

# LifeKeeper for Linux on IIJ GIO

## クラウド検証レポート



## — 目次 —

1. はじめに .....	3
2. システム構成 .....	3
2.1 構成図 .....	3
2.2 クラウド構成 .....	4
2.3 ソフトウェア構成 .....	4
2.4 LifeKeeper の構成 .....	4
3. 検証内容 .....	5

## 1. はじめに

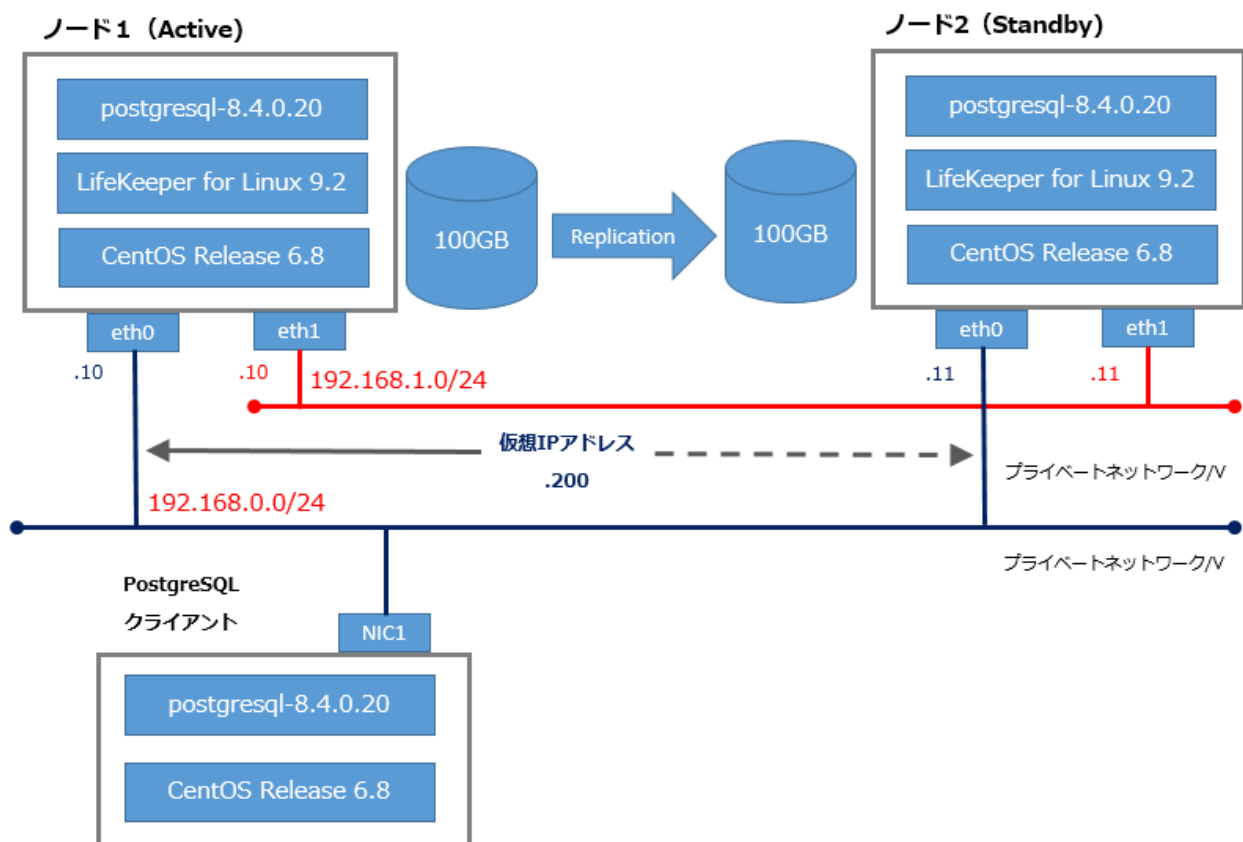
インターネットイニシアティブ社が提供するパブリッククラウドサービス IJ GIO インフラストラクチャーP2 での LifeKeeper for Linux の動作検証をいたしましたので、本書に検証構成と検証内容を記載いたします。

## 2. システム構成

検証実施時の構成は以下のとおりです。

### 2.1 構成図

#### IJ GIO インフラストラクチャーP2 パブリック環境



## 2.2 クラウド構成

- ノード1、ノード2システム構成

サーバー品目	VB0-1(ベストエフォートタイプ)
CPU	1vcore
メモリ	1GB
ネットワーク	プライベートネットワーク/v x 2
システムストレージ	S30GB/CentOS 6
追加ストレージ	B100G (ベストエフォートタイプ)

- 確認用 PostgreSQL クライアント構成

サーバー品目	VB0-1(ベストエフォートタイプ)
CPU	1vcore
RAM	1GB
ネットワーク	プライベートネットワーク/v x 1
システムストレージ	S30GB/CentOS 6

## 2.3 ソフトウェアの構成

OS	CentOS 6.8
クラスタソフトウェア (LifeKeeper)	SteelEye Protection Suite for Linux Version 9.1.2 Release 6609
Recovery Kits	DataKeeper for Linux LifeKeeper PostgreSQL RDBMS Recovery Kit
PostgreSQL	postgresql-8.4.20

## 2.4 LifeKeeper の構成

LOCAL	TAG	ID	STATE	PRIO	PRIMARY
LKDK01	pgsql-5432	LKDK01.pgsql-5432	ISP	1	LKDK01
LKDK01	/postgres	/postgres	ISP	1	LKDK01
LKDK01	datarep-postgres	/dev/vdb1	ISP	1	LKDK01
LKDK01	ip-192.168.0.200	ISP	1	LKDK01	
MACHINE	NETWORK	ADDRESSES/DEVICE	STATE	PRIO	
LKDK02	TCP	10.203.0.24/10.203.0.25	ALIVE		1
LKDK02	TCP	192.168.0.10/192.168.0.11	ALIVE		2

## 3. 検証内容

以下のとおり検証を実施し、結果に問題がないことを確認しました。

### 1 システム起動確認

- 1.a システム起動後、アクティブノード側で全保護対象リソースステータスがアクティブ、スタンバイノード側で Volume リソースがミラーリング、それ以外がスタンバイのステータスとなっていること。
- 1.b アクティブノードで PostgreSQL が起動し、データベースにログインし、SQL クエリの発行したときに正常に結果が返ること。スタンバイノードで PostgreSQL が停止していること。

### 2 LifeKeeper 停止確認

- 2.a lstop コマンドを実行し、LifeKeeper が正常に停止できること。
- 2.b LifeKeeper が停止した後、仮想 IP リソースが停止していること。
- 2.c LifeKeeper が停止した後、保護対象のディスクがマウントされていないこと。

### 3 手動リソース切り替え確認

- 3.a スタンバイノードでリソースを起動させ、全てのリソースが正常に起動すること。
- 3.b 切り替え元のノードで保護対象のディスクがマウントされていないこと。
- 3.c 切り替え元のノードで PostgreSQL が停止していること。

### 4 仮想 IP アドレス障害時のフェイルオーバー確認

- 4.a アクティブノードで仮想 IP アドレスが使用しているネットワークインタフェースを停止し、意図的に障害を引き起こす。その後、仮想 IP リソースの障害を検知し、スタンバイノードへ正常にフェイルオーバーすること。
- 4.b 停止させたネットワークインタフェースを起動させた際にフェイルオーバーが発生しないこと。
- 4.c 障害ノードで保護対象のディスクがマウントされていないこと。
- 4.d 障害ノードで PostgreSQL が停止していること。

### 5 アプリケーションリソース障害によるフェイルオーバー確認

- 5.a アクティブノードで PostgreSQL を停止し、意図的に障害を引き起こす。その後、PostgreSQL リソースの障害を検知してスタンバイノードへフェイルオーバーし、リソースステータスが全てアクティブになっていること。

### 6 ノード障害によるフェイルオーバー確認

- 6.a OS を強制停止しスタンバイノードが障害を検知し、スタンバイノードにフェイルオーバーすること。
- 6.b スタンバイノードにフェイルオーバー後、スタンバイノードの PostgreSQL にログインして SQL クエリを発行し結果が正常に返ってくること。