

応募区分：研究型論文

VersaStack と UCS Director による災害対策の自動化検証

森 純也 (もり じゅんや)
株式会社ネットワーク SI 技術本部
データセンタソリューション課

■ 要約

昨今、ユーザーの求めるシステム基盤において、運用の容易性と正確性は必須条件となっており、合わせて SDDC、SDS や SDN といった、注目されているキーワードから IT が向いている方向を鑑みると、今後、オーケストレーションツールは、注目すべきソリューションの 1 コンポーネントと言える。また、オーケストレーションツールの利点としては、従来、Storage、Network、Compute など、各分野の有識者を多数擁して管理していたシステムの属人性を緩和し、作業スピードも向上させることで、運用コストの大幅削減も期待できる。

そこで、今回の検証題材として、垂直統合基盤(コンバージドインフラストラクチャ)である VersaStack の環境に Cisco UCS Director を導入し、DR 環境のレプリケーションの自動化と、論理障害時の逆向きのミラー、本番サイト災害時の切り替え作業をオーケストレーションすることで、障害復旧までの時間を短縮し、又、有事の際、限られた人的リソース下でも、シンプルにサイト切り替えや、復旧後の切り戻しができることを実証し、その有用性を確認した。

目次

1. 本件検証の概要.....	4
1.1. 全体構成図.....	4
1.1.1. Storwize V7000 のリモート・コピー自動化検証概要.....	5
1.1.2. 逆向きミラーによる、データのリストア検証概要.....	5
1.1.3. 本番サイトの激甚被害時の災対サイト切り替え検証概要.....	6
2. 検証手順・結果.....	6
2.1. Storwize V7000 のリモート・コピー自動化検証結果.....	6
2.1.1. Storwize V7000 のリモート・コピー自動化検証設定・手順.....	6
2.1.2. Storwize V7000 のリモート・コピー自動化検証動作.....	7
2.1.3. Storwize V7000 のリモート・コピー自動化検証まとめ.....	7
2.2. 逆向きミラーによる、データのリストア検証結果.....	8
2.2.1. 逆向きミラーによる、データのリストア検証設定・手順.....	8
2.2.2. 逆向きミラーによる、データのリストア検証動作.....	10
2.2.3. 逆向きミラーによる、データのリストア検証まとめ.....	10
2.3. 本番サイトの激甚被害時の災対サイト切り替え検証結果.....	10
2.3.1. 本番サイトの激甚被害時の災対サイト切り替え検証設定・手順.....	10
2.3.2. 本番サイトの激甚被害時の災対サイト切り替え検証動作.....	15
2.3.3. 本番サイトの激甚被害時の災対サイト切り替え検証まとめ.....	15
3. 総評.....	16
3.1. 機能面での改善点.....	16
3.2. 所感.....	18

1. 本件検証の概要

1.1. 全体構成図

今回の検証では、以下の『図1. 全体構成図』の通り、2サイトの仮想基盤環境を構成し、Storwize V7000 ストレージのリモート・コピー機能でDRを行う。その際に必要となるミラーの定期更新や、障害復旧に至る、煩雑かつ属人化しがちな以下の作業を、UCS Director のオーケストレーションの機能を活用し、オペレーションの簡略化の可否を検証する。

- ・Storwize V7000 のリモート・コピー自動化
- ・逆向きミラーによる、データのリストア作業
- ・本番サイトの甚大被害時の災対サイト稼働切り替え作業

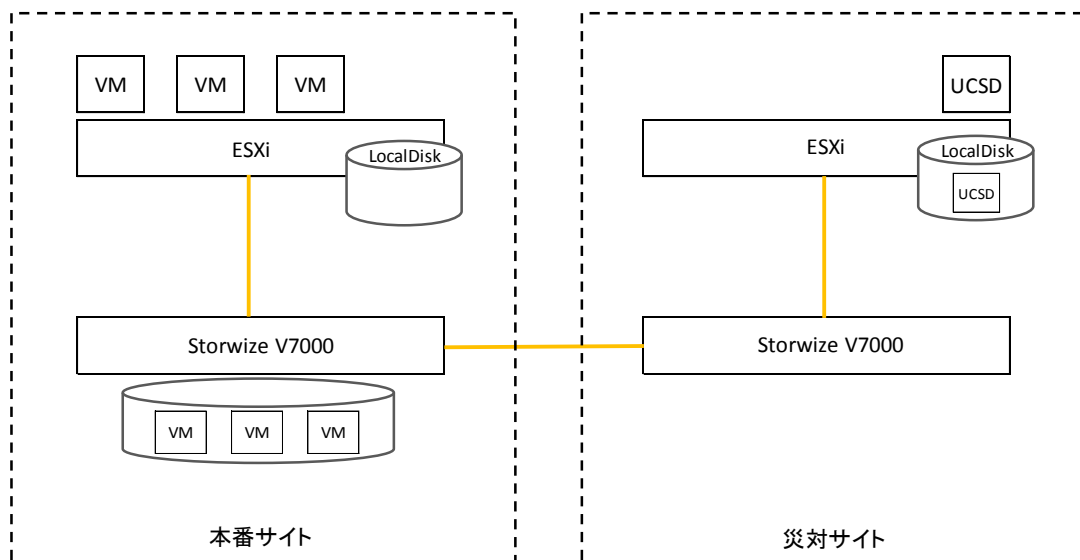


図1. 全体構成図

本検証を開始するに当たって、それぞれのコンポーネントに対して通信が出来る環境を設定し、UCS Director に対して以下の環境設定を実施した。

表1. UCS Director 環境設定

【Physical Accounts】-【Physical Accounts】

Account Name	Account Type	備考
V7K_TOKYO_Site	IBM Storwize V7000	擬似本番サイトストレージ
V7K_OSAKA_Site	IBM Storwize V7000	擬似災対サイトストレージ

【Virtual Accounts】-【Virtual Accounts】

Cloud Name	Account Type	備考
TOKYO_Site	VM ware	擬似本番サイト
OSAKA_Site	VM ware	擬似災対サイト

1.1.1. Storwize V7000 のリモート・コピー自動化検証概要

Storwize V7000 は、リモート・コピーと呼ばれる、遠隔地とボリュームや、コンシステンシーグループ単位でミラーを実施する機能を備えている。遠隔地とレプリケーションを実施する場合は一般的に、メトロ・ミラーと呼ばれる非同期型のミラーを実施するケースが多い。ソースボリュームとターゲットボリュームに関しては、数秒遅れで常に同期が取られるが、通常、論理障害や、本番サイトの障害によるデータの不整合による、両ボリュームの論理破損を担保する為に、災対サイト側で、正常時のターゲットボリュームを静止点としたバックアップとして、定期的にフラッシュ・コピーを実施する事が推奨される。このフラッシュ・コピーの定期的な更新処理に関しては、Storwize V7000 の基本機能では実施不可能で、スクリプトを構成して、別に用意したサーバのタスクスケジュール機能等で実施する必要がある。この定期更新処理を『図2. リモート・コピー自動化』に示したとおり、UCS Director 内のオーケストレーションワークフロースケジュール実行機能で、容易にエラーハンドル、メール通知も機能するワークフローの作成を行い、正常に機能するか検証する。

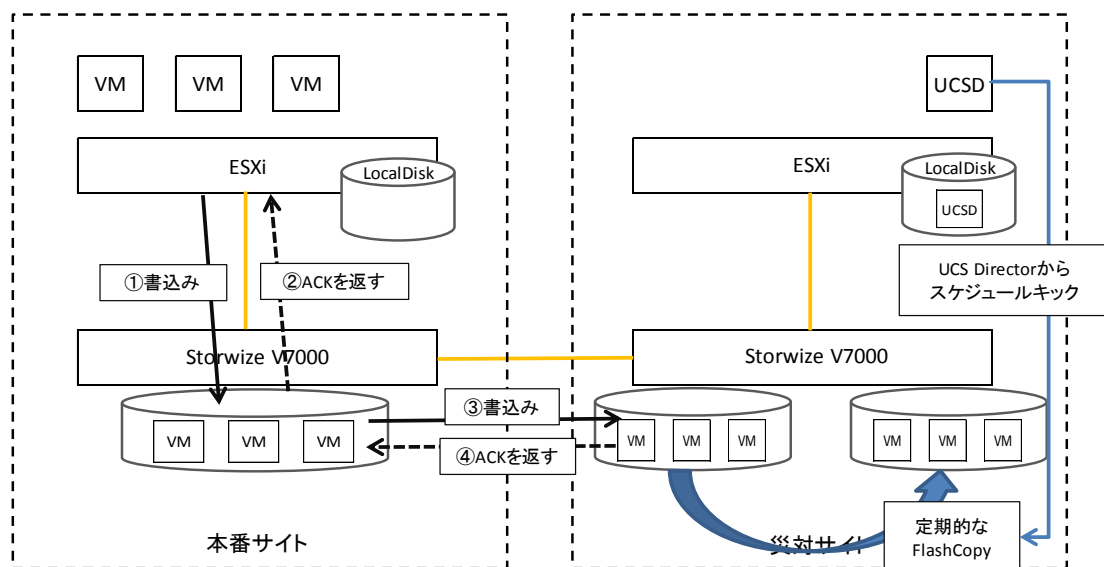


図2. リモート・コピー自動化

1.1.2. 逆向きミラーによる、データのリストア検証概要

本検証環境は、前項にて記載したリモート・コピーを利用した DR 環境だが、非同期という名称ではあるものの、ソースとターゲットは数秒遅れで常に同期をしている為、論理障害発生時は、以下の様なフローで災対先のフラッシュ・コピーからリストアを実施する必要がある。

- ① 本番サイト→災対サイトへのリモート・コピー一時停止
- ② 本番 ESXi のシャットダウン
- ③ 災対フラッシュ・コピーの逆向きのコピー
- ④ 災対サイト→本番サイトへのリモート・コピー

- ⑤ 本番サイト→災対サイトへのリモート・コピー再開
- ⑥ 本番 ESXi の起動

上記を手動で実施する場合は、ESXi 管理者と Storage 管理者間で、それぞれの実行フェイズを確認しながら、シーケンシャルに実施する必要があるため、一つのヒューマンエラーでデータ破損などの甚大な被害を起こしかねない作業となる。検証の目的として UCS Director のオーケストレーション機能で、簡易的なオペレーションのみで構成できるかを検証する。

1.1.3. 本番サイトの激甚被害時の災対サイト切り替え検証概要

本番サイト側が、自然災害などによる甚大な被害で復旧困難な場合を想定し、災対サイト側で仮想マシンを起動する場合には、以下のフローを実施する必要がある。

- ① 本番サイト→災対サイトへのリモート・コピー一時停止
- ② 災対サイトのフラッシュ・コピーを逆向きコピー
- ③ 災対サイトボリュームの ESXi データストアマウント
- ④ 災対 ESXi の仮想マシンインベントリ追加

又、本番サイト復旧後には、以下のフローを実施し、正常な状態に戻す必要がある。

- ① 災対サイト→本番サイトへのリモート・コピー
- ② 災対サイト→本番サイトへのリモート・コピー削除
- ③ 災対 ESXi の仮想マシンインベントリ削除
- ④ 災対 ESXi のデータストアアンマウント
- ⑤ 本番サイト→災対サイトへのリモート・コピー再開
- ⑥ 本番サイト ESXi 起動
- ⑦ 本番サイト ESXi で災対稼働時に追加された仮想マシンのインベントリ追加

本番サイト復旧後の処理に関しては、復旧の状況、運用設計に応じて、手動による設定や更に詳細なタスクが必要となるケースも想定されるが、本検証では災害前の状況を想定し、UCS Director のオーケストレーション機能で、簡易的なオペレーションのみで構成できるかを検証する。

2. 検証手順・結果

2.1. Storwize V7000 のリモート・コピー自動化検証結果

2.1.1. Storwize V7000 のリモート・コピー自動化検証設定・手順

『図 3』『表 1』の通り、ワークフロー「RemoteSiteFlashCopyUpdate」を作成し、5 分おきにワークフローが実行され、フラッシュ・コピーの更新も実施されることを確認する。

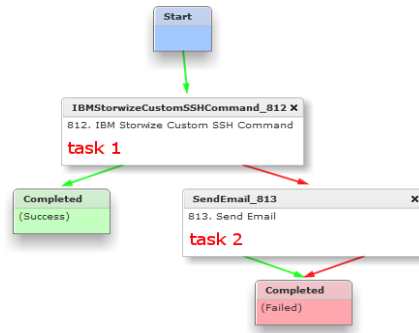


図3. リモート・コピー自動化ワークフロー図

表2. リモート・コピー自動化ワークフロー設定

Task Name		Task Type
設定項目(InputLabel)	設定値	
RemoteSiteFlashCopyUpdate		
task 1	IBM Storwize Custom SSH Command	
FlashCopy_Update	svctask mkfcmap -name <FCname> -cleanrate 100 -copyrate 100 -incremental -source <src_volname> -target <dst_volname>	
task 2	Send Email	
E-mail Addresses	<mail address>	
Subjet	remote FlashCopy failed	
Body	-	

Workflow ID	Workflow Name	Schedules Time	Frequency Type	Frequency Interval	RecurrenceType	Repeat Count
70	RemoteSiteFlashCopyUpdate	設定した日時	Minutes	5	No END	Not Applicable

2.1.2. Storwize V7000 のリモート・コピー自動化検証動作

『図4』の通り、Frequency 設定で定義した様に、5分おきにフラッシュ・コピーが更新されることが確認できた。本フラッシュ・コピーは、取得開始時のデータを静止点としてコピーを取得する為、仮にコピー中に論理障害や、本番サイトで障害が発生しても影響することなく、リストア可能なイメージとしてバックアップを取得することが可能となる。



図4. リモート・コピー自動化実行結果

2.1.3. Storwize V7000 のリモート・コピー自動化検証まとめ

動作は想定どおり、かつ、安定して動作した。仮に UCS Director が無い環境で、本シナリオを実行するにあたって必要となるスクリプトの作成期間や、実行するサーバの用意等の手間と比べて工数も少なく、ミラーの更新を実施するコマンドを確認し、ワークフロー作成、スケジュール定義まで、わずか数分で構成することが出来た。

2.2. 逆向きミラーによる、データのリストア検証結果

2.2.1. 逆向きミラーによる、データのリストア検証設定・手順

『図5』『表3』の通り、ワークフロー「RemoteCopy 切り戻し」を作成する。その後、本番サイトの仮想マシンを1台、ディスクから消去した上で、作成したワークフローを実施し、災対サイトからの逆向きリモート・コピーによるリストアで、削除した仮想マシンが、データストア上にリストアされることを確認する。



図5. リストア自動化ワークフロー図

表3. リストア自動化ワークフロー設定

【Orchestration】-【Workflows】	
Workflow Name	
Task Name	設定値
項目	設定値
RemoteCopy 切り戻し	
TOKYO_Site ESXi Shutdown	
Task Category	VMware Host Tasks
Task Type	VMware Host Power Action
Task Input	
Select Host	TOKYO_Site
Host Power Action	Shutdown
RemoteCopyStop	
Task Category	IBM Storwize RemoteCopy Tasks
Task Type	Stop IBM Storwize Remote Copy Relationship
Task Input	
Remote Copy Relationship	V7K_OSAKA_Site:0:RemoteCopy
Allow Secondary read/write access	チェック有り
RemoteFCReverseStart	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K_OSAKA_Site
Command	mkfcmap -name RemoteFCRe -cleanrate 100 -copyrate 100 -incremental -source Datastore_OSAKA_Site_01 -target Datastore_OSAKA_Site:startfcmap -prep RemoteFCRe
RemoteFCReverseCheck	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K_OSAKA_Site
Command	func_fc_check 0 { srcfcmap while read id name source_vdisk_id source_vdisk_name target_vdisk_id target_vdisk_name status progress copy_rate clean_progress incremental partner_FC_id partner_FC_name restoring start_time rc_controlled ; do if ["\$1" = "\$name"] ; then if ["idle_or_copied" = "\$status"] ; then return 100 ; fi ; fi ; done ; if ["\$?" = "100"] ; then return 0 ; else return 1 ; fi ; } ; func_fc_retry 0 { func_fc_check \$1 ; if ["1" = "\$?"] ; then echo retry after 1minut & sleep 60 && func_fc_retry \$1 ; else return 0 ; fi ; } ; func_fc_retry RemoteFCRe
RemoteFCReverseDelete	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K_OSAKA_Site
Command	rnfcmmap -force RemoteFCRe
RemoteCopyReverse	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K_OSAKA_Site
Command	startrcrelationship -force -primary aux RemoteCopy
RemoteCopyCheck	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K_OSAKA_Site
Command	func_rc_check 0 { srcrelationship while read id name master_cluster_id master_cluster_name master_vdisk_id master_vdisk_name aux_cluster_id aux_cluster_name aux_vdisk_id aux_vdisk_name primary state bg_copy_priority progress copy_type cycling_mode freeze_time ; do if ["\$1" = "\$name"] ; then if ["consistent_synchronized" = "\$state"] ; then return 100 ; fi ; fi ; done ; if ["\$?" = "100"] ; then return 0 ; else return 1 ; fi ; } ; func_rc_retry 0 { func_rc_check \$1 ; if ["1" = "\$?"] ; then echo retry after 1minut & sleep 60 && func_rc_retry \$1 ; else return 0 ; fi ; } ; func_rc_retry RemoteCopy
RemoteCopySwitch	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K_OSAKA_Site
Command	switchrcrelationship -primary master RemoteCopy
UCSServerPowerOn	
Task Category	General Tasks
Task Type	SSH Command
Task Input	
Host IP Address	xxx.xxx.xxx.xxx (SSH接続用IPアドレス)
Login	admin (SSHログイン用ユーザー名)
Password	***** (SSHログイン用パスワード)
Command	scope org IBM;scope service-profile IBM_ESX1;power up;commit-buffer

2.2.2. 逆向きミラーによる、データのリストア検証動作

ワークフローは全て正常に完了した。ワークフロー内で再起動を実行していた ESXi が起動した後、『図 6』の通り、データのリストア成功が確認でき、インベントリの再登録と起動も確認することができた。

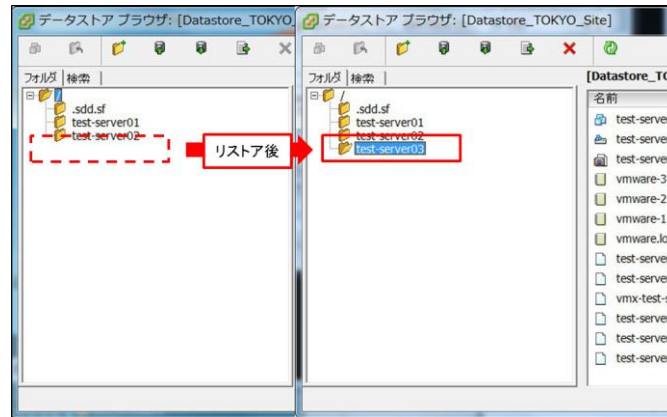


図 6. リストア前後のデータストア

2.2.3. 逆向きミラーによる、データのリストア検証まとめ

本検証においては、本来煩雑な作業を極簡単なオペレーションで実施でき、また、ヒューマンエラーなども限りなく少なくすることに成功したと言えるが、ワークフローを作成するに至って、プリセットのタスクでは前ジョブの正常完了を確認した後、次のタスクに遷移することが実現できなかった。SSH の状態確認コマンドで特定のアウトプットが出力されるまで、数分おきに実施するなどのカスタマイズが必要となった。カスタマイズする際のコマンドは操作対象の機器の SSH CLI で実装されている内容に依存する部分も多い。今後の課題として、特にコンバージドインフラで構成されているハードウェアに関しては、カスタマイズで準備することなくプリセットコマンドの標準機能として実装されていることが望ましい。

例として、現時点で実装されているアウトプットの文字列に応じてタスクエラーとする機能を、逆に許可一覧(ホワイトリスト)化し、エラー時の繰り返し処理回数を 5 回以上実施できるようにするだけでも、大幅な改善が期待出来る。

2.3. 本番サイトの激甚被害時の災対サイト切り替え検証結果

2.3.1. 本番サイトの激甚被害時の災対サイト切り替え検証設定・手順

まず、切り替え処理の検証として、『図 7』『表 4』の通り、ワークフロー「災対サイト稼働」を作成する。作成したワークフローを実施し、災対サイト側の LUN がアクティブに切り替わり、災対サイト側の ESXi 環境で LUN マッピングやデータストアマウント、仮想マシンのインベントリ追加が自動で実施されることを確認する。

次に切り戻し処理の検証として、『図 8』『表 5』の通り、ワークフロー「災対サイト稼働からの切り戻し」を作成し、擬似的な災対サイト運用時の変更点として、新規仮想マシンを作成した上で、作成し

たワークフローを実行する。ワークフローで災対サイト側が切り替えワークフロー前の状態にもどることと、災対サイトの稼働開始、又、災対サイトで作成した仮想マシンのデータが逆リモート・コピーで更新され、インベントリ追加が自動で行われていることを確認する。

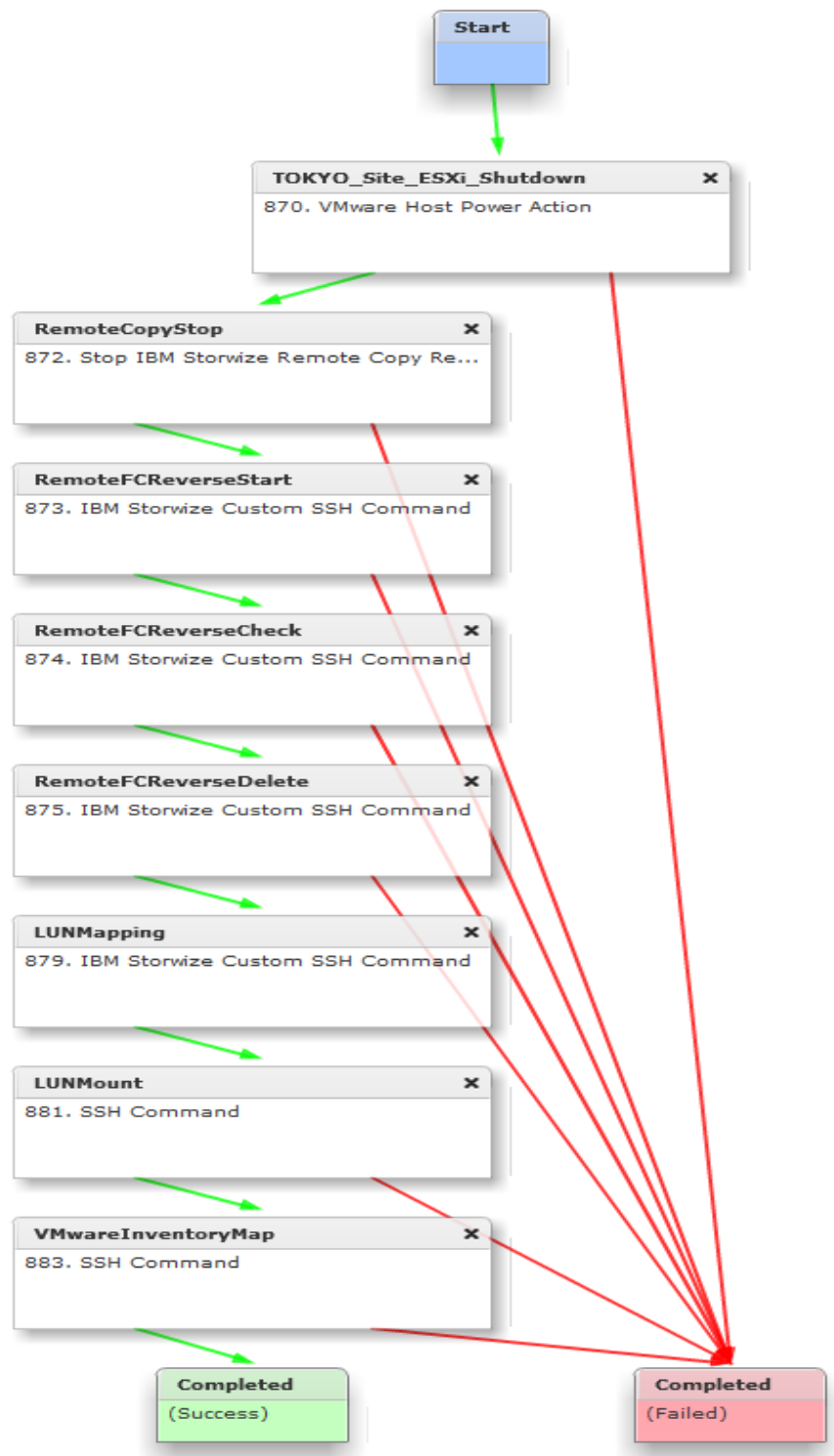


図7. 災対稼働ワークフロー図

表4. 災対稼働ワークフロー設定

【Orchestration】-【Workflows】	
Workflow Name	
Task Name	設定値
項目	
災対サイト稼働	
TOKYO Site ESXi Shutdown	
Task Category	VMware Host Tasks
Task Type	VMware Host Power Action
Task Input	
Select Host	TOKYO Site
Host Power Action	Shutdown
RemoteCopyStop	
Task Category	IBM Storwize RemoteCopy Tasks
Task Type	Stop IBM Storwize Remote Copy Relationship
Task Input	
Remote Copy Relationship	V7K_OSAKA_Site:0:RemoteCopy
Allow Secondary read/write access	チェック有り
RemoteFCReverseStart	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K_OSAKA_Site
Command	mkfcmmap -name RemoteFCRe -cleanrate 100 -copyrate 100 -incremental -source Datastore_OSAKA_Site_01 -target Datastore_OSAKA_Site:startfcmmap -prep RemoteFCRe
RemoteFCReverseCheck	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K_OSAKA_Site
Command	func_fc_check () { lsfcmmap while read id name source_vdisk_id source_vdisk_name target_vdisk_id target_vdisk_name status progress copy_rate clean_progress incremental partner_FC_id partner_FC_name restoring start_time rc_controlled ; do if ["\$1" = "\$name"]; then if ["idle_or_copied" = "\$status"]; then return 100 ; fi ; fi ; done ; if ["\$?" = "100"]; then return 0 ; else return 1 ; fi ; }; func_fc_retry () { func_fc_check \$1 ; if ["\$1" = "\$?"]; then echo retry after 1minit & sleep 60 && func_fc_retry \$1 ; else return 0 ; fi ; }; func_fc_retry RemoteFCRe
RemoteFCReverseDelete	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K_OSAKA_Site
Command	rmfcmmap -force RemoteFCRe
LUNMapping	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Host IP Address	V7K_OSAKA_Site
Command	mkvdiskhostmap -force -host UCSD_OSAKA_Site -scsi 30 Datastore_OSAKA_Site sleep 10
LUNMount	
Task Category	General Tasks
Task Type	SSH Command
Task Input	
Host IP Address	xxx.xxx.xxx.xxx (OSAKA_Site ESXi 管理IPアドレス)
Login	admin (SSHログイン用ユーザー名)
Password	***** (SSHログイン用パスワード)
Command	esxcli storage core adapter rescan --all esxcfg-volume -m 55d53cca-2f49eb9a-d0ea-0025b5db0001 sleep 10
VMwareInventoryMap	
Task Category	General Tasks
Task Type	SSH Command
Task Input	
Host IP Address	xxx.xxx.xxx.xxx (OSAKA_Site ESXi 管理IPアドレス)
Login	admin (SSHログイン用ユーザー名)
Password	***** (SSHログイン用パスワード)
Command	for vmname in `ls /vmfs/volumes/Datastore_TOKYO_Site/*/*.vmx` do vim-cmd solo/registervm \$vmname done



図8. 災対稼働からの切り戻しワークフロー図

表5. 災対稼働からの切り戻しワークフロー設定

[Orchestration]-[Workflows]	
Workflow Name	
Task Name	設定値
災対サイト稼働からの切り戻し	
RemoteCopyReverse	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K OSAKA Site
Command	startrcrelationship -force -primary aux RemoteCopy
RemoteCopyCheck	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K OSAKA Site
Command	func_rc_check () { lsrelationship while read id name master_cluster_id master_cluster_name master_vdisk_id master_vdisk_name aux_cluster_id aux_cluster_name aux_vdisk_id aux_vdisk_name primary state bg_copy_priority progress copy_type cycling_mode freeze_time;do if ["\$1" = "\$name"]; then if ["consistent_synchronized" = "\$state"]; then return 100 ;fi ;fi ;done ;if ["\$?" = "100"]; then return 0 ;else return 1 ;fi ;func_rc_retry () { func_rc_check \$1 ; if ["\$1" = "\$?"]; then echo retry after 1minit & sleep 60 && func_rc_retry \$1 ; else return 0 ; fi ;func_rc_retry RemoteCopy
VMwareInventoryUnmap	
Task Category	General Tasks
Task Type	SSH Command
Task Input	
Host IP Address	xxx.xxx.xxx.xxx (OSAKA Site ESXi 管理IPアドレス)
Login	admin (SSHログイン用ユーザー名)
Password	***** (SSHログイン用パスワード)
Command	IFS=BACKUP=\$IFS IFS=\$'\n' for vmid in `vim-cmd vmsvc/getallvms` do case "\$vmid" in *Datastore.TOKYO.Site*) vim-cmd vmsvc/unregister \$(vmid:0:4); *) esac done IFS=\$IFS.BACKUP
VMwareDatastoreUnmount	
Task Category	General Tasks
Task Type	SSH Command
Task Input	
Host IP Address	xxx.xxx.xxx.xxx (OSAKA Site ESXi 管理IPアドレス)
Login	admin (SSHログイン用ユーザー名)
Password	***** (SSHログイン用パスワード)
Command	esxcfg-volume -u 55d53cca-2f49eb9a-d0ea-0025b5db0001 (VMFS UUID)
RemoteCopySwitch	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K OSAKA Site
Command	switchrcrelationship -primary master RemoteCopy
LunUnmap	
Task Category	IBM Storwize Tasks
Task Type	IBM Storwize Custom SSH Command
Task Input	
Account Name	V7K OSAKA Site
Command	rmvdiskhostmap -host UCSD.OSAKA.Site Datastore.OSAKA.Site
UCSServerPowerOn	
Task Category	General Tasks
Task Type	SSH Command
Task Input	
Host IP Address	xxx.xxx.xxx.xxx (SSH接続用IPアドレス)
Login	admin (SSHログイン用ユーザー名)
Password	***** (SSHログイン用パスワード)
Command	scope org IBM:scope service-profile IBM_ESX1:power up;commit-buffer
VMInventoryAdd	
Task Category	General Tasks
Task Type	SSH Command
Retry Execution	チェック有り
Retry Count	5
Retry Frequency	300
Task Input	
Host IP Address	xxx.xxx.xxx.xxx (SSH接続用IPアドレス)
Login	admin (SSHログイン用ユーザー名)
Password	***** (SSHログイン用パスワード)
Command	for vmname in `ls /vmfs/volumes/Datastore.TOKYO.Site/` do echo \$vmname IFS_BACKUP=\$IFS IFS=\$:' for vmid in `vim-cmd vmsvc/getallvms` do case "\$vmid" in *\$vmname*) :: *) vim-cmd solo/registervm "/vmfs/volumes/Datastore.TOKYO.Site/\$vmname/\$vmname.vmx" esac done IFS=\$IFS.BACKUP done

2.3.2. 本番サイトの激甚被害時の災対サイト切り替え検証動作

切り替えに関しては、『図9』の通り、期待通り全て自動でフラッシュ・コピーのターゲットボリュームをアクティブなボリュームとして稼働させることができた。



図9. 災対稼働確認

切り戻しに関しても、『図10』の通り、想定どおりに災対サイトの戻し作業、逆りモード・コピー、本番サイトの再稼働を自動化することが出来た。

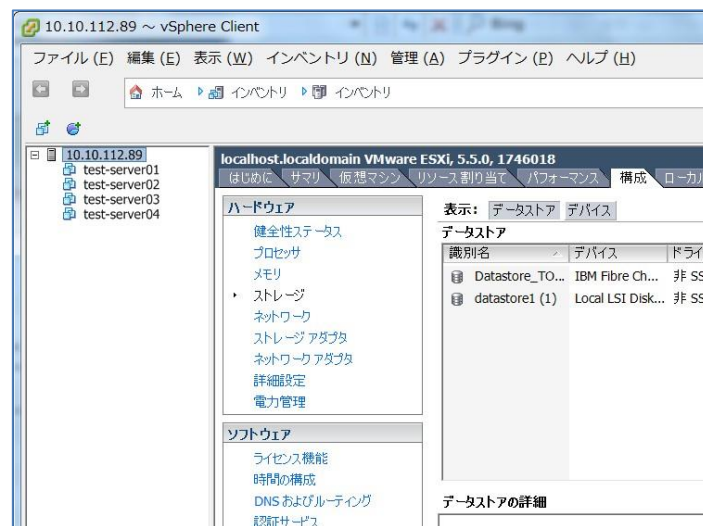


図10. 災対稼働からの切り替え確認

2.3.3. 本番サイトの激甚被害時の災対サイト切り替え検証まとめ

動作としては、想定通りの動きを再現させることができたが、埋め込み処理の多くがSSHセッションを利用したカスタマイズ設定となった。例としてVMwareのインベントリ登録のプリセットなどは、UCS Director側の標準プリセットとして実装されていないため、カスタマイズで対応を行った。

3. 総評

3.1. 機能面での改善点

プリセットで定義されているタスクに制限があるため、項目で見つからない場合は、`CustomSSHCommand` から、スクリプトを作成する必要がある。今後のバージョンアップなどでプリセット項目の拡充が望まれる。`CustomSSHCommand` の構成時も、簡単な実行コマンドの定義を実施して、UCS Director 側で、エラーワードを検知し、`UndoCommand` 実行、エラー通知などを実行することが可能な為、通常のスクリプト作成に比べて容易に構成できるといえる。

現存のプリセットタスクを精査すると、プリセットタスクを使用した際のワークフロー検証の機能が煩わしいケースがあった。今回の『災対サイトからの切り戻しタスク』は災害切り替え時に作成するのではなく、事前に作成して用意していく必要があるが、例として、切り戻し後に不要な `LUNMap` を外すタスクをプリセットで作成しようとする、現存の `LUNMap` から外す `LUN` と `Host` を選ばなければならない為、切り替えていない状態では候補に挙がらない。又、一時的に `LUNMap` を行い、候補に上げてタスクを作成しても、`LUNMap` を外した後、タスクデザイナーを一度でも開いてしまうと、ワークフロー検証の警告が出てしまうようになるなどが発生して、多くの `CustomSSHCommand` タスクの作成作業が発生した。この点は、タスク毎に検証を意図的にスキップできる設定などの実装が望まれる。

`CustomSSHCommand` 作成においては、それぞれのコンポーネントに対して、コマンドの流し込みや、スクリプトで動作を確認したものであっても、UCS Director から実行するとうまくいかないケースが多い。その原因も多岐にわたった為、構成する際に多くの時間を要した。例えば、`Storwize V7000` に関しては、本来『図 11』のコマンドを流せば、状態確認をして、コピー中なら 1 分後に再度確認を実施し、コピー完了ステータスになるまでループさせる動きを実機で確認していた。

しかしながら UCS Director で同じコマンドを入力したところ、「;」をつけても改行が許されていなかった為、全て一行で記載しなくてはならなかった。更に 1 行で記載をしても、UCS Director で、処理中にリターンコードが空欄になるケースや、2 つ目の `if` にリターンコードが引き渡されない事象が発生した為、『図 12』の様に、ファンクションで纏めた上で 1 行にすることで正常動作を確認することが出来た。カスタマイズ処理は柔軟な対応が出来る反面、スクリプトの実装経験に依存することになる。


```
func_fc_check () { |sfcmmap | while read id name source_vdisk_id source_vdisk_name target_vdisk_id target_vdisk_name sta
tus progress copy_rate clean_progress incremental partner_FC_id partner_FC_name restoring start_time rc_controlled;
do if [ "$1" = "$name" ];
then if [ "idle_or_copied" = "$status" ];
then return 100 ;
fi ;
done ;
if [ "$?" = "100" ] ;
then return 0 ;
else return 1 ;
fi ;
}
func_fc_retry () {
func_fc_check $1 ;
if [ "$?" = "$?" ] ;
then echo retry after 1minit & sleep 60 && func_fc_retry $1 ;
else return 0 ;
fi ;
}
```

図 11. フラッシュ・コピー状態確認コマンド Before

```
func_fc_check () { |sfcmmap | while read id name source_vdisk_id source_vdisk_name target_vdisk_id target_vdisk_name sta
tus progress copy_rate clean_progress incremental partner_FC_id partner_FC_name restoring start_time rc_controlled;
do if [ "$1" = "$name" ]; then if [ "idle_or_copied" = "$status" ]; then return 100 ; fi ; done ; if [ "$?" = "100
" ] ; then return 0 ; else return 1 ; fi ; }; func_fc_retry () { func_fc_check $1 ; if [ "$?" = "$?" ] ; then echo retry
after 1minit & sleep 60 && func_fc_retry $1 ; else return 0 ; fi ; }; func_fc_retry RemoteFCRe
```

図 12. フラッシュ・コピー状態確認コマンド After

VMware の CustomSSHCommand に関しては、改行は可能であったものの、『図 13』の様に、インベントリ追加のコマンドの時は、「\$」も正しく ESXi に渡していたが、インベントリ削除の時は、コマンド内の全ての「\$」以下文字列全てが消えてしまう事象が発生した。エスケープ文字で回避する策はすぐに対応できたものの、同様に CustomSSHCommand タスクで、同じようなコマンド内容であるにも関わらず、エスケープが必要なケースと不要なケースが起こるのは、実装後のテストで確認する必要がある。

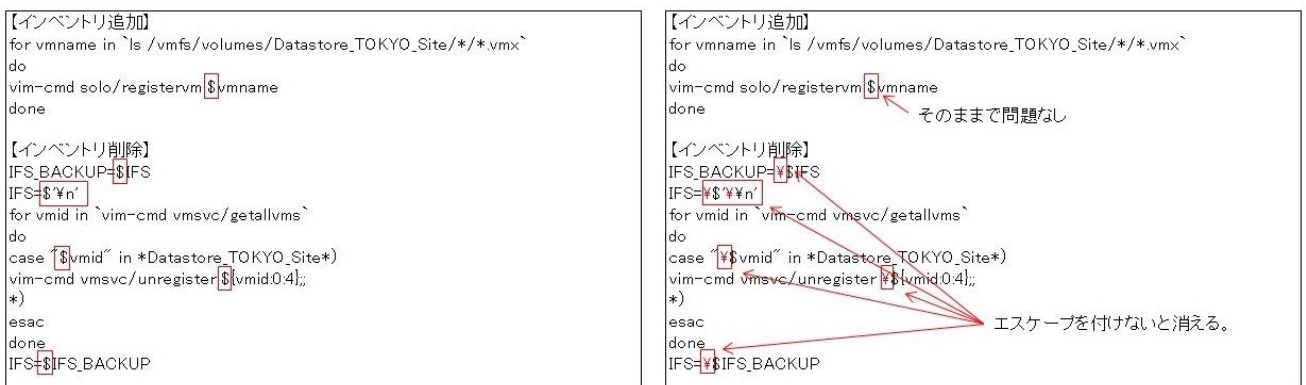


図 13. VMware インベントリ追加コマンド

又、災対サイトからの切り戻しタスク内に、データストア内の仮想マシンで、インベントリに無いものをインベントリ追加するコマンドを記載したが、ESXi に対する流し込みや、スクリプトでは問題なく動いていたが、UCS Director で動かすと、一定の仮想マシンを処理すると、途中で終了してしまう

現象が確認できている。各プロセスで echo などを挟み、止まる原因を追究したが、処理の途中で突然終了しており、なおかつタスク上は正常完了になっていた。ソフトウェアの不具合なのか、1 タスク内で操作機器に渡せる処理に行数、Byte などの制限があるのかは定かではないが、自動化させたい要望に対して、for などを実現する繰り返し処理は、当然想定される為、改善が望まれる。

上記の様な SSH のカスタマイズに関しては、コンポーネントによって、大きく仕様も変わるため、コマンドセットや実現方法が変わることは仕方の無い部分だが、各コンポーネントに対して、流し込み処理やスクリプト等の簡易テストで実施できた内容と、大きく記載方法を変えなくては動かない部分の改善は必須であり、又、エラー発生時の原因追求において、ブラウザ上のアウトプット表示が『図 13』の様に、冒頭の数行しか確認が出来ず、スクロールバーは付いているものの、下に動かすと次のタスクのログに移ってしまい、まともに確認できなかった。全ての表示を確認するには、ログを csv 等にエクスポートして毎回確認しなくてはならず、この点の改善も強く望まれる。

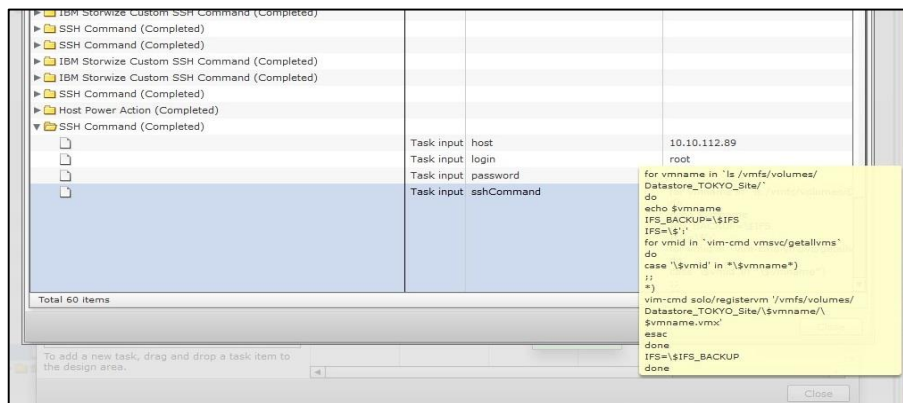


図 13. エラー発生時のアウトプット確認画面

3.2. 所感

本検証で使用した UCS Director バージョン 5.3 は、プリセットタスクについて不足していると感じる点もあったが、プリセットタスクが用意されている処理や、複雑な条件分岐が不要な処理に関しては、オペレーションの簡略化において、非常に有効である点が実証できた。VersaStack に構成される、Storwize V7000 に関しては、比較的どのような要件にもマッチする優れたストレージで、データ保護機能も潤沢に用意されているが、一部機能、特に今回検証を実施した、スケジュール実行などは、別のサーバからスクリプト管理が必要となるケースもあるが、UCS Director との組み合わせで、唯一懸念されるその点もカバーできることがわかった。今回は、特に人的リソースが制限される事が想定され、又、迅速に対応が必要な、DR 切り替え時のオーケストレーションを主題にして検証を実施したが、構成次第で、システム導入時の自動化や、通常運用時の設定変更など、多種多様なタスクに対して効果が見込める優れたツールであると同時に、特に VersaStack をはじめとする垂直統合基盤においては、作りこんだワークフローも含めて、ソリューションパッケージ化することも可能な分野になり、インテグレーターが付加価値、他との差別化ポイントとしてオーケストレーション環境を提供できるという意味でも、今後の展望が楽しいソリューションである。

以上