



APPENDIX **B**

ハイ アベイラビリティおよび障害回復保証 テスト プラン

HA/DR 保証テスト プランでは、Security Manager アプリケーションの可用性が高く、さまざまなハードウェア障害とソフトウェア障害に対応できることを検証します。テスト プランは、サーバ間のアプリケーションを手動で切り替えるなどのメンテナンス アクティビティでも有効です。



(注)

Security Manager クライアント セッションでは、アプリケーションのフェールオーバー後にアクティブ ユーザが再びログインする必要があります。この動作は、サーバで実行される Security Manager サービスを停止して開始することと同じことです。

次の事例カテゴリはこのマニュアルの付録に記載されています。

- 「手動切り替え」(P.B-1)
- 「イーサネット/ネットワーク障害」(P.B-3)
- 「サーバ障害」(P.B-10)
- 「アプリケーション障害」(P.B-16)

手動切り替え

ここでは 2 つの異なるタイプの手動切り替えを説明しています。2 台のサーバを備えた単一クラスタでは、クラスタ内の 2 台のサーバ間での切り替え (クラスタ内切り替え) ができます。個々のクラスタに 1 台のサーバを備えたデュアル クラスタ構成では、クラスタ間での切り替え (クラスタ間切り替え) ができます。

ここでは、次の内容について説明します。

- 「クラスタ内切り替え」(P.B-1)
- 「クラスタ間切り替え」(P.B-2)

クラスタ内切り替え

事例タイトル: クラスタ内でのアプリケーションの手動切り替え

説明: アプリケーションを、VCS を使用する同じクラスタ内の別のサーバへ手動で切り替えます。

テスト セットアップ: 単一クラスタ コンフィギュレーションでのデュアル ノード クラスタ (図 1-1 (P.12))。

- ステップ 1** APP サービス グループがプライマリ サーバで実行されているかどうか確認します。VCS Cluster Explorer で、[APP] サービス グループを選択します。ショートカットメニューから、[Switch To] を選択し、セカンダリ サーバを選択します。あるいは、次のコマンドを実行します。

```
C:¥> hagrps -switch APP -to secondary_server_name
```

- ステップ 2** APP サービス グループのリソース ビューで、このサービス グループにあるリソースがプライマリ サーバでオフラインになり、次にセカンダリ サーバでオンラインになるのを確認します。または次のコマンドを実行して、APP サービス グループの状況を観察します。

```
C:¥> hagrps -state APP
```

- ステップ 3** クライアント マシンで、[login] ダイアログボックスの [Server Name] フィールドで、仮想ホスト名または IP アドレスを使用して Security Manager クライアントを起動します。アプリケーションに正常にログインできることを確認します。

クラスタ間切り替え

事例タイトル : クラスタ間の手動によるアプリケーション切り替え

説明 : アプリケーションを、VCS を使用する別のクラスタ内のサーバへ手動で切り替えます。

テスト セットアップ : 各クラスタに単一サーバを備えた [図 1-2 \(P.14\)](#) で示すデュアル クラスタ コンフィギュレーション。

- ステップ 1** VCS Cluster Explorer で、[APP] サービス グループを選択します。ショートカットメニューで [Switch To] を選択してから [Remote Switch(...)] を選択し、[Switch global] ダイアログボックスを開きます。このダイアログボックスで、リモート クラスタを指定し、必要に応じてリモート クラスタでの特定のサーバを指定します。あるいは、次のコマンドを実行します。

```
C:¥> hagrps -switch APP -any -clus secondary_cluster_name
```

- ステップ 2** APP サービス グループのリソース ビューで、このサービス グループにあるリソースがプライマリ クラスタでオフラインになるのを確認します。ツリー内のクラスタ ノードを選択し、[Remote Cluster Status] ビューを使用して APP サービス グループがリモート クラスタでオンラインになることを確認します。または次のコマンドを実行して、APP サービス グループの状況を観察します。

```
C:¥> hagrps -state APP
```

#Group	Attribute	System	Value
APP	State	csm_primary:<Primary Server>	OFFLINE
APP	State	localclus:<Secondary Server>	ONLINE

- ステップ 3** クライアント マシンで、[Login] ダイアログボックスの [Server Name] フィールドで、セカンダリ クラスタで使用されている適切なホスト名またはアプリケーションの IP アドレスを入力して Security Manager クライアントを起動します。アプリケーションに正常にログインできることを確認します。

- ステップ 4** Security Manager からログアウトし、VCS Cluster Explorer または次のコマンドを使用して、APP サービス グループをプライマリ クラスタに切り替えます。

```
C:¥> hagrps -switch APP -any -clus primary_cluster_name
```

イーサネット/ネットワーク障害

HA/DR コンフィギュレーションには、2つのタイプのサーバイーサネット接続があります。最初のタイプは、ネットワーク通信に使用されるイーサネット接続（パブリック インターフェイス）で、2番目のタイプは、クラスタ内通信専用として使用されるイーサネット インターフェイス（専用インターフェイス）です。ここでは、イーサネット インターフェイスの個々のタイプに関する障害事例を説明します。

- 「ネットワーク通信障害」(P.B-3)
- 「クラスタの通信障害」(P.B-8)

ネットワーク通信障害

ここでは、VCS がネットワーク通信に使用されるネットワーク イーサネット ポートの障害を検出できるかどうかの検証に使用されるテストについて説明します。ここでは、次の内容について説明します。

- 「セカンダリ サーバ、単一クラスタでのネットワーク イーサネット障害」(P.B-3)
- 「プライマリ サーバ、単一クラスタでのネットワーク イーサネット障害」(P.B-4)
- 「セカンダリ サーバ、デュアル クラスタでのネットワーク イーサネット障害」(P.B-5)
- 「プライマリ サーバ、デュアル クラスタでのネットワーク イーサネット障害」(P.B-7)

セカンダリ サーバ、単一クラスタでのネットワーク イーサネット障害

事例タイトル: 単一クラスタ コンフィギュレーションでのセカンダリ サーバ上のネットワーク イーサネット接続で障害が発生します。

説明: この事例は、VCS がセカンダリ サーバ上のネットワーク イーサネット ポートでの障害を検出し、次にこの障害が復旧してから回復できることを検証します。

テスト セットアップ: サーバ単位で単一ネットワーク接続を備えた単一クラスタ コンフィギュレーションでのデュアル ノードクラスタ (図 1-1 (P.12))。

ステップ 1 アプリケーションがプライマリ サーバで実行されていることを確認します。

ステップ 2 クライアント マシンからアプリケーションにログインします。

ステップ 3 セカンダリ サーバ上のネットワーク ポートからイーサネット ケーブルを取り外し、スイッチ ネットワーク/ルータ ネットワークとの通信からサーバを切断します。VCS がネットワーク ポート障害を検出するまで 60 秒以上待ちます。次のコマンドを実行して、VCS がセカンダリ サーバ上で NIC リソースの障害を検出することを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System                State                Frozen
A <PrimaryServer>        RUNNING             0
A <SecondaryServer>      RUNNING             0

-- GROUP STATE
-- Group                System                Probed    AutoDisabled    State
B APP                   <PrimaryServer>       Y         N                ONLINE
B APP                   <SecondaryServer>     Y         N                OFFLINE | FAULTED

-- RESOURCES FAILED
-- Group                Type                Resource                System
C APP                   NIC                 NIC                     <SecondaryServer>
```

- ステップ 4** セカンダリ サーバ上のネットワーク ポートでイーサネット ケーブルを元に戻します。次のコマンドを実行して、障害の解消を VCS が検出することを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  RUNNING      0
A <SecondaryServer> RUNNING      0

-- GROUP STATE
-- Group          System          Probed      AutoDisabled  State
B APP            <PrimaryServer> Y           N             ONLINE
B APP            <SecondaryServer> Y           N             OFFLINE
```

プライマリ サーバ、単一クラスタでのネットワーク イーサネット障害

事例タイトル：単一クラスタ コンフィギュレーションでのプライマリ サーバ上のネットワーク イーサネット接続で障害が発生します。

説明：この事例は、VCS がプライマリ サーバのネットワーク イーサネット ポートでの障害を検出し、自動的にアプリケーションをセカンダリ サーバへ切り替えることを検証します。問題が修正されると、アプリケーションを手動でプライマリ サーバに戻すことができます。

テスト セットアップ：各サーバに単一ネットワーク接続を備えたデュアル ノードクラスタ (図 2-2 (P.23))。

- ステップ 1** アプリケーションがプライマリ サーバで実行されていることを確認します。

- ステップ 2** プライマリ サーバ上のネットワーク ポートからイーサネット ケーブルを取り外し、スイッチ ネットワーク/ルータ ネットワークとの通信からサーバを切断します。VCS が NIC リソースの障害を検出し、APP サービス グループをセカンダリ サーバに自動的に切り替えることを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  RUNNING      0
A <SecondaryServer> RUNNING      0

-- GROUP STATE
-- Group          System          Probed      AutoDisabled  State
B APP            <PrimaryServer> Y           N             OFFLINE | FAULTED
B APP            <SecondaryServer> Y           N             ONLINE

-- RESOURCES FAILED
-- Group          Type          Resource          System
C APP            NIC          NIC              <PrimaryServer>
C APP            IP          APP_IP          <PrimaryServer>
```

- ステップ 3** アプリケーションがセカンダリ サーバ上で実行されている間に、このアプリケーションにログインできることを確認します。

- ステップ 4** プライマリ サーバのネットワーク ポートにイーサネット ケーブルを取り付け、プライマリ サーバ上の障害が発生している IP リソースを手動で解消します。

```
C:¥> hares -clear APP_IP -sys primary_server_name
```

ステップ 5 APP サービス グループを手動でプライマリ サーバに戻します。

```
C:¥> hagrps -switch APP -to primary_server_name
```

セカンダリ サーバ、デュアル クラスタでのネットワーク イーサネット障害

事例タイトル: デュアル クラスタ コンフィギュレーションでのセカンダリ サーバ上のネットワーク イーサネット接続で障害が発生します。

説明: この事例は、VCS がネットワーク イーサネット ポート上で障害を検出し、次にこの障害が復旧してから回復できることを検証します。

テスト セットアップ: 各クラスタ内に単一ノードを備え、各サーバに単一イーサネット ネットワーク 接続を備えたデュアル クラスタ コンフィギュレーション (図 1-2 (P.14))。

ステップ 1 APP サービス グループがプライマリ クラスタ/プライマリ サーバで実行されているかどうか確認します。

ステップ 2 クライアント マシンから Security Manager にログインします。

ステップ 3 セカンダリ クラスタで、サーバ上のネットワーク ポートからイーサネット ケーブルを取り外します。これにより、スイッチ ネットワーク/ルータ ネットワークとの通信からサーバが切り離され、レプリケーションが中断します。次のコマンドを実行して、プライマリ サーバからレプリケーションが中断 (切断) されたことを確認します。

```
C:¥> vxprint -P1
Diskgroup = datadg

Rlink      : rlk_172_6037
info       : timeout=500 packet_size=1400
            latency_high_mark=10000 latency_low_mark=9950
            bandwidth_limit=none
state      : state=ACTIVE
            synchronous=off latencyprot=off srlprot=off
assoc      : rvg=CSM_RVG
            remote_host=172.25.84.34
            remote_dg=datadg
            remote_rlink=rlk_172_32481
            local_host=172.25.84.33
protocol   : UDP/IP
flags      : write attached consistent disconnected
```

ステップ 4 プライマリ サーバから次のコマンドを実行して、セカンダリ クラスタとの通信が失われたことを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  RUNNING      0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B APP              <PrimaryServer> Y         N             ONLINE
B APPrep           <PrimaryServer> Y         N             ONLINE
B ClusterService  <PrimaryServer> Y         N             ONLINE

-- WAN HEARTBEAT STATE
-- Heartbeat      To              State
```

```

L Icmp                csm_secondary      ALIVE

-- REMOTE CLUSTER STATE
-- Cluster            State
M csm_secondary      LOST_CONN

-- REMOTE SYSTEM STATE
-- cluster:system     State              Frozen
N csm_secondary:<SecondaryServer>  RUNNING          0

-- REMOTE GROUP STATE
-- Group              cluster:system     Probed      AutoDisabled  State
O APP csm_secondary:<SecondaryServer>  Y           N            OFFLINE

```

ステップ 5 ネットワーク イーサネット ケーブルをセカンダリ サーバに取り付け直し、レプリケーションが再開されたことを確認します。

```

C:¥> vxprint -Pl

Diskgroup = datadg

Rlink      : rlk_172_6037
info       : timeout=29 packet_size=1400
            latency_high_mark=10000 latency_low_mark=9950
            bandwidth_limit=none
state      : state=ACTIVE
            synchronous=off latencyprot=off srlprot=off
assoc      : rvg=CSM_RVG
            remote_host=172.25.84.34
            remote_dg=datadg
            remote_rlink=rlk_172_32481
            local_host=172.25.84.33
protocol   : UDP/IP
flags      : write attached consistent connected

```

ステップ 6 セカンダリ クラスタへの通信が復元されていることを確認します。

```

C:¥> hastatus -sum
-- SYSTEM STATE
-- System            State              Frozen
A <PrimaryServer>   RUNNING          0
-- GROUP STATE
-- Group             System              Probed      AutoDisabled  State
B APP                <PrimaryServer>  Y           N            ONLINE
B APPrep             <PrimaryServer>  Y           N            ONLINE
B ClusterService    <PrimaryServer>  Y           N            ONLINE

-- WAN HEARTBEAT STATE
-- Heartbeat         To              State
L Icmp              csm_secondary  ALIVE

-- REMOTE CLUSTER STATE
-- Cluster           State
M csm_secondary     RUNNING

-- REMOTE SYSTEM STATE
-- cluster:system    State              Frozen
N csm_secondary:<SecondaryServer>  RUNNING          0

-- REMOTE GROUP STATE
-- Group             cluster:system     Probed      AutoDisabled  State
O APP csm_secondary:<SecondaryServer>  Y           N            OFFLINE

```

- ステップ 7** レプリケーションが回復されていない場合は、IP リソースに障害がある場合はこれを手動で解消し、次のようにセカンダリ側で APPrep サービス グループを開始します。

```
C:¥> hares -clear APP_IP
C:¥> hagrp -online APPrep -sys secondary_server_name
```

プライマリ サーバ、デュアル クラスタでのネットワーク イーサネット障害

事例タイトル: プライマリ サーバ上のネットワーク イーサネット接続で障害が発生します。

説明: この事例は、VCS がプライマリ サーバのネットワーク イーサネット ポートでの障害を検出し、セカンダリ サーバでアプリケーションを開始することにより回復できることを検証します。イーサネット接続の復元後、手動で元のプライマリ サーバにフェールオーバーできます。セカンダリ サーバでの実行中に行われたデータの変更はすべて保持されます。

テスト セットアップ: 各サーバに単一ノードを備えたデュアル クラスタ コンフィギュレーション (図 1-2 (P.14))。

- ステップ 1** APP サービス グループがプライマリ クラスタで実行されているかどうか確認します。

- ステップ 2** プライマリ クラスタ内のサーバ上のポートからネットワーク イーサネット ケーブルを取り外し、スイッチ ネットワーク / ルータ ネットワークとの通信からサーバを切断します。VCS は、これを IP リソースおよび NIC リソースの障害として検出するはずですが、VCS が障害を検出し、APP サービス グループを停止したことを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  RUNNING      0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B APP              <PrimaryServer> Y         N             OFFLINE
B APPrep          <PrimaryServer> Y         N             OFFLINE | FAULTED
B ClusterService <PrimaryServer> Y         N             ONLINE

-- RESOURCES FAILED
-- Group           Type           Resource    System
C APPrep          IP             APP_IP      <PrimaryServer>
C APPrep          NIC            NIC         <PrimaryServer>

-- WAN HEARTBEAT STATE
-- Heartbeat       To             State
L Icmp            csm_secondary DOWN

-- REMOTE CLUSTER STATE
-- Cluster         State
M csm_secondary  FAULTED

-- REMOTE SYSTEM STATE
-- cluster:system  State          Frozen
N csm_secondary:<SecondaryServer> FAULTED 0

-- REMOTE GROUP STATE
-- Group           cluster:system Probed    AutoDisabled  State
O APP             csm_secondary:<SecondaryServer> Y         N             OFFLINE
```

- ステップ 3** セカンダリ サーバで次のコマンドを使用して、セカンダリ クラスタで APP サービス グループを開始します。
- ```
C:¥> hagr -online -force APP -sys secondary_server_name
```
- ステップ 4** クライアント マシンから Security Manager にログインして、稼動中であることを確認します。一部のデータを変更し、プライマリ サーバに戻しても変更が保持されていることを確認します。
- ステップ 5** プライマリ クラスタ サーバにネットワーク イーサネット ケーブルを再接続します。
- ステップ 6** IP リソースに障害があればこれを解消し、プライマリ サーバから APPrep サービスを有効にします。
- ```
C:¥> hares -clear APP_IP
C:¥> hagr -online APPrep -sys primary_server_name
```
- ステップ 7** 元のプライマリ RVG をセカンダリに変換し、高速フェールバック機能を使用して、元のプライマリ RVG 内のデータ ボリュームと新しいプライマリ RVG 内のデータ ボリュームを同期します。セカンダリ クラスタの Cluster Explorer を使用して、RVGPrimary リソース ([APP_RVGPrimary]) を右クリックし、[actions] を選択し、[Actions] ダイアログボックスから [fbsync] を選択して、次に [OK] をクリックします。あるいは、次のコマンドを実行することもできます。
- ```
C:¥> hares -action APP_RVGPrimary fbsync 0 -sys secondary_server_name
```
- ステップ 8** セカンダリ クラスタで VCS Cluster Explorer を使用して、[APP] サービス グループを選択します。ショートカット メニューで [Switch To] を選択してから [Remote Switch(...)] を選択し、[Switch global] ダイアログボックスを開きます。このダイアログボックスで、プライマリ クラスタとプライマリ サーバを指定します。あるいは、次のコマンドを実行します。
- ```
C:¥> hagr -switch APP -any -clus primarycluster
```
- ステップ 9** アプリケーションにログインして、セカンダリ サーバで行った変更が保持されたことを確認します。

クラスタの通信障害

事例タイトル: クラスタ通信に使用するイーサネットで障害が発生します。

説明: クラスタ内通信のクラスタ内のサーバ間で使用される専用イーサネット接続で障害が発生します。このテストでは、3 つの冗長通信パスのうち 2 つまでが失われた場合でも、クラスタ通信が機能し続けることを検証します。

テスト セットアップ: 単一クラスタ コンフィギュレーションにおいて、2 つの専用クラスタ通信イーサネット接続と、ネットワーク イーサネット接続に設定されるプライオリティの低いクラスタ通信接続を備えたデュアル ノード クラスタ (図 1-1 (P.12))。



- (注) この事例で指定されたコマンドに加えて、ツリーのルート ノードと [System Connectivity] タブを選択することで、Cluster Explorer からクラスタ通信の状況を監視できます。

- ステップ 1** 次のコマンドを実行して、すべてのシステムが GAB を介して通信していることを確認します。



- (注) Group Membership Services/Atomic Broadcast (GAB) は、クラスタ メンバーシップおよびクラスタ通信を管理する VCS プロトコルです。


```
# gabconfig -a
GAB Port Memberships
=====
Port a gen   e8cc02 membership 01
Port h gen   e8cc01 membership 01
```

ステップ 2 プライマリ サーバ上のクラスタ通信で使用される最初の専用イーサネット ポートから、イーサネット ケーブルを取り外します。

ステップ 3 次のコマンドを実行して、クラスタ通信で使用されるリンクの詳細状況を表示し、最初の専用クラスタ通信ポートがダウンしていることを確認します。



(注) 出力中のアスタリスク (*) は、コマンドが実行されるサーバを示します。コマンドが実行されるサーバでは、ポートのうち 1 つ以上が物理的に切断されていても、常にサーバのリンクが機能していると表示されます。

```
# lltstat -nvv
LLT node information:
  Node          State   Link  Status  Address
  * 0 <PrimaryServer> OPEN
                                Adapter0  UP      00:14:5E:28:52:9C
                                Adapter1  UP      00:14:5E:28:52:9D
                                Adapter2  UP      00:0E:0C:9C:20:FE
  1 <SecondaryServer> OPEN
                                Adapter0  DOWN
                                Adapter1  UP      00:14:5E:28:27:17
                                Adapter2  UP      00:0E:0C:9C:21:C2
  ...
```

ステップ 4 優先順位の低いハートビート リンクをネットワーク インターフェイス上に設定した場合、プライマリサーバ上のクラスタ通信で使用される 2 番目の専用イーサネット ポートからイーサネット ケーブルを取り外します。

ステップ 5 次のコマンドを実行して、すべてのシステムが GAB を介して通信していることを確認します。各サーバではハートビートが 1 つしか動作していないため、クラスタ内の両方のサーバが **Jeopardy** 状態となったことも確認します。

```
# gabconfig -a
GAB Port Memberships
=====
Port a gen   e8cc02 membership 01
Port a gen   e8cc02  jeopardy ;1
Port h gen   e8cc01 membership 01
Port h gen   e8cc01  jeopardy ;1
```

ステップ 6 次のコマンドを実行して、クラスタ通信に使用されるリンクの詳細状況を表示し、プライマリ サーバ上にあるクラスタ通信用の 2 番目の専用イーサネット ポートがダウンしていることを確認します。

```
# lltstat -nvv
LLT node information:
  Node          State   Link  Status  Address
  * 0 <PrimaryServer> OPEN
                                Adapter0  UP      00:14:5E:28:52:9C
                                Adapter1  UP      00:14:5E:28:52:9D
                                Adapter2  UP      00:0E:0C:9C:20:FE
  1 <SecondaryServer> OPEN
                                Adapter0  DOWN
                                Adapter1  UP      00:14:5E:28:27:17
                                Adapter2  DOWN
```

ステップ 7 プライマリ サーバ上にあるクラスタ通信用の 2 番目の専用イーサネット ポートでイーサネット ケーブルを交換します。

ステップ 8 次のコマンドを実行して、Jeopardy 状態が解消したことを確認します。

```
# gabconfig -a
GAB Port Memberships
=====
Port a gen   e8cc02 membership 01
Port h gen   e8cc01 membership 01
```

ステップ 9 プライマリ サーバ上にあるクラスタ通信用の最初の専用イーサネット ポートに、イーサネット ケーブルを再び取り付けます。

サーバ障害

ここでは、障害を発生させるためにサーバの電源を切断することによりサーバ障害が発生するケースについて説明します。次の 4 つのケースについて説明します。

- 「スタンバイ サーバの障害、単一クラスタ」(P.B-10)
- 「プライマリ サーバの障害、単一クラスタ」(P.B-11)
- 「スタンバイ サーバの障害、デュアル クラスタ」(P.B-12)
- 「プライマリ サーバの障害、デュアル クラスタ」(P.B-14)

スタンバイ サーバの障害、単一クラスタ

事例タイトル: 単一クラスタ コンフィギュレーション内のスタンバイ サーバで障害が発生します。

説明: この事例は、プライマリ サーバで実行中のアプリケーションは影響されず、スタンバイ サーバが復旧した後に、アプリケーションがクラスタ コンフィギュレーションに正常に復帰できることを確認します。

テスト セットアップ: 2 つの専用クラスタ通信イーサネット接続と、ネットワーク イーサネット接続でのプライオリティの低いクラスタ通信接続を備えたデュアル ノード クラスタ (図 2-2 (P.23))。

ステップ 1 アプリケーションがクラスタ内のプライマリ サーバで実行されていることを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System                State                Frozen
A <PrimaryServer>        RUNNING              0
A <SecondaryServer>      RUNNING              0

-- GROUP STATE
-- Group                 System                Probed    AutoDisabled    State
B APP                    <PrimaryServer>      Y         N                ONLINE
B APP                    <SecondaryServer>    Y         N                OFFLINE
```

- ステップ 2** セカンダリ サーバの電源を切り、VCS が障害を検出し、アプリケーションがプライマリ サーバで動作し続けることを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  RUNNING      0
A <SecondaryServer> FAULTED     0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B APP             <PrimaryServer> Y         N              ONLINE
```

- ステップ 3** 電源を再び入れて、セカンダリ サーバを起動します。サーバの回復後、次のコマンドを実行して、サーバがクラスタに正常な状態で復帰したことを確認します。出力は、ステップ 1 の出力と同じになるはずです。

```
C:¥> hastatus -sum
```

プライマリ サーバの障害、単一クラスタ

事例タイトル: 単一クラスタ コンフィギュレーション内のプライマリ サーバで障害が発生します。

説明: この事例は、プライマリ サーバで障害が発生した場合、アプリケーションはセカンダリ サーバでの実行を開始し、プライマリ サーバの復元後、アプリケーションがプライマリ サーバで再確立できることを検証します。

テスト セットアップ: デュアル ノード クラスタ (図 1-1 (P.12))。

- ステップ 1** 次のコマンドの出力を確認することにより、APP サービス グループがクラスタ内のプライマリ サーバで実行中であることを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  RUNNING      0
A <SecondaryServer>  RUNNING      0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B APP             <PrimaryServer> Y         N              ONLINE
B APP             <SecondaryServer> Y         N              OFFLINE
```

- ステップ 2** プライマリ サーバの電源を切り、VCS が障害を検出し、APP サービス グループが自動的にセカンダリ サーバに移行することを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  FAULTED     0
A <SecondaryServer>  RUNNING      0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B APP             <SecondaryServer> Y         N              ONLINE
```

- ステップ 3** クライアント マシンから Security Manager に正常にログインできることを確認します。
- ステップ 4** プライマリ サーバの電源を入れて、このサーバが正常な状態でクラスタに復帰できることを確認します。次のコマンドを実行します。出力は、ステップ 1 の出力と同じになるはずですが。

```
C:¥> hastatus -sum
```

- ステップ 5** APP サービス グループを手動でプライマリ サーバに戻します。

```
C:¥> hagr -switch APP -to primary_server_name
```

スタンバイ サーバの障害、デュアル クラスタ

事例タイトル :デュアル クラスタ コンフィギュレーション内のスタンバイ サーバで障害が発生します。

説明 :この事例は、プライマリ クラスタで実行されるアプリケーションはスタンバイ サーバの障害による影響を受けず、スタンバイ サーバの復旧後、アプリケーションがデュアル クラスタ コンフィギュレーションに正常に復帰できることを確認します。

テスト セットアップ :各サーバに単一ノードを備えたデュアル クラスタ コンフィギュレーション (レプリケーションあり) (図 1-2 (P.14))。

- ステップ 1** 次のコマンドをプライマリ サーバで実行して、APP サービス グループおよび ClusterService サービス グループがプライマリ サーバで実行されていることを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  RUNNING      0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B APP              <PrimaryServer> Y         N             ONLINE
B APPrep          <PrimaryServer> Y         N             ONLINE
B ClusterService <PrimaryServer> Y         N             ONLINE

-- WAN HEARTBEAT STATE
-- Heartbeat      To              State
L Icmp            csm_secondary  ALIVE

-- REMOTE CLUSTER STATE
-- Cluster        State
M csm_secondary  RUNNING

-- REMOTE SYSTEM STATE
-- cluster:system          State          Frozen
N csm_secondary:<SecondaryServer> RUNNING      0

-- REMOTE GROUP STATE
-- Group           cluster:system          Probed    AutoDisabled  State
O APP              csm_secondary:<SecondaryServer> Y         N             OFFLINE
```

- ステップ 2** セカンダリ サーバの電源を切り、プライマリ クラスタが、セカンダリ クラスタへの通信が切断されたことを検出します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  RUNNING      0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B APP              <PrimaryServer> Y         N            ONLINE
B APPrep           <PrimaryServer> Y         N            ONLINE
B ClusterService  <PrimaryServer> Y         N            ONLINE

-- WAN HEARTBEAT STATE
-- Heartbeat       To              State
L Icmp             csm_secondary  ALIVE

-- REMOTE CLUSTER STATE
-- Cluster         State
M csm_secondary   LOST_CONN

-- REMOTE SYSTEM STATE
-- cluster:system  State          Frozen
N csm_secondary:<SecondaryServer> RUNNING      0

-- REMOTE GROUP STATE
-- Group           cluster:system  Probed    AutoDisabled  State
O APP              csm_secondary:<SecondaryServer> Y         N            OFFLINE
```

- ステップ 3** セカンダリ サーバの電源を入れます。サーバの再起動後、次のコマンドを実行して、プライマリ クラスタがセカンダリ クラスタとの通信を再確立したことを確認します。出力は、ステップ 1 の出力と同じになるはずですが。

```
C:¥> hastatus -sum
```

- ステップ 4** 次のコマンドを実行して、レプリケーションが動作中で一貫していることを確認します。

```
C:¥> vxprint -P1
Diskgroup = BasicGroup

Diskgroup = datadg

Rlink      : rlk_172_6037
info       : timeout=16 packet_size=1400
            latency_high_mark=10000 latency_low_mark=9950
            bandwidth_limit=none
state      : state=ACTIVE
            synchronous=off latencyprot=off srlprot=off
assoc      : rvg=CSM_RVG
            remote_host=172.25.84.34
            remote_dg=datadg
            remote_rlink=rlk_172_32481
            local_host=172.25.84.33
protocol   : UDP/IP
flags      : write attached consistent connected
```

プライマリ サーバの障害、デュアル クラスタ

事例タイトル: デュアル クラスタ コンフィギュレーション内のプライマリ サーバで障害が発生します。

説明: この事例は、プライマリ サーバで障害が発生した場合、アプリケーションはセカンダリ サーバでの実行を開始し、プライマリ サーバの復元後、アプリケーションがプライマリ サーバで再確立できることを検証します。

テスト セットアップ: 各サーバに単一ノードを備えたデュアル クラスタ コンフィギュレーション (レプリケーションあり) (図 1-2 (P.14))。

ステップ 1 次のコマンドをセカンダリ サーバで実行して、APP サービス グループおよび ClusterService サービスグループがプライマリ サーバで実行されていることを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System                State                Frozen
A <SecondaryServer>     RUNNING             0

-- GROUP STATE
-- Group                 System              Probed              AutoDisabled        State
B APP                   <SecondaryServer>  Y                  N                   OFFLINE
B APPPrep               <SecondaryServer> Y                  N                   ONLINE
B ClusterService       <SecondaryServer> Y                  N                   ONLINE

-- WAN HEARTBEAT STATE
-- Heartbeat             To                  State
L Icmp                  csm_primary        ALIVE

-- REMOTE CLUSTER STATE
-- Cluster               State
M csm_primary          RUNNING

-- REMOTE SYSTEM STATE
-- cluster:system        State                Frozen
N csm_primary:<PrimaryServer> RUNNING             0

-- REMOTE GROUP STATE
-- Group                 cluster:system      Probed              AutoDisabled        State
O APP                   csm_primary:<PrimaryServer> Y                  N                   ONLINE
```

ステップ 2 サーバ障害を発生させるためにプライマリ サーバの電源を切ります。セカンダリ クラスタがプライマリ クラスタへの接続の切断を報告したことを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System                State                Frozen
A <SecondaryServer>     RUNNING             0

-- GROUP STATE
-- Group                 System              Probed              AutoDisabled        State
B APP                   <SecondaryServer>  Y                  N                   OFFLINE
B APPPrep               <SecondaryServer> Y                  N                   ONLINE
B ClusterService       <SecondaryServer> Y                  N                   ONLINE

-- WAN HEARTBEAT STATE
-- Heartbeat             To                  State
L Icmp                  csm_primary        ALIVE

-- REMOTE CLUSTER STATE
-- Cluster               State
```

```

M csm_primary      LOST_CONN

-- REMOTE SYSTEM STATE
-- cluster:system      State      Frozen
N csm_primary:<PrimaryServer> RUNNING      0

-- REMOTE GROUP STATE
-- Group      cluster:system      Probed      AutoDisabled      State
O APP      csm_primary:<PrimaryServer> Y      N      ONLINE

```

ステップ 3 レプリケーションが切断状態であることを確認します。次のコマンドの出力に含まれる **flags** パラメータでこの状態を確認できます。

```

C:¥> vxprint -P1
Diskgroup = BasicGroup

Diskgroup = datadg

Rlink      : rlk_172_32481
info      : timeout=500 packet_size=1400
           latency_high_mark=10000 latency_low_mark=9950
           bandwidth_limit=none
state     : state=ACTIVE
           synchronous=off latencyprot=off srlprot=off
assoc    : rvg=CSM_RVG
           remote_host=172.25.84.33
           remote_dg=datadg
           remote_rlink=rlk_172_6037
           local_host=172.25.84.34
protocol  : UDP/IP
flags     : write attached consistent disconnected

```

ステップ 4 次のコマンドを使用して、セカンダリ サーバでアプリケーションを開始します。

```
C:¥> hagrp -online -force APP -sys secondary_server_name
```

ステップ 5 アプリケーションにログインし、一部のデータを変更して、アプリケーションがセカンダリ サーバで動作中に行われた変更が、プライマリ サーバに戻る際に保持されることを後から確認できるようにします。

ステップ 6 プライマリ サーバの電源を入れ、サーバが完全に起動できるようにします。

ステップ 7 レプリケーションは接続しているが、2つのサイドは同期を維持していないことを示すレプリケーションの状況を確認します。

```

C:¥> vxprint -P1
Diskgroup = BasicGroup

Diskgroup = datadg

Rlink      : rlk_172_32481
info      : timeout=500 packet_size=1400
           latency_high_mark=10000 latency_low_mark=9950
           bandwidth_limit=none
state     : state=ACTIVE
           synchronous=off latencyprot=off srlprot=off
assoc    : rvg=CSM_RVG
           remote_host=172.25.84.33
           remote_dg=datadg
           remote_rlink=rlk_172_6037
           local_host=172.25.84.34
protocol  : UDP/IP
flags     : write attached consistent connected dcm_logging failback_logging

```

- ステップ 8** 元のプライマリ RVG をセカンダリに変換し、高速フェールバック機能を使用して、元のプライマリ RVG 内のデータ ボリュームと新しいプライマリ RVG 内のデータ ボリュームを同期します。セカンダリ クラスタの Cluster Explorer を使用して、RVGPrimary リソース ([APP_RVGPrimary]) を右クリックし、[actions] を選択し、[Actions] ダイアログボックスから [fbsync] を選択して、次に [OK] をクリックします。あるいは、次のコマンドを実行することもできます。

```
C:¥> hares -action APP_RVGPrimary fbsync 0 -sys secondary_server_name
```

- ステップ 9** 次のコマンドの出力の **flags** パラメータで、キーワード **consistent** を検索し、現在のセカンダリ（以前のプライマリ）が、現在のプライマリ（以前のセカンダリ）との同期を維持していることを確認します。

```
C:¥> vxprint -Pl
Diskgroup = BasicGroup

Diskgroup = datadg

Rlink      : rlk_172_32481
info       : timeout=29 packet_size=1400
            latency_high_mark=10000 latency_low_mark=9950
            bandwidth_limit=none
state      : state=ACTIVE
            synchronous=off latencyprot=off srlprot=off
assoc      : rvg=CSM_RVG
            remote_host=172.25.84.33
            remote_dg=datadg
            remote_rlink=rlk_172_6037
            local_host=172.25.84.34
protocol   : UDP/IP
flags      : write attached consistent connected
```

- ステップ 10** セカンダリ クラスタで VCS Cluster Explorer を使用して、[APP] サービス グループを選択します。ショートカット メニューで [Switch To] を選択してから [Remote Switch(...)] を選択し、[Switch global] ダイアログボックスを開きます。このダイアログボックスで、プライマリ クラスタとプライマリ サーバを指定します。あるいは、次のコマンドを実行します。ここで *primarycluster* はプライマリ クラスタの名前です。

```
C:¥> hagrps -switch APP -any -clus primarycluster
```

- ステップ 11** アプリケーションにログインして、セカンダリ サーバで行った変更が保持されたことを確認します。

アプリケーション障害

ここでは、Security Manager アプリケーションで障害が発生する事例について説明します。単一クラスタ コンフィギュレーションと、デュアルクラスタ コンフィギュレーションの 2 つのケースについて説明します。ここでは、次の内容について説明します。

- 「アプリケーション障害、単一クラスタ」(P.B-16)
- 「アプリケーション障害、デュアルクラスタ」(P.B-17)

アプリケーション障害、単一クラスタ

事例タイトル：単一クラスタ コンフィギュレーションでのプライマリ サーバでアプリケーションに障害が発生します。

説明：この事例は、VCS がアプリケーション障害を検出し、VCS がアプリケーションをセカンダリサーバに自動的に移行することを検証します。

テスト セットアップ：デフォルトのアプリケーション フェールオーバーの動作を使用したデュアルノード クラスタ (図 1-1 (P.12))。

- ステップ 1** 次のコマンドを実行することにより、APP サービス グループがクラスタ内のプライマリ サーバで実行中であることを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System                State                Frozen
A <PrimaryServer>        RUNNING              0
A <SecondaryServer>      RUNNING              0

-- GROUP STATE
-- Group                  System                Probed    AutoDisabled    State
B APP                    <PrimaryServer>      Y         N                ONLINE
B APP                    <SecondaryServer>   Y         N                OFFLINE
```

- ステップ 2** Security Manager が実行されているサーバで次のコマンドを実行して、アプリケーションを停止します。

```
C:¥> net stop crmdmgtd
```

- ステップ 3** VCS が、プライマリ サーバで Security Manager に障害が発生したことを検出し、アプリケーションをセカンダリ サーバで開始することを確認します。

```
# hastatus -sum
-- SYSTEM STATE
-- System                State                Frozen
A <PrimaryServer>        RUNNING              0
A <SecondaryServer>      RUNNING              0

-- GROUP STATE
-- Group                  System                Probed    AutoDisabled    State
B APP                    <PrimaryServer>      Y         N                OFFLINE | FAULTED
B APP                    <SecondaryServer>   Y         N                ONLINE

-- RESOURCES FAILED
-- Group                  Type                Resource                System
C APP                    CSManager            APP_CSManager          <PrimaryServer>
```

- ステップ 4** APP サービス グループの障害を手動で解消します。

```
C:¥> hagr -clear APP -sys primary_server_name
```

- ステップ 5** APP サービス グループを手動でプライマリ サーバに戻します。

```
C:¥> hagr -switch APP -to primary_server_name
```

アプリケーション障害、デュアル クラスタ

事例タイトル：デュアル クラスタ コンフィギュレーションでのプライマリ サーバでアプリケーションに障害が発生します。

説明：この事例は、VCS がアプリケーション障害を検出することを検証します。

テスト セットアップ: 各サーバに単一ノードを備えたデュアル クラスタ コンフィギュレーション (レプリケーションあり) (図 1-2 (P.14))。同様に、デフォルトのアプリケーション フェールオーバー動作は変更されていない (つまり、クラスタ間のフェールオーバーには手動介入が必要である) ことを前提にしています。

ステップ 1 次のコマンドをプライマリ サーバから実行して、APP サービス グループおよび ClusterService サービス グループがプライマリ サーバで実行されていることを確認します。

```
C:¥> hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <SecondaryServer>  RUNNING      0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B APP              <SecondaryServer> Y         N           OFFLINE
B APPPrep         <SecondaryServer> Y         N           ONLINE
B ClusterService <SecondaryServer> Y         N           ONLINE

-- WAN HEARTBEAT STATE
-- Heartbeat       To              State
L Icmp            csm_primary    ALIVE

-- REMOTE CLUSTER STATE
-- Cluster         State
M csm_primary     RUNNING

-- REMOTE SYSTEM STATE
-- cluster:system  State          Frozen
N csm_primary:<PrimaryServer> RUNNING      0

-- REMOTE GROUP STATE
-- Group           cluster:system  Probed    AutoDisabled  State
O APP              csm_primary:<PrimaryServer> Y         N           ONLINE
```

ステップ 2 Security Manager が実行されているサーバで次のコマンドを実行して、アプリケーションを停止します。

```
C:¥> net stop crmdmgt
```

ステップ 3 VCS が、アプリケーションが失敗したことを検出し、APP サービス グループを停止することを確認します。次のコマンドを実行して出力を確認します。

```
# hastatus -sum
-- SYSTEM STATE
-- System          State          Frozen
A <PrimaryServer>  RUNNING      0

-- GROUP STATE
-- Group           System          Probed    AutoDisabled  State
B APP              <PrimaryServer> Y         N           OFFLINE | FAULTED
B APPPrep         <PrimaryServer> Y         N           ONLINE
B ClusterService <PrimaryServer> Y         N           ONLINE

-- RESOURCES FAILED
-- Group           Type            Resource          System
C APP              CSManager      APP_CSManager    <PrimaryServer>

-- WAN HEARTBEAT STATE
-- Heartbeat       To              State
L Icmp            csm_secondary  ALIVE
```

```
-- REMOTE CLUSTER STATE
-- Cluster          State
M csm_secondary    RUNNING

-- REMOTE SYSTEM STATE
-- cluster:system   State          Frozen
N csm_secondary:<SecondaryServer> RUNNING      0

-- REMOTE GROUP STATE
-- Group            cluster:system   Probed    AutoDisabled  State
O APP              csm_secondary:<SecondaryServer> Y         N         OFFLINE
```

ステップ 4 APP サービス グループの障害を手動で解消します。

```
C:¥> hagr -clear APP
```

ステップ 5 プライマリ サーバで APP サービス グループをオンラインにして、アプリケーションを再起動します。

```
C:¥> hagr -online APP -sys primary_server_name
```

■ アプリケーション障害