームページ**

<http://azure.microsoft.com/ja-jp/>

## 10 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は以下となります。

### ■ LifeKeeper 製品の導入を検討中のお客様

弊社製品ご購入前窓口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://lp.sios.jp/BC\\_web\\_salesvisit.html](https://lp.sios.jp/BC_web_salesvisit.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](https://bc.sios.jp/support_lk.html)

## 11 免責事項

---

- 本書に記載された情報は予告なしに変更、削除される場合があります。最新のものをご確認ください。
- 本書に記載された情報は、全て慎重に作成され、記載されていますが、本書をもって、その妥当性や正確性についていかなる種類の保証もするものではありません。
- 本書に含まれた誤りに起因して、本書の利用者に生じた損害については、サイオステクノロジー株式会社は一切の責任を負うものではありません。
- 第三者による本書の記載事項の変更、削除、ホームページ及び本書等に対する不正なアクセス、その他第三者の行為により本書の利用者に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

〒106-0047 東京都港区南麻布 2-12-3 サイオスビル

電話: 03-6401-5111

FAX: 03-6401-5112

URL: <http://www.sios.com>