SIOS Protection Suite for Linux

Amazon EC2 Cross Region

v9.2

Quick Start Guide

October 2017

To maintain the quality of our publications, we welcome your comments on the accuracy, clarity, organization, and value of this document.

Address correspondence to:
ip@us.sios.com

Table of Contents

Chapter 1: Introduction

Starting with v8.3 LifeKeeper for Linux provides support for Amazon EC2 Cross Region configurations. With this support HA cluster configurations can now be created between 2 different regions (Cross Region) allowing services running on Amazon EC2 instances to be switched between the Regions. This document includes requirements and basic operations for using Amazon EC2 Cross Region configuration support with LifeKeeper for Linux.

Please refer to SIOS Protection Suite for Linux documentation and Amazon Web Service (AWS) for technical information.

Note: The "Amazon Web Services" logo, the "Powered by Amazon Web Services" logo, "AWS", "Amazon EC2", "EC2", "Amazon Elastic Compute Cloud", "Amazon Virtual Private Cloud", "Amazon Route53", and "Amazon VPC" are registered trademarks of Amazon.com, Inc or related companies in the United States and other countries.

Chapter 2: Configuration

The following example can be used for creating a redundant configuration for the protection of back-end services using LifeKeeper for Linux support for Amazon EC2 Cross Region configurations. This can be accomplished using local IP addresses in Amazon EC2 environments. This allows for the switching of services between regions and provides business continuity solutions for a wider range of failures.

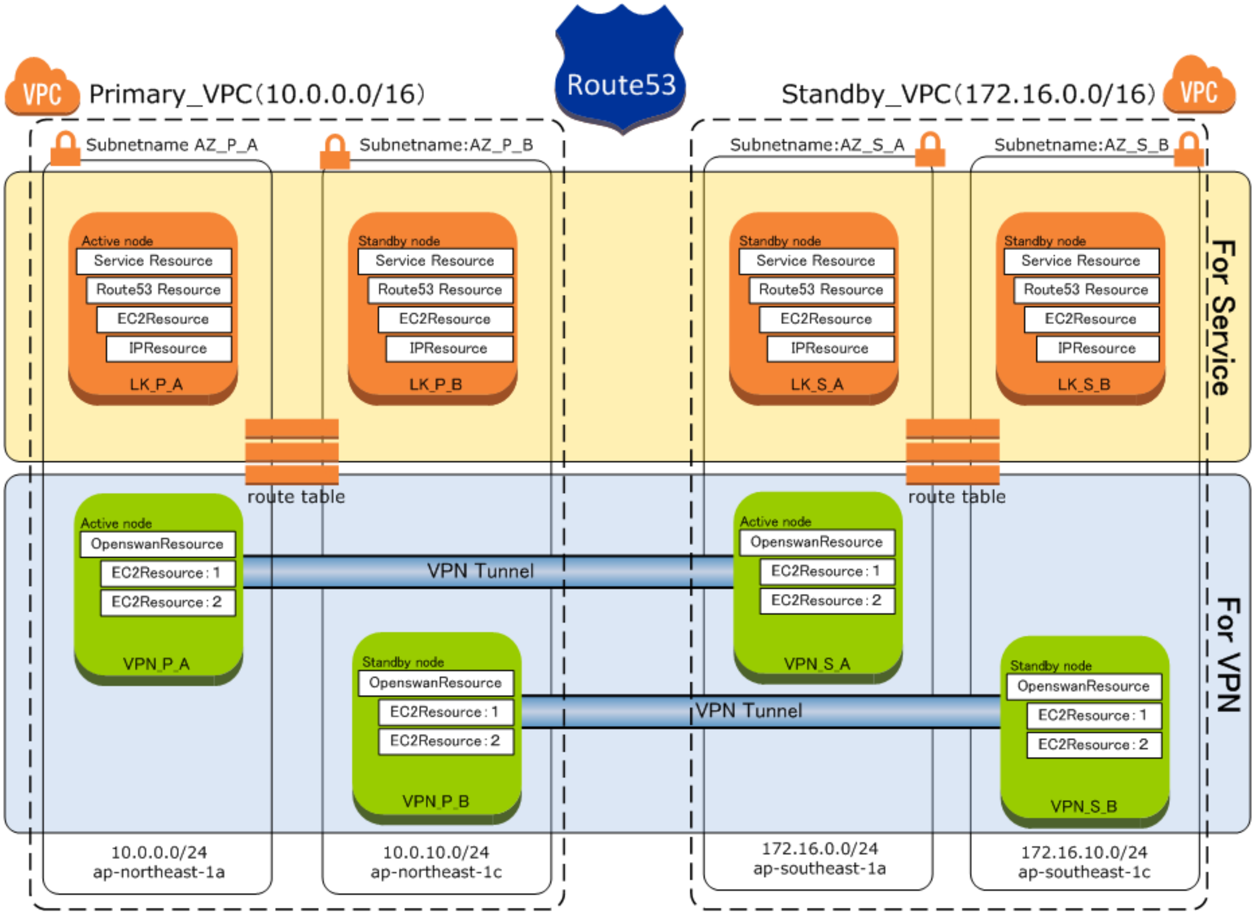


Figure 1 Amazon EC2 Cross Region configuration example

The following table contains definitions for the Items used in Figure 1.

Item	Description
Region	A named set of AWS resources in the same geographical area. The “Primary Region” is predefined in this document as the geographical area where the protected resource instance is currently running and providing access to the protected services. The "Standby Region" is predefined in this document to identify the geographical area that is "standing by" to become a "Primary Region" but is not currently active and providing access to the protected services
VPC (Virtual Private Cloud)	A logically independent network area which can be created in one Region.
Region AZ (Availability Zone)	A distinct location within a region that is insulated from failures in other Availability Zones (AZ) within the region. In this document "AZ_P_A" is predefined as an AZ within the Primary VPC where an instance is currently running and providing access to the services. "AZ_P_B" is predefined as a AZ within the Primary VPC in which a standby instance of the protected service resides. Additionally, "AZ_S_A" is predefined as an AZ within the Standby VPC in which a standby instance of the protected service resides. "AZ_S_B" is predefined as a AZ within the Standby VPC in which a standby instance of the protected service resides.
Route Table	A routing table defined when creating a VPC. It defines routes for communication between AZs and communication between Regions.
Route 53	DNS provided by Amazon Route53 service.
Openswan	Openswan is an open-source project that provides a kernel level implementation for Linux IPsec. For more information, refer to the Openswan website.
Service instance	An EC2 instance where an HA cluster is configured by installing a protected service. It is referred to as a "service instance" in this documentation.
VPN instance	An EC2 instance used for construction of a VPN via Openswan which connects to a VPC in another Region. The server which Openswan is installed and used in is referred to a "VPN instance" in this documentation.
Route53 resource	A LifeKeeper resource instance created using the Route53 Recovery Kit (Route53 RK). The resource instance updates a virtual IP address and related DNS records so that clients can continue to access unique URLs when services are switched between Regions.
Openswan resource	A LifeKeeper resource instance created using the Openswan Recovery Kit (Openswan RK). It monitors the process start state of Openswan and the session state of the IPsec-VPN.
EC2 resource	A LifeKeeper resource instance used with Openswan or Route53 resource instances. EC2 resource instances provide a mechanism to recovery Elastic IPs and enables IP resource instances to work in multiple availability zones. It also updates the related Route table entries when a switching between AZs or between Regions to ensure continued communications with the protect services.
IP resource	A LifeKeeper resource instance created using the IP Recovery Kit (IP RK). It creates and protects switchable virtual IP addresses.

Notes: An Internet Gateway (IGW) is used for the communication between Regions.

Notes: Refer to AWS documentation and LifeKeeper Online Documentation for the terminologies and the details for AWS. Also refer to Related LifeKeeper Resources for the details of LifeKeeper related resources for the configuration described in this guide.

Service Instances

This section describes each LifeKeeper resource instance role in this configuration (see Related LifeKeeper Resources for more information). Figure 1 shows 2 nodes are configured in the VPC in the Primary Region and 2 nodes are configured in the VPC in the Standby Region. The LK_P_A LifeKeeper instance in the Primary Region is the primary node (the primary node is the node currently running and providing access to the protected services) and the other LifeKeeper instances are backup nodes (LK_P_B in the Primary Region, LK_S_A and LK_S_B in Standby Region). A failover is performed automatically when it occurs between the AZs in the same Region. If a failover occurs to a different Region, LifeKeeper will not automatically failover. In this case an administrator must switch the resources over manually to bring them in service.

Oracle 12c or PostgreSQL 8.3, 8.4, 9.0, 9.1, 9.2, 9.3, 9.5 are the only available services in this release of EC2 Cross Region configuration (no other versions of Oracle or PostgreSQL are supported). Either the Oracle or PostgreSQL Recovery Kit is required to protect these database services with LifeKeeper. When either the Oracle or PostgreSQL Recovery Kit is used with the EC2 Cross Region configuration, all of the file system resources the Recovery Kits protects must reside on a replicated device in order for the data to be "shared" between availability zones. "DataKeeper for Linux" provides the necessary data replication services to meet this requirement and is required with these Recovery Kits.

VPN Instances

When configuring an HA cluster across Regions LifeKeeper must be able to communicate with the nodes in the other Region. This is done by configuring VPN servers using Openswan. In Figure 1 the VPN servers are VPN_P_A and VPN_P_B in the Primary Region and VPN_S_A and VPN_S_B in the Standby Region. These VPN instances establish VPN tunnels between a node in the Primary Region and a node in the Standby Region and it is through these tunnels that connections to the opposing Region are established. The LifeKeeper Openswan Recovery Kit is used to secure redundancy of the VPN connections.

LifeKeeper Protection

In EC2 Cross Region configurations LifeKeeper is used to provide HA for the configured applications and services. Access to the applications and services is provided via a switchable private IP address that is also protected by LifeKeeper. When creating the IP resource with LifeKeeper and extending to the other Regions different IP addresses may need to be specified to allow a common access point to the protected applications and services. The Route53 resource instance created by the LifeKeeper Route53 Recovery Kit ensures this common access point when switching to another Region.

When a switchover to another Region occurs, the Route53 resource instance uses the Amazon Route 53 services to update DNS records. This is done so the service corresponds to the IP address of the network segment in the Region in which the protected applications and services now reside. This allows a client to connect to a protected application and be able to maintain that connection regardless of the changes to the IP address because of Region switchovers.

The communication paths between AZs and Regions are updated when a switchover of the resource hierarchy for the service instances and/or the VPN instances occurs. These updates are done by EC2 resource which is dependent resource in the hierarchy.

In the EC2 Cross Region configuration, HA is possible between AZs because of LifeKeeper's control and protection of connections done by updating the communication paths and name resolution.

Chapter 3: Requirements

Prior to using LifeKeeper for Linux Amazon EC2 Cross Region Support, be sure your configuration meets the following Amazon Web Service (AWS) and EC2 requirements.

Amazon Web Service (AWS) Requirements

- Create an environment on AWS as a base for the Amazon EC2 Cross Region configuration.
- Setup a VPC in each Region. (Note: An HA cluster configuration between more than 2 VPCs is not currently supported.) For an HA cluster configuration between Availability Zone(AZ)s in a VPC, you must use the LifeKeeper for Linux "Recovery Kit for EC2".
- Two subnets are required in the AZ of each Region. Each node in the cluster (Primary and all Standby) must be deployed in different AZs in order to secure redundancy within the Region.
- AWS Administrator rights, an AWS access key ID, and a secret access key are required.
- A LifeKeeper instance and a VPN instance are required in each AZ (for a total of 8 EC2 instances).
- A registered Amazon Route 53 domain name is required to create a LifeKeeper for Linux Route53 resource.
- All clients accessing services protected by LifeKeeper Route53 resource instances must be able to perform host name resolution to the Amazon EC2 Cross Region configuration.

Requirements for Creating an EC2 Instance

You will create a total of 8 EC2 instances for an EC2 Cross Region Configuration solution. Refer to Setting Up Your SPS Environment for configuration settings.

The following are the common requirements for all VPN and Service instances. Refer to SIOS Protection Suite Installation Guide for installation information.

- Install LifeKeeper in the EC2 instances prepared for this configuration. Refer to Distribution and Software Requirements for more information.

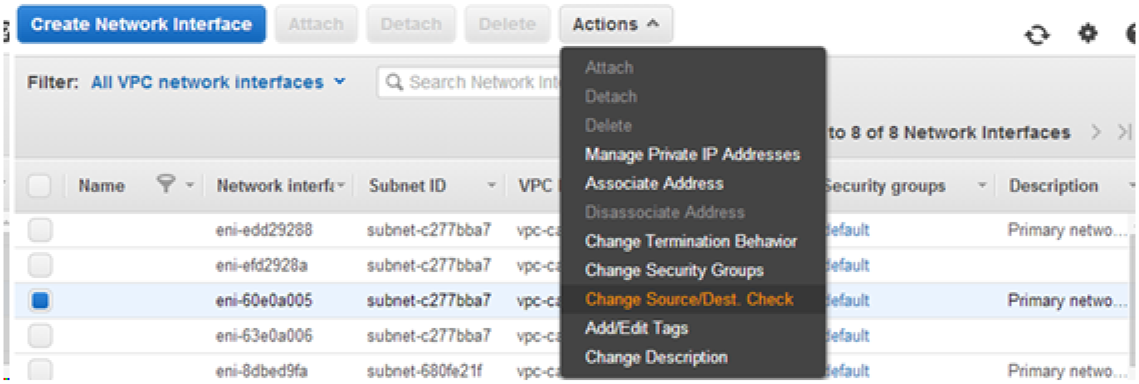
DataKeeper for Linux is used to provide replication services to ensure all file system data for the protected applications is available to all nodes in the configuration. Refer to DataKeeper for Linux Hardware and Software Requirements for more information

Note: Confirm the **Security Groups** settings in **NETWORK & SECURITY** in configuring EC2 instances. There are some ports used for communication in the requirements for using LifeKeeper.

- When configuring an HA cluster in the same Region prepare the instances so that the primary instances and the standby instances run in different AZs.

- Install Amazon EC2 API Tools in advance. This is required to run the related LifeKeeper resources. Refer to the API tools documentation for installation information.
- Set **Change Souce/Dest Check** to **Disabled** for the ENI of each HA cluster instance.

  Select **Actions** > **Change Source/Dest Check** in the EC2 instance management console to change the settings.



	Name	Network interfa	Subnet ID	VPC	ecurity groups	Description
□		eni-edd29288	subnet-c277bba7	vpc-ca	efault	Primary netwo...
□		eni-efd2928a	subnet-c277bba7	vpc-ca	efault	
■		eni-60e0a005	subnet-c277bba7	vpc-ca	efault	Primary netwo...
□		eni-63e0a006	subnet-c277bba7	vpc-ca	efault	
□		eni-8dbed9fa	subnet-680fe21f	vpc-ca	efault	Primary netwo...

Figure 2 Change Source/Dest Check

VPN Instance

- Install Openswan for the VPN instances. The VPN instances must meet the Openswan and LifeKeeper installation requirements.
- Establish a VPN session.

Service Instance

- If Oracle is installed in the Service instances, the Service instances must meet Oracle and LifeKeeper installation requirements.

  Note: If Oracle 12c is installed in the Service instances it becomes the protected application.
- Create one or more dedicated local disks or partitions to be used for the file systems used by the protected application. The data on these file systems will be replicated to the other nodes in the configuration using DataKeeper for Linux.
- If you use Oracle as a protected application, create the Oracle resource instance using the Oracle Recovery Kit (Oracle RK). When creating the Oracle resource, deploy the database, archive files, log file and control file on the replicated file systems (see previous bullet).

The disk size or the disk configuration depends on the scale of the database and the database configuration. Refer to Oracle Specific Configuration Considerations and Configuration Examples for more information.

Distribution and Software Requirements

This version supports the following software.

Category	OS and Software
Distribution for each instance	Redhat Enterprise Linux v6.4/6.5/7.0 64-bit CentOS v6.4/6.5/7.0 64-bit
VPN instance protected software	Openswan rpm, or Libreswan rpm included in the distributions listed above
Service instance protected software	Oracle 12c (no other versions are supported) PostgreSQL 8.3, 8.4, 9.0, 9.1, 9.2, 9.3, 9.5 (no other versions are supported)

We will keep to widen a scope of application of these softwares to be protected and distributions to be supported. If installing Libreswan rpm for the VPN instances, "Openswan" shall be replaced with "Libreswan".

Chapter 4: Setting Up Your Environment

The following describes the steps for setting up an Amazon EC2 Cross Region configuration protected by LifeKeeper.

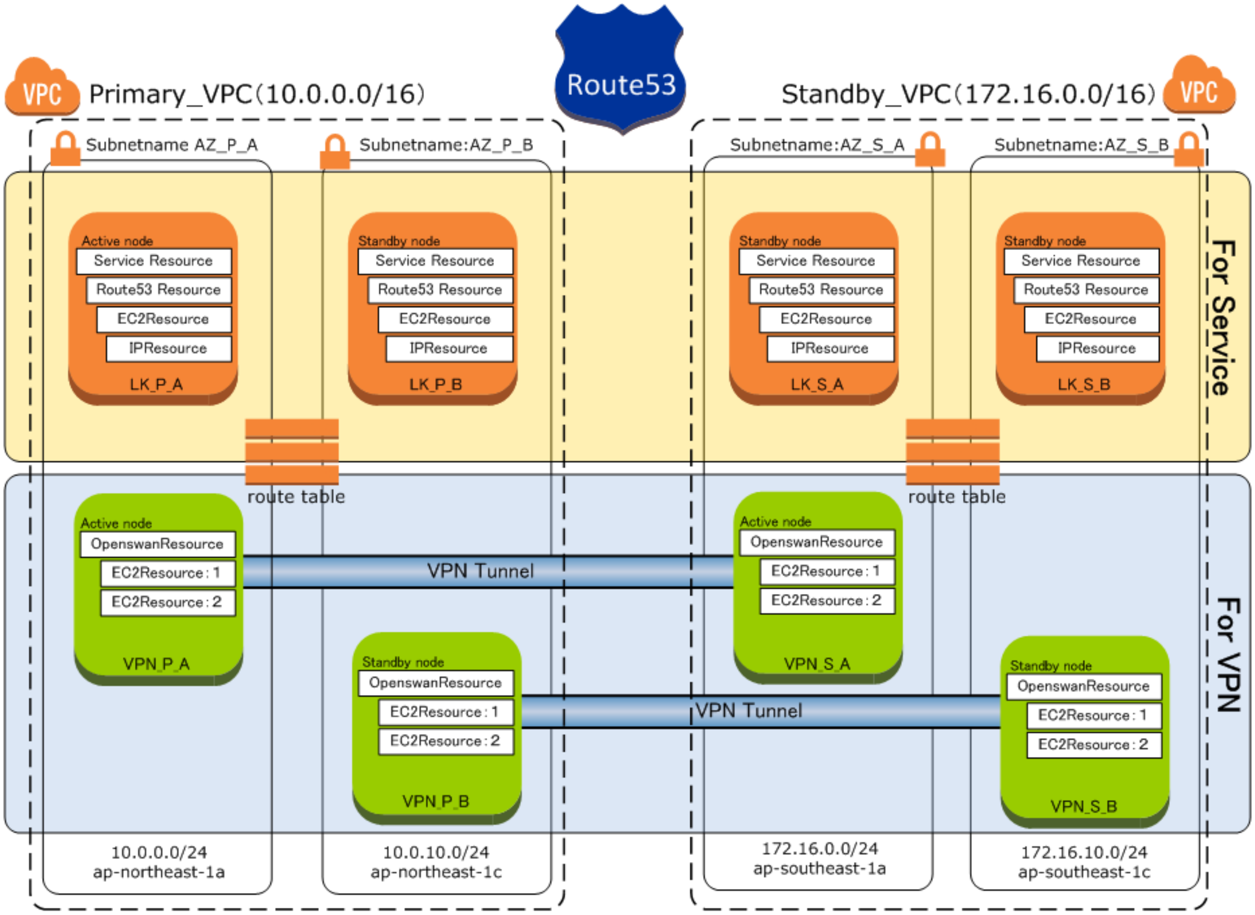


Figure 1 Configuration Example

- Communication between Regions passes through IGW (not shown in this example).
- When creating the LifeKeeper IP resource instance the virtual IP address for the Primary_VPC should be set to 10.1.0.10/32. When the LifeKeeper IP resource is extended to the Standby_VPC the virtual IP address should be set to 10.2.0.10/32 (**Note:** IP addresses in this section are only for the example).

1. Create 2 VPCs in different Regions

Refer to Amazon Web Service documentation for information on creating a VPC.

In the Figure 1 Configuration example, the VPC in the Primary Region (known as the Primary_VPC) and the VPC in the Standby Region (known as the Standby VPC) were created using 10.0.0.0/16 for the CIDR Block in the Primary Region and 172.16.0.0/16 for the CIDR block in the Standby Region.

2. Create 2 subnets on a VPC and deploy each of them in different AZs

Create 2 subnets for the VPCs created in step 1 specifying 1 AZ per subnet. Refer to Amazon Web Service documentation for information on creating a subnet. In the Figure 1 example, the Primary_VPC located in the Primary_Region the 10.0.0.0/24 subnet (ap-northeast-1a AZ) and the 10.0.10.0/24 subnet (ap-notheast-1c AZ) have been created. Additionally, in the Standby_VPC located in the Standby_Region the 172.16.0.0/24 subnet (ap-southeast-1a AZ) and the 172.16.10.0/24 subnet (ap-southeast-1c AZ) have been created.

3. Create a service instance and a VPN instance in each AZ (8 total)

Refer to Requirements for creating - an EC2 Instance for requirements and configuration information.

4. Define a Domain name in a hosted zone for Amazon Route 53

A Route53 resource (created in a later step) adds a virtual host name as an A record in the hosted zone's DNS. DNS records are updated when switching between Regions.

5. Add routing information in the Route Tables for Amazon Route 53

Open the **Routing table** screen for the VPC. Click the **Routes** tab to display the Routing table. This is an example for the default Routing table of the Primary_VPC from Figure 3.

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

You must specify Routing information in the default Route table based on the following conditions:

- The Virtual IP address running in this VPC area
- The subnet for the Virtual IP address running in the opposite region.
- The Subnet of the VPC in the opposite Region.

The following example shows the changes to the Route table for the Primary_VPC in the Figure 1 example.

Primary _VPC Route table	
Destination	**Target**
10.0.0.0/16	local
0.0.0.0/0	igw-xxxxxxx
10.1.0.10/32	eni-zzzzzzzz : ENI associated to a virtual IP of a service instance (ENI of LK_P_A in the configuration example)
10.2.0.0/16	eni-yyyyyyyy : ENI of a VPN instance (ENI of VPN_P_A in the configuration example)
172.16.0.0/16	eni-yyyyyyyy : ENI of a primary node of the same VPN instance as above (ENI of VPN_P_A in the configuration example)

Note: The text in red above must be added and saved. The same Route Table settings are required in both the Primary and Standby Region. The following example shows sample settings for the Standby Region.

Standby _VPC Route table	
Destination	**Target**
172.16.0.0/16	local
0.0.0.0/0	igw-xxxxxxx
10.2.0.10/32	eni-aaaaaaaa : ENI associated to a virtual IP of a service instance (ENI of LK_S_A in the configuration example)
10.1.0.0/16	eni-bbbbbbbb: ENI of a VPN instance (ENI of VPN_S_A in the configuration example)
10.0.0.0/16	eni-bbbbbbbb: ENI of a primary node of the same VPN instance as above (ENI of VPN_S_A in the configuration example)

6. Install the Openswan software.

The Openswan software must be installed and configured on each VPN instance so a VPN session can be establish to a VPN instance in another Region. The Openswan software must be installed and configured and VPN session established before creating a LifeKeeper Openswan resource instance.

The Openswan software can be installed via an rpm package that is included with the supported distribution software. Add the package using the normal installation command `(rpm or yum)`. Specify necessary configuration settings after you install Openswan in all VPN instances. The following is an example of ipsec.conf, configuration file for Openswan.

```
version 2.0
config setup
protostack=netkey
nat_traversal=yes
virtual_private=
oe=off
conn vpn1
type=tunnel
authby=secret
left=%defaultroute
leftid=54.249.2.50
#Public IP address of VPN_P_A
leftnexthop=%defaultroute
leftsubnets={10.0.0.0/16,10.1.0.0/16}
#subnet of Primary_VPC and subnet of virtual IP address
leftsourceip=10.0.0.5
#local address of VPN_P_A
right=54.254.156.254
#public IP address of VPN_S_A
rightsubnets={172.16.0.0/16,10.2.0.0/16}
#subnet of Standby_VPC and subnet of virtual IP address
rightsourceip=172.16.0.5
pfs=yes
auto=start
```

Be aware of the following values specified in the configuration file `left, leftid, leftsubnets, leftsourceip, right, rightid, rightsubnets, and rightsourceip` as they are used in the creation of the LifeKeeper Openswan resource.

- The public IP address of the VPN server should be specified for left or leftid.
- The local Subnet of the VPN server should be specified for leftsubnets.

- The local IP address of the VPN server should be specified for leftsourceip. The right side is the same.
- Left and right configuration parameters specify settings for the Primary and Standby sides of the VPN session. Both must be specified.

Verify the VPN sessions are established between VPN_P_A and VPN_S_A and between VPN_P_B and VPN_S_B and that communications have been established . Leave the VPN sessions open that have been established. Create the LifeKeeper resources once the 2 VPN connections are live.

7. Install the LifeKeeper Software on each EC2 instance

Refer to SIOS Protection Suite Installation Guide for installation information.

Additional considerations for this configuration are as follows.

- Install the Openswan RK, the Route53 RK, and the EC2 RK using the rpm command. The kits are located in the `Amazon` directory below the mount path of the LifeKeeper CD image.
- Use SSH-X forwarding to display the LifeKeeper GUI.

  When the start-up of LifeKeeper completes, run the LifeKeeper GUI to create resources instance. SSH -X forwarding is used for displaying the LifeKeeper GUI on your management console in the EC2 environment. Refer to Running the LifeKeeper GUI Through a Firewall SSH -X forwarding information.

8. Tune LifeKeeper for running in EC2 Cross Region configurations

Set the LifeKeeper tuneable values needed for this configuration before creating any LifeKeeper resource instances. The settings for VPN and Service instances are different and are specified below. LifeKeeper will need to restarted to pick up these changes.

- Required settings for all VPN instances
  - Change the value for LKCHECKINTERVAL in the `/etc/default/LifeKeeper` file from the default of 120 to 260 seconds.
- Required settings for all Service instances
  - Change the value for LKCHECKINTERVAL in the `/etc/default/LifeKeeper` file from the default of 120 to 360 seconds.Restart LifeKeeper to reflect the changes.
  - Change the value for CONFIRMSODEF in the `/etc/default/LifeKeeper` file from the default of 0 to 1.
  - Change the value for NOBCASTPING in the `/etc/default/LifeKeeper` file from the default of 0 to 1 to disable the monitoring using broadcast Ping for the IP resource.
  - Specify **[Set Confirm Failover On]** in the Server Properties to prevent automatic failover initiation between Regions. See the examples below. (Note: the host name corresponds to the configuration of Figure1. It shows the example set and opened Server Properties on each node.)

LK_P_A

Host Name	Set Confirm Failover On	Set Block Resource Failover On
LK_P_A	(none)	(none)
LK_P_B	(none)	(none)
LK_S_A	✔	(none)
LK_S_B	✔	(none)

LK_P_B

Host Name	Set Confirm Failover On	Set Block Resource Failover On
LK_P_B	(none)	(none)
LK_P_A	(none)	(none)
LK_S_A	✔	(none)
LK_S_B	✔	(none)

LK_S_A

Host Name	Set Confirm Failover On	Set Block Resource Failover On
LK_S_A	(none)	(none)
LK_P_A	✔	(none)
LK_P_B	✔	(none)
LK_S_B	(none)	(none)

LK_S_B

Host Name	Set Confirm Failover On	Set Block Resource Failover On
LK_S_B	(none)	(none)
LK_S_A	(none)	(none)
LK_P_A	✔	(none)
LK_P_B	✔	(none)

The flg_list command can be used if checking the content set on GUI by the command.

After setting as the example, you can check the confirmso! flag below is created.

Host Name	Confirmso! flag
LK_P_A	confirmso!LK_S_A confirmso!LK_S_B
LK_P_B	confirmso!LK_S_A confirmso!LK_S_B
LK_S_A	confirmso!LK_P_A confirmso!LK_P_B
LK_S_B	confirmso!LK_P_A confirmso!LK_P_B

9. Specify a communication path in the VPN instance

The VPN instance configures an HA cluster with a pair of instances in the same Region.

Note: You don't need to configure a communication path across Regions.

- Configure a communication path between the VPN_P_A instance and the VPN_P_B instance and a communication path between the VPN_S_A instance and the VPN_S_B instance.

Refer to Creating a Communication Path for more information.

10. Create and extend the Openswan resource in the VPN instance in each Region

Run the LifeKeeper GUI on the primary node and launch the **Create Resource** wizard.

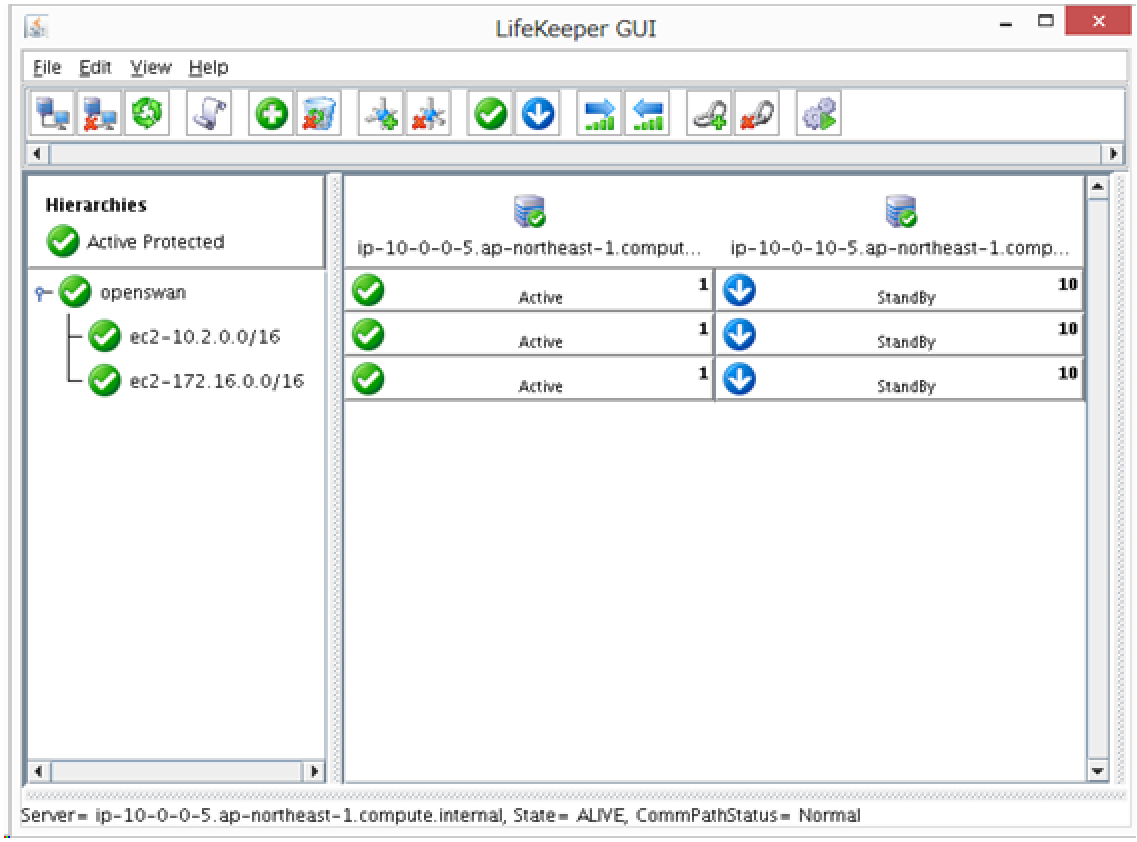
Once the **Create Resource** wizard is launched select Openswan to begin the creation process. The Create Wizard Dialog prompts and the associated input values are listed in the table below.

Item	Setting
Creation	
Select Recovery Kits	Choose Openswan.
Switchback Type	Enter the switchback type. Automatic or intelligent.
Server	Select the node of the primary server for the Openswan resource hierarchy. VPN_P_A in the Primary Region was selected in the example.

Item	Setting
IPSec connection name	Select an IPSec connection name. The value is obtained from the Openswan configuration file.
EC2 Home	Select or enter the EC2_HOME directory path. EC2_HOME is the path to the EC2 API Tools. Note: Default value is /opt/aws. Confirm whether `<$EC2_HOME>/bin/ec2-describe-addresses` exists as it is used for validation of inputs.
EC2 URL	Select or enter the actual EC2_URL. The EC2_URL is the URL of Amazon EC2 Web service end point.
AWS Access Key	Enter the access key for AWS. The AWS access key is the access key ID used for user identification. REST or query protocol requests against the AWA service API are protected by using a combination of the access key and the security key.
AWS Secret Key	Enter an AWS security key. The AWS security key is a secret key. REST or query protocol requests against the AWA service API are protected by using a combination of the access key and the security key.
Openswan Tag	Tag name for the Openswan resource instance.
Extension	
Target Server	Select the target node from the list. The resource hierarchy will be extended to this node.
Switchback Type	Select the switchback type automatic or intelligent.
Template Priority	Specify a priority value for the resource extension template node.
Target Priority	Specify a priority value for the backup node, i.e. a resource extension target node.
Openswan Tag	Tag name for the Openswan resource instance.

When the **Create Resource** Wizard completes, the resource hierarchy looks as follows:

Note: An EC2 resource is automatically created as a child resource of the Openswan resource.



Hierarchies

✓ Active Protected

- openswan
  - ec2-10.2.0.0/16
  - ec2-172.16.0.0/16

ip-10-0-0-5.ap-northeast-1.comput...		ip-10-0-10-5.ap-northeast-1.comp...	
✓ Active	1	StandBy	10
✓ Active	1	StandBy	10
✓ Active	1	StandBy	10

Server= ip-10-0-0-5.ap-northeast-1.compute.internal, State= ALIVE, CommPathStatus= Normal

Figure 2 Example of an Openswan Resource Hierarchy

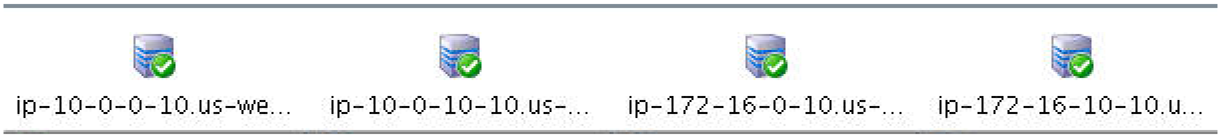
11. Repeat step 10 above on the VPN instance in the other Region.

12. Perform a switchover test on VPN instance

    Initiate a manual switchover for the VPN resource created using the LifeKeeper GUI in previous steps and then restore the resource on the primary node and proceed to the next step.

13. Specify communication paths to the service instances in all Regions

    Specify communication paths for the 4 service instances created across the Regions and configure them as one cluster group. The procedure to specify communication paths is the same as step 9. Perform the procedure so that all 4 nodes can be connected. As a result, all 4 nodes are displayed in the LifeKeeper GUI as follows.



ip-10-0-0-10.us-we... ip-10-0-10-10.us-... ip-172-16-0-10.us-... ip-172-16-10-10.u...

14. Create a virtual private IP address and extend it to each instance in all AZs

Run the LifeKeeper GUI on the primary node and create an IP resource. Refer to Creating an IP Resource Hierarchy and Extending Your Hierarchy for more information.

Consider the following when creating an IP resource.

- Specify the virtual IP address outside of the CIDR allocated in the VPC
- Change the IP tag name from the default to something more meaningful for its intended use, for example <ip-route53>.
- When extending the IP resource to a different region, you must specify a virtual IP address corresponding to the network segment in that Region.

When the IP resource is created, the LifeKeeper GUI displays as follows.

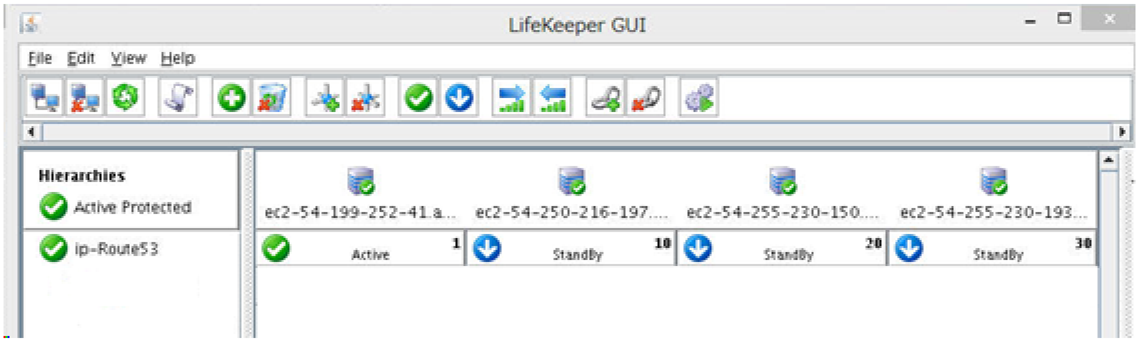


Figure 3 Example of an IP Resource Hierarchy

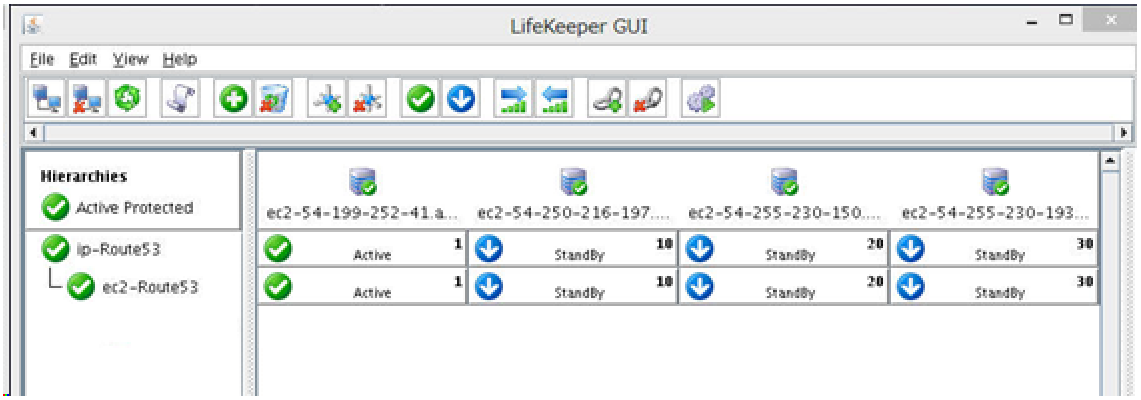
14. Create an EC2 resource for the service instance

Create an EC2 resource for the service instance based on the back-end scenario for the Recovery Kit for EC2. Do this in advance. Refer to Creating a Resource Hierarchy and Extending Your Hierarchy for more information.

Consider the following when creating an EC2 resource.

- Choose the **Route Table Scenario** for the **EC2 Resource Type** in the **Create Resource** wizard.
- Change the EC2 resource tag name from the default to something more meaningful for its intended use, for example <ec2-route53> (if you used <ip-route53> for the IP resource tag name).

When the EC2 resource is created, the LifeKeeper GUI displays as follows.



Hierarchies	ec2-54-199-252-41.a...	ec2-54-250-216-197....	ec2-54-255-230-150....	ec2-54-255-230-193...
✓ ip-Route53	✓ Active 1	StandBy 10	StandBy 20	StandBy 30
✓ ec2-Route53	✓ Active 1	StandBy 10	StandBy 20	StandBy 30

Figure 4 Example of a EC2 Resource Hierarchy

16. Create a Route53 resource on a service instance

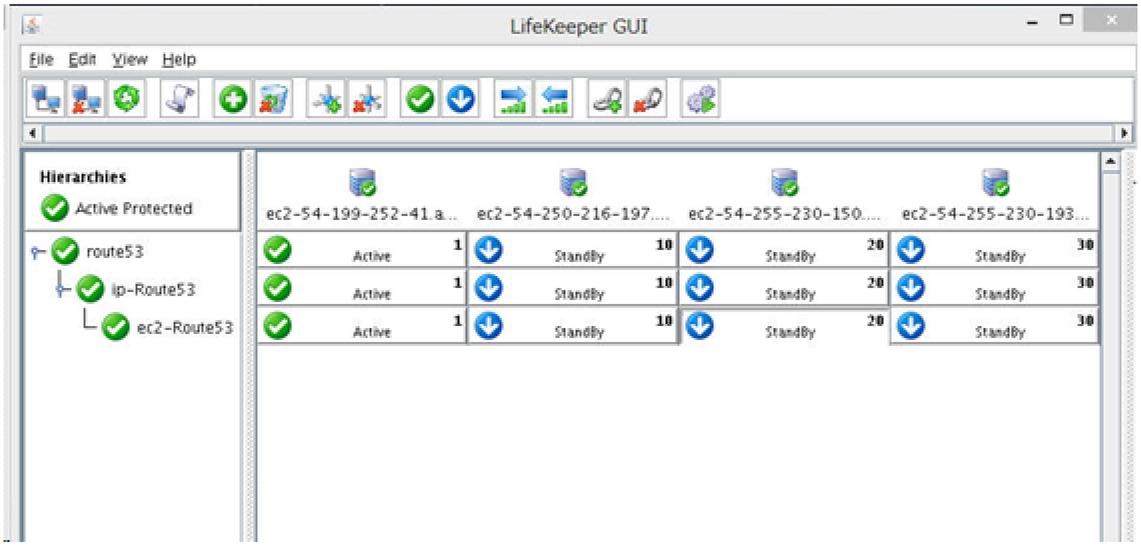
Run the LifeKeeper GUI on the primary node and launch the Create Resource Wizard and enter the following responses to the create wizard dialogs:

Item	Content of Setting
Creation	
Select Recovery Kits	Choose Route53.
Switchback Type	Select the switchback type, intelligent for automatic.
Server	Choose the node of the primary server for the Route53 resource hierarchy. VPN_P_A in the Primary Region is chosen in the example.
AWS Access Key	Enter the access key to connect to Route 53
AWS Secret Key	Enter the secret key to connect to Route 53
Domain name (Route53 hosted zone)	Select a Domain Name from the list provided. The list provide is built from the Route 53 DNS information and should have already been setup. See the requirements for creating AWS environments for more information.
Host Name	Enter the host name to be used for the A record in the Route 53 DNS. Do not enter a Fully Qualified Domain Name.

Item	Content of Setting
IP resource	Select an IP address from the list to register it to a DNS A record. The listed IP addresses are the virtual IP addresses protected by the IP Recovery Kit.
Route53 Resource Tag	Tag name for the Route53 resource instance.
Extension	
Target Server	Select the target node from the list. The resource hierarchy will be extended to this node.
Switchback Type	Select the switchback type automatic or intelligent.
Template Priority	Specify a priority value for the resource extension template node.
Target Priority	Specify a priority value for the backup node, i.e. a resource extension target node.
Route53 Tag	Tag name for the Route53 resource.

Perform the same steps extending a resource on the resource hierarchy to the other target nodes.

When the Route53 resource hierarchy is created, the LifeKeeper GUI displays as follows.



Hierarchies ✓ Active Protected	ec2-54-199-252-41.a...	ec2-54-250-216-197....	ec2-54-255-230-150....	ec2-54-255-230-193...
✓ route53	✓ Active 1	↓ StandBy 10	↓ StandBy 20	↓ StandBy 30
✓ ip-Route53	✓ Active 1	↓ StandBy 10	↓ StandBy 20	↓ StandBy 30
✓ ec2-Route53	✓ Active 1	↓ StandBy 10	↓ StandBy 20	↓ StandBy 30

Figure 5 Example of a Route53 Resource Hierarchy

17. Install the Oracle software and specify the settings to create an Oracle resource hierarchy (refer to Oracle Recovery Kit Administration Guide).

Install the Oracle software using the identical installation settings on all nodes in the cluster. During installation and configuration of the Oracle database instances use the file systems that will become the DataKeeper for Linux mirrors for the directories that must be protected by Oracle Recovery Kit. Prior to the installation the file system must be created and mounted.

When protecting the listener, specify the IP resource tag name used in the Route53 resource that matches the Listen address.

When the installation and configuration is complete, stop the Oracle instances on all servers, and un-mount all disks on the target nodes that will be used for replication.

18. Create DataKeeper resources

A DataKeeper resource instance must be created for each file system that must be protected by the Oracle Recovery Kit so that data can be shared to all nodes in the cluster via replication. For a basic procedure on creating a DataKeeper resource instance, refer to the following documentation.

Creating a DataKeeper Resource Hierarchy

Extending Your Hierarchy

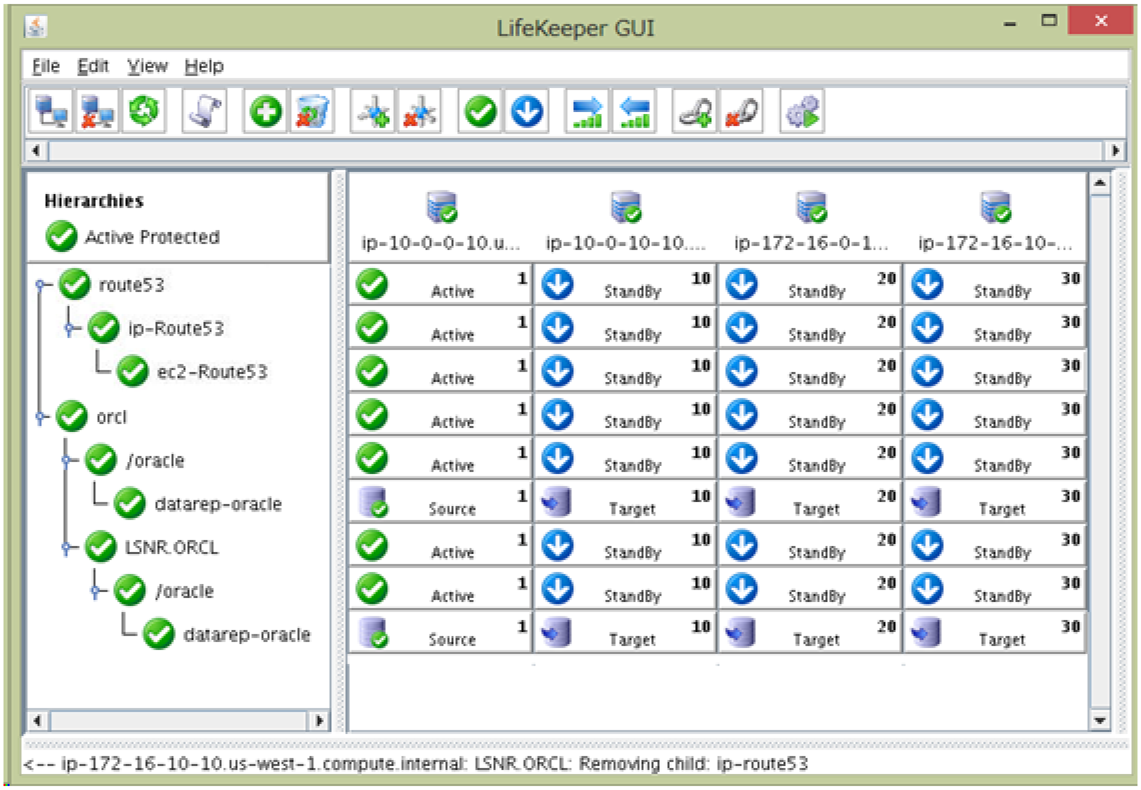
When creating a DataKeeper resource instance, choose **Replicate Existing File System** for **Hierarchy Type** in the **Create Resource Wizard**. The files related to Oracle have been previously stored on the disk for replication. Mount the replication target disk to the mount point in the primary node and then create a DataKeeper resource.

19. Create an Oracle Resource

Run the LifeKeeper GUI on the primary node and launch the Create Resource Wizard selecting Oracle Database Lister followed by Oracle Database to create the Oracle Database Listener and Oracle Database resource hierarchies. Refer to Creating an Oracle Resource Hierarchy and Extending Your Hierarchy for more information.

Extend to all nodes in the cluster.

When the Oracle Database Listener and Database resources creation completes, the LifeKeeper GUI displays as follows:



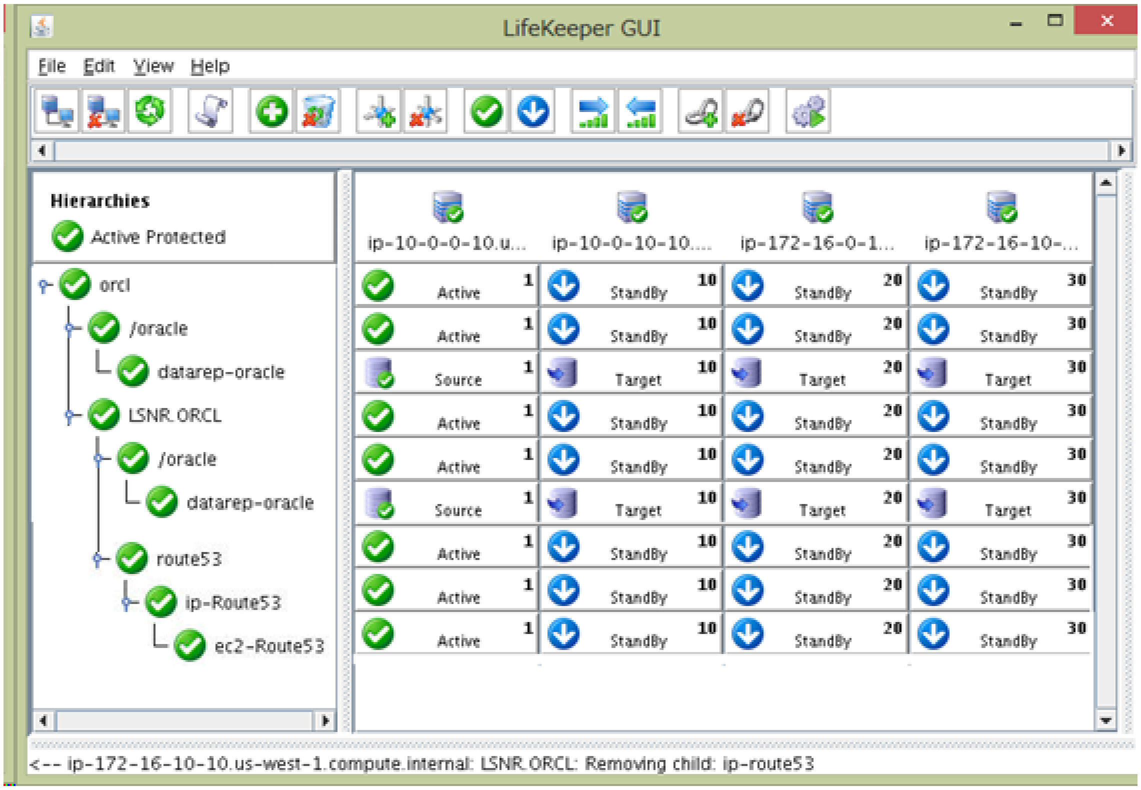
Hierarchies ✓ Active Protected	ip-10-0-0-10.u...	ip-10-0-10-10....	ip-172-16-0-1...	ip-172-16-10-...
✓ route53	✓ Active 1	StandBy 10	StandBy 20	StandBy 30
✓ ip-Route53	✓ Active 1	StandBy 10	StandBy 20	StandBy 30
✓ ec2-Route53	✓ Active 1	StandBy 10	StandBy 20	StandBy 30
✓ orcl	✓ Active 1	StandBy 10	StandBy 20	StandBy 30
✓ /oracle	✓ Active 1	StandBy 10	StandBy 20	StandBy 30
✓ datarep-oracle	Source 1	Target 10	Target 20	Target 30
✓ LSNR.ORCL	✓ Active 1	StandBy 10	StandBy 20	StandBy 30
✓ /oracle	✓ Active 1	StandBy 10	StandBy 20	StandBy 30
✓ datarep-oracle	Source 1	Target 10	Target 20	Target 30

<-- ip-172-16-10-10.us-west-1.compute.internal: LSNR.ORCL: Removing child: ip-route53

Figure 6 Example of an Oracle Resource Hierarchy

20. Create a dependency between the Route53 resource and the Oracle resource

Manually create a dependency between the Oracle Database Resource Hierarchy and the Route53 Resource Hierarchy to ensure the Route53 resource starts before the Oracle Database resource (create with Oracle as the parent and Route53 as the child).



Hierarchies	ip-10-0-0-10.u...	ip-10-0-10-10....	ip-172-16-0-1...	ip-172-16-10-...
orcl	Active 1	StandBy 10	StandBy 20	StandBy 30
/oracle	Active 1	StandBy 10	StandBy 20	StandBy 30
datarep-oracle	Source 1	Target 10	Target 20	Target 30
LSNR.ORCL	Active 1	StandBy 10	StandBy 20	StandBy 30
/oracle	Active 1	StandBy 10	StandBy 20	StandBy 30
datarep-oracle	Source 1	Target 10	Target 20	Target 30
route53	Active 1	StandBy 10	StandBy 20	StandBy 30
ip-Route53	Active 1	StandBy 10	StandBy 20	StandBy 30
ec2-Route53	Active 1	StandBy 10	StandBy 20	StandBy 30

```
<-- ip-172-16-10-10.us-west-1.compute.internal: LSNR.ORCL: Removing child: ip-route53
```

Figure 7 Example of a dependency between a Route53 Resource Hierarchy and an Oracle Resource Hierarchy

A Route53 resource hierarchy is specified as a child resource of an Oracle Database resource instance (orcl) in this example.

A dependency is specified to ensure the Route53 resource starts before Oracle.

Configuring of resource hierarchies is now complete. Verify each service after all mirrors are in sync by switching over to the standby nodes.

Chapter 5: Related LifeKeeper Resources

As mentioned in the configuration overview, there are 4 LifeKeeper resources used in this configuration.

Each function and operation overview are as follows.

Openswan Resource

Overview

The LifeKeeper Openswan resource monitors the VPN connections that are required for connecting two Regions.

The resource instance monitors the connectivity of both the Active and Standby VPN tunnels. When the resource instance connects to another Region, the connection is established through a VPN tunnel of a VPN instance whose Openswan resource status is Active.

The Route Table determines the communication route. The communication route must be changed to one through a Standby node when the Openswan resource status changes. Then EC2 resource which is a dependent child to the Openswan resource changes the Route Table when the Openswan status changes. Refer to EC2 Resource for more information.

Monitoring and Recovery of Openswan Resource

The Openswan resource, using the Openswan software installed on the system, can protect one IPsec-VPN session that has been configured between opposing regions (note: the VPN session cannot be configured within the same region). Monitoring of resource processes are performed on both the Active and Standby nodes.

1. Check if the Openswan process exists. If the process is not found, LifeKeeper determines that a resource failure has occurred and initiates a recovery operation. Any other verification items are not performed.

2. Ping the local IP on the opposite VPN server (primary to standby region) via the VPN Tunnel and check for an acknowledgement. (The maximum response time for a ping response is tunable)
3. Ping the public IP on the opposite VPN server (primary to standby region) and check for an acknowledgement.
4. Check if the VPN Tunnel has been successfully established using the ipsec command.

If any of these checks fail, LifeKeeper initiates a recovery operation.

If LifeKeeper determines that a failure occurred on the active node, the standby node, or both, then LifeKeeper initiates a local recovery. The conditions for a failover are based on the results of the recovery and occur as follows.

- A failover will not be performed if the local recovery is successful on both the active and standby nodes.
- A failover will not be performed if the local recovery is successful on active node but fails on the standby node.
- A failover will be performed if the local recovery fails on the active node but is successful on the standby node.
- A failover will not be performed if the local recovery fails on both the active and standby nodes. In this case a successful recovery status will be returned and monitoring and recovery attempts of the VPN Tunnel will continue as long as it registers as disconnected.

With most all LifeKeeper resources, a failover of the resource is performed to a standby node if the local recovery fails. In the case of an Openswan resource, a communication path is protected by a VPN tunnel on both the active node and the standby node. A failover is performed only when the communication path can be kept active by failing over.

Openswan Resource Tunable Items

The following parameters can be added to the `/etc/default/LifeKeeper` file on all nodes in the cluster to change default behavior. The value becomes valid immediately when the changes are saved. It is not necessary to restart Lifekeeper or the OS.

Parameter name	Defalt value	Description
OPENSWAN_REMOVE_TIMEOUT	15 (seconds)	Time-out period for the remove operation
OPENSWAN_PING_TIMEOUT	5 (seconds)	Time-out period for the local IP ping test during resource monitoring.
OPENSWAN_CHECK_TRY_COUNT	3 (times)	The maximum number of retry attempts to confirm the status of the VPN Tunnel during resource monitoring.

Parameter name	Defalt value	Description
OPENSWAN_CHECK_INTERVAL	15 (seconds)	Interval in seconds for confirmation of the VPN Tunnel status during resource monitoring.

Route53 Resource

Overview

To ensure continued access to the protected services after a region switchover requires an update to the Amazon Route 53 DNS information correspond to the virtual IP address for the IP resource as defined in the new region. This function is provided by the Route53 resource. When the switchover occurs the Route53 resource will create a DNS A record corresponding to the virtual IP address in the new region and signal the Amazon Route 53 services to perform an update.

Monitoring and Recovery of Route53 Resources

The Route53 resource monitors virtual IP addresses using Amazon Route 53 DNS A Record information as follows (note: DNS A record information is registered when the Route53 resource is brought in service at create time or when a switchover occurs):

1. In order to monitor Amazon Route 53 DNS information the Route53 resource must be able to obtain the DNS A record information (see point 2 for the how A record information is used). If the Route53 resource fails to obtain the data using Amazon Route 53 APIs it will attempt 2 additional times (for a total of 3 attempts by default) waiting 2 seconds between attempts (by default). After the third unsuccessful attempt it will stop and record the failure in the log. No local recovery or failover will occur.

2. Ensure that the Amazon Route 53 DNS information is correct pertaining to the IP address information. This is done by first retrieving the Amazon Route 53 DNS A record information associated with the protect Route53 resource. Using this information the IP address in the DNS A record information is compared to the virtual IP address from the dependent IP resource. If the IP addresses do not match or the A record does not exist then exit with a failure to initiate a local recovery. If the IP address information matches, then exit with a success as no errors exist.

Route53 Resource Tunable Items

The following parameters can be added to the `/etc/default/LifeKeeper` file on all nodes in the cluster to change default behavior. The value becomes valid immediately when the changes are saved. It is not necessary to restart Lifekeeper or the OS.

Parameter name	Default value	Description
ROUTE53_TTL	10 (seconds)	The default setting value for TTL(Time To Live) of the A record created for the Route53 resource. *Switchover is required to reflect the setting.
ROUTE53_RECORD_INTERVAL	2 (seconds)	The amount of time in seconds between Route 53 DNS A record update attempts.
ROUTE53_RECORD_TRY_COUNT	3 (times)	The number of retry attempts to successfully update the Route 53 DNS A record before failing the update attempt.
ROUTE53_CHANGEID_INTERVAL	20 (seconds)	The amount of time in seconds between batch change requests via Route 53 APIs.
ROUTE53_CHANGEID_TRY_COUNT	5 (times)	The number of retry attempts for batch change requests using the Route 53 APIs.
ROUTE53_RECORDCHECK_INTERVAL	2 (seconds)	The amount of time in seconds between attempts to retrieve Route 53 A record information via Route 53 APIs.
ROUTE53_RECORDCHECK_TRY_COUNT	4 (times)	The number of retry attempts for Route 53 A record information requests via the Route 53 APIs.

EC2 Resource

Overview

An EC2 resource is created as a child resource for both Openswan and Route53 resource instances.

The EC2 resource is responsible for updating the routing information required to keep the communication paths functional when the resource hierarchy is switched to other AZs and Regions. This basic operation is the same as the "Route table scenario" of the "Recovery Kit for EC2". Refer to the Recovery Kit for EC2 Administration Guide for more information.

Monitoring and Recovery of EC2 Resources

The EC2 resource verifies that the target of the protected IP routing in Route Table is correctly set to the ENI on the active server using Amazon EC2 API Tools. Otherwise, an EC2 local recovery process is initiated.

Updating a Route Table when switching occurs on EC2 Resource

An EC2 resource created with an Openswan resource updates the associated ENI, which is an outlet for communication with another Region. An EC2 resource created with a Route53 resource updates the associated ENI when a virtual IP address is switched between AZs.

The following shows examples of the Route table transition when switchover occurs.

1. In this example a Route53 resource is Active on LK_P_A (service instance) and an Openswan resource is Active on VPN_P_A (VPN instance). The state of the Route Table is as follows. (**Note:** Settings of the VPC and IGW are not listed since they are specified by default and are not controlled objects).

Destination	Target	Description
10.1.0.10/32	ENI on LK_P_A	Route53 resource is Active on LK_P_A
10.2.0.0/16	ENI on VPN_P_A	Openswan resource is Active on VPN_P_A
172.17.0.0/24	ENI on VPN_P_A (same as above)	Openswan resource is Active on VPN_P_A

2. When the Openswan resource is switched from VPN_P_A to VPN_P_B, the Route Table appears as follows.

Destination	Target	Description
10.1.0.10/32	ENI on LK_P_A	No change from 1 above
10.2.0.0/16	ENI on VPN_P_B	Information is updated for the ENI on VPN_P_B which is the target of the switchover.
172.17.0.0/24	ENI on VPN_P_B	Information is updated for the ENI on VPN_P_B which is the target of the switchover.

3. When the Route53 resource is switched from LK_P_A to LK_P_B, the Route Table appears as follows.

Destination	Target	Description
10.1.0.10/32	ENI on LK_P_B	Information is updated for the ENI associated with the virtual IP on LK_P_B which is the target of the switchover.
10.2.0.0/16	ENI on VPN_P_B	No change from 2 above
172.17.0.0/24	ENI on VPN_P_B	No change from 2 above

IP resource

Overview

An IP resource is a virtual IP address created using the IP Recovery Kit which is included with the LifeKeeper Core product. As with all LifeKeeper resources, the IP resource instance is switchable between all nodes in the cluster. When IP resource instances are used in EC2 Cross Region configurations the virtual IP address specified during create will not be valid when the resource is switched to nodes in other Regions. Therefore in these configurations when the resource extension occurs to a node in another region, LifeKeeper will provide the opportunity to specify a virtual IP address appropriate for that network segment.

Refer to IP Recovery Kit Administration Guide for more information.

Chapter 6: Configration Considerations

Quorum/Witness Server

In an environment using EC2 Cross Region configurations that are configured across AZs, the HA cluster has a risk of communication path failures and therefore split-brain scenarios. This risk goes goes up when the configuration is setup across Regions. Therefore, the use of Quorum/Witness Servers for I/O fencing is highly recommended. If you use the TCP_REMOTE setting for the Quorum mode, it is easy to use in a cloud environment as the I/O fencing function can be implemented without a separate deployment of a Quorum server. Refer to Quorum/Witness for more information.

Updating Route53 Resource Records

It may take several minutes to update the Route53 resource records when starting a Route53 resource. Amazon provides the following information for propagation velocity for updating Route 53 DNS records.

> Amazon Route 53 FAQ
>
> Q: How quickly will changes I make to my DNS settings on Amazon Route 53 propagate globally?
>
> https://aws.amazon.com/route53/faqs/

The Route53 resource confirms status information when updating DNS records using the Amazon Route 53 APIs. An INSYNC status indicates the update completed and the LifeKeeper operation will complete successfully. If a PENDING status is returned, then the update is still in progress and the LifeKeeper operation must retry the update check. Because of how this process is designed in Route 53, an update request that takes a long time to propagate the information will cause a LifeKeeper in service request to fail when the LifeKeeper retry attempts have been reached. This will be the case even though the Route 53 update was initiated correctly.

See the Route 53 Management Console to confirm that the A record has been properly updated if the in service request for the Route53 resource fails. If the Route 53 Management Console shows the A record has been properly updated, then the DNS service update has completed and only LifeKeeper needs to be updated to reflect this. In this case re-issue the in service request for the Route53 resource via the LifeKeeper GUI.

If the in service request for the Route53 resource always fails, increment the ROUTE53_CHANGEID_TRY_COUNT tunable in the /etc/default/LifeKeeper file by 1 or 2 (if the value does not exist in the file add it with a setting of 5 or 6, otherwise increment the existing value by 1 or 2) and see if the new value has an impact on in service requests. Restarting of LifeKeeper and/or the OS is not required for this modification.

Chapter 7: Known Issues and Troubleshooting

There are no known issues at this time.