



IT Automation

オールインワン型HA同期構成 インストールマニュアル

※本書では「Exastro IT Automation」を「ITA」として記載します。

Exastro IT Automation Version
1.9.1

はじめに

1. 本資料について

- 本資料は、ITAオールインワンインストール済サーバ2台とOSS製品群を組み合わせ、HAクラスタを構成するのに必要な情報を記述したものです。ITAサーバをHAクラスタ化することにより、単一Linuxサーバでは実現できない可用性・信頼性の高いITAサーバを構築することができます。
- 本資料にもとづいて構成したHAクラスタは、以下の機能を提供します。
 - HAクラスタは2台のサーバシステムで構成される。それぞれのサーバは物理サーバであっても仮想サーバであっても構いません。
 - 常にどちらか1台のサーバ上でITAサーバが稼働し、ITAのWeb/AP、Backyard、DBMS、Ansibleドライバなどの全機能を提供します。
 - サービスを提供しているサーバがダウンすると、約1分前後で他のサーバ上でITAサーバが起動し、サービスが引き継がれます。この時、データベースのデータは失われません。
- デプロイ例は「4. システム構成例」の図を参照してください。ユーザ環境に合わせて、IPアドレス、ホスト名、ディレクトリ名、ファイル名を読み替えて構築してください。

- 本資料は、以下の環境で動作確認しています。

主要ソフトウェア	用途	検証バージョン
Red Hat Enterprise Linux release	OS	8.4
Exastro IT Automation	Exastro IT Automation	1.9.1
Pacemaker	クラスタ制御	2.1.0-8.el8
Corosync	ハートビート	3.1.5
pcs	Pacemakerコマンド	0.10.10
DRBD	レプリケーション	9.17.0-1

2. 責任範囲

- 本資料は、ITAサーバをクラスタ化する為の注意点や設定例を参考情報として示すものであり、これらの動作保証を行うものではありません。

3. Linux-HAクラスタスタックについて

- Linux-HAクラスタスタックは、サーバシステムの可用性を向上したり、ストレージ故障によるデータ喪失を防止したりすることを目的としたソフトウェア群です。次の3つのソフトウェアおよび関連パッケージによって構成されています。

DRBD (Distributed Replicated Block Device)

2台のサーバのストレージ(パーティションまたは論理ボリューム)にリアルタイムに同一データを書き込む(リアルタイムレプリケーション)。この為、片方のサーバやストレージ自体が故障した場合にもデータを喪失せず、正常なサーバ側でサービスを継続できるようになります。

Corosync

HAクラスタを構成するサーバの正常稼働を相互に監視するソフトウェア。

Pacemaker

HAクラスタが提供するサービスを監視し、サービスを提供していたサーバがダウンした場合に他のサーバでサービスを継続させるなどして、クラスタ全体の可用性を保ちます。

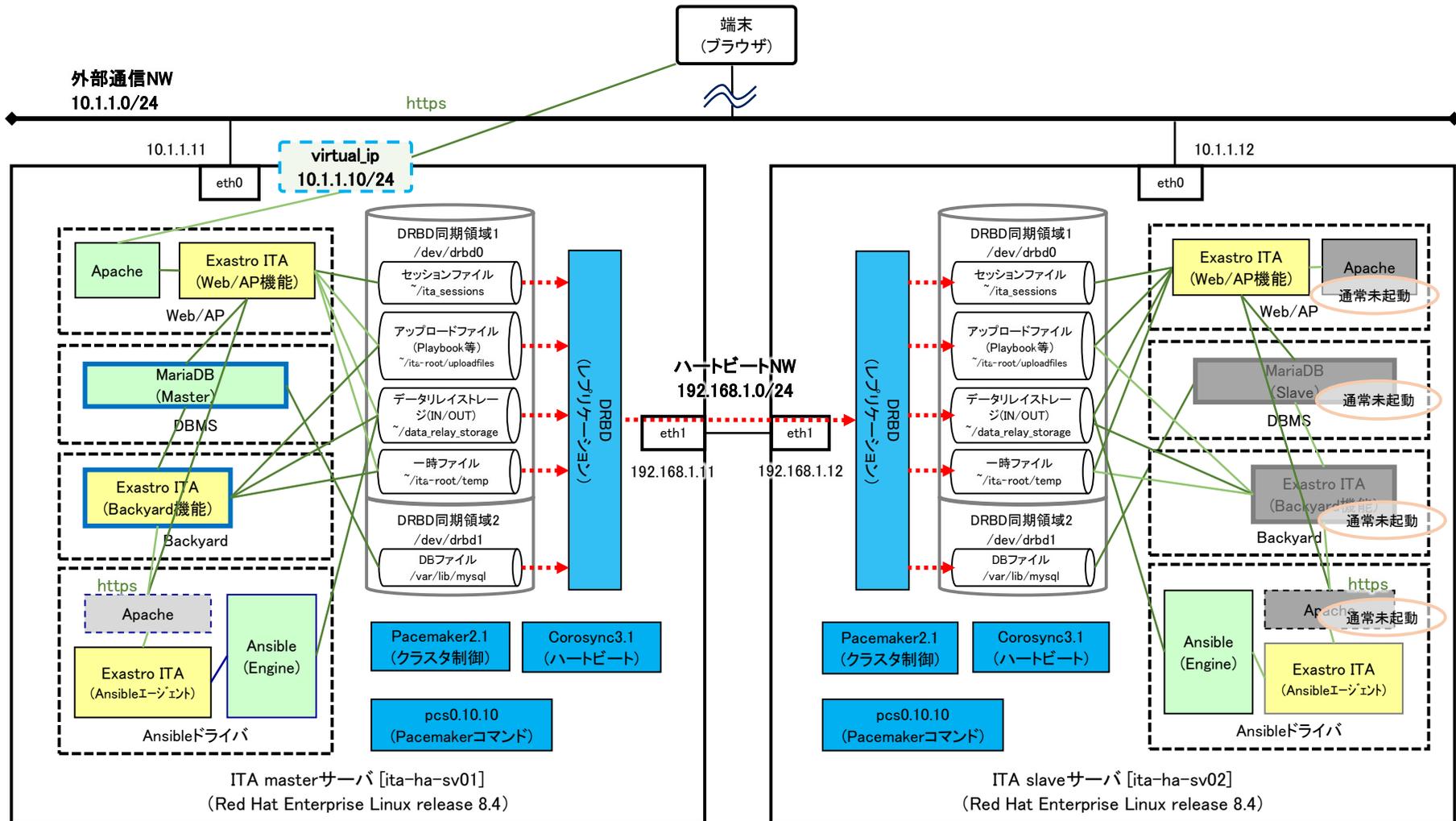
参照URL:

<http://linux-ha.osdn.jp/wp/>

はじめに

4. システム構成例

- ITAクラスタを構成する2台のサーバのディスクやネットワーク構成、関連プログラムは、下図のようになります。



はじめに

・ ネットワーク条件

- ・ 通常、各ネットワークインタフェースは、それぞれ固有のIPアドレスを持ちますが、ITAサーバのHAクラスタ構成を実現する為に追加のIPアドレス(仮想IPアドレス)が必要になります。用意した仮想IPアドレスは、構築後、Pacemakerにより自動的にmasterサーバ側に割り当てられます。

・ 本資料のネットワーク設定例

ネットワークIF	用途	masterサーバ	slaveサーバ
仮想IPアドレス	ITA接続	10.1.1.10/24	
eth0	外部通信NW	10.1.1.11/24	10.1.1.12/24
eth1	内部通信NW	192.168.1.11/24	192.168.1.12/24

- ・ 利用クライアントからサーバへ以下の通信ポートが利用可能なこと。

通信種別	ポート番号
http	80/tcp
https	443/tcp

- ・ クラスタ構成サーバを同一ネットワーク上に構成し、以下の通信ポートが利用可能であること。

通信種別	ポート番号
ssh(scp)	22/tcp
mariadb	3306/tcp
drbd	7788,7789/tcp ※1
Pacemaker	2224/tcp
corosync	5405/udp

※1 左記の通信ポートは、構築手順の設定により変更可能です。

・ ディスク条件

- ・ システムディスクとは別にHA構成サーバ間でのITAデータ連携の為、DRBD同期用ボリュームを準備し、パーティションを作成する必要があります。本資料の構築例では事前に作成・アタッチした"/dev/vdc"を利用して、パーティションおよび仮想ブロックデバイスを作成する手順を記述しています。

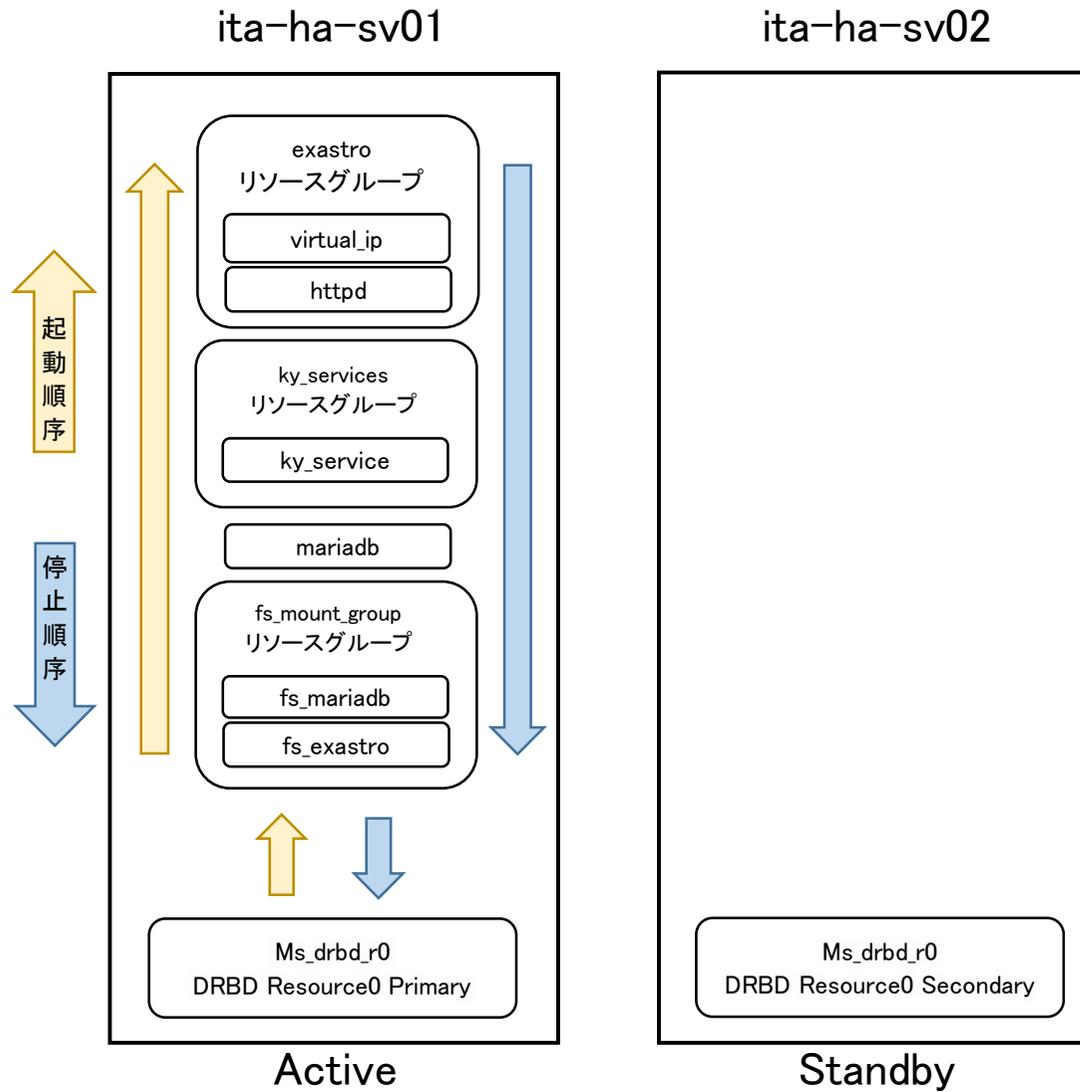
- ・ 必要となるボリュームサイズはITAの運用方法や作業の実行数により異なりますので想定されるITA運用を行い、実サイズを確認した上で、容量に余裕を持ったサイジングを行ってください。
- ・ 最低10GB以上を推奨します。また、必ずHAクラスタを構成するサーバ間で同一サイズのボリュームをそれぞれ用意してください。

・ 本資料でのDRBD同期用パーティション構築例

パーティション	仮想ブロックデバイス名	用途	サイズ
/dev/vdc1	/dev/drbd0	ITA同期用ファイルの格納先	5GB
/dev/vdc2	/dev/drbd1	MariaDBデータファイルの格納先	5GB

はじめに

- Pacemakerクラスタリソースの起動／停止順序について
 - 本資料でのPacemakerクラスタリソースの定義された起動／停止順序は下図のようになります。



5. 作業前提

- 以下の構築手順は作業済の前提で記載しています。
 - ① OS設定(ネットワーク設定など)
 - ② DRBD同期用ボリューム作成・接続 ※詳細は「4.システム構成例/ディスク条件」を参照のこと
 - ③ RHEL (Red Hat Enterprise Linux)のサブスクリプション登録
- 各種ソフトウェアのインストールはオンライン環境での手順を記載しています。
オフラインで実行する場合はソフトウェアの依存関係を構築サーバと合致させた上で、事前のライブラリ収集を行ってください。
- 構築手順の前に、以下の基本的なOS設定を実施して下さい。なお、特に断らない限り、OS基本設定は両方のサーバで同一に揃える必要があります。
 - SELinuxの無効化
SELinuxが有効な状態でHAクラスタを正常に運用するには、SELinuxに関する、極めて高度な知識と経験が必要になります。
本手順ではSELinuxを無効での検証となっておりますが、無効化はユーザの自己責任で対応ください。
 - 名前解決の設定
HAクラスタを構成するサーバ間で名前解決を行う必要があります。
本構築手順では、構築するサーバのホスト名とIPアドレスを、相互に/etc/hostsに登録しています。
ユーザ環境で利用するDNSサーバがある場合は、本手順の/etc/hostsの登録を行わず、HAサーバのホスト情報をDNSサーバに登録して名前解決を適宜実施してください。
 - NTPの設定
HAクラスタの運用管理にあたって、システムクロックが正確に同期していることは、きわめて重要です。
この為、可能な限りNTPプロトコルで時刻同期します。

はじめに

- ITAのインストールについて

- ITAの動作要件については、以下のドキュメントを参照してください。

https://exastro-suite.github.io/it-automation-docs/asset/Documents_ja/Exastro-ITAシステム構成/環境構築ガイド基本編.pdf

IT Automation BASE システム構成/環境構築ガイド 基本編

1.1 サーバ動作要件

https://exastro-suite.github.io/it-automation-docs/asset/Documents_ja/Exastro-ITAシステム構成/環境構築ガイドAnsible-driver編.pdf

Ansible-driver システム構成/環境構築ガイド Ansible-driver編

3. システム要件

- インストール要件

以下URLのオールインワン構成のインストールマニュアルを参照のこと。

ITAはオンライン/オフラインのどちらのインストール手順でも問題ありません。

https://exastro-suite.github.io/it-automation-docs/learn_ja.html#deploy

本資料では以下の機能有効化を前提としています。

ita_answers.txtの機能設定例

```
ita_base:yes
createparam:yes
hostgroup:yes
ansible_driver:yes
cobbler_driver:no
terraform_driver:yes
cicd_for_iac:no
```

構築手順「4.ITAサービス停止設定」と「6-22. 各ITAサービス(ky_*)のリソース設定をする」のITAの起動サービスはITAのバージョン、インストールする機能によって変動します。

上記の想定と異なるバージョンの利用と機能のインストールを行う場合は、本手順の下記の対象サービス内容について、適宜修正が必要となります。

- 構築手順「4.ITAサービス停止設定」の対象サービス
- 構築手順「6.Pacemaker設定」のPacemakerクラスタリソース、リソースグループ設定の対象サービス

構築手順

実行ユーザ / 接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
1.ITAのインストール					
1-1	ITAをインストールする	●	●	以下URLのオールインワン構成のインストールマニュアル参照 https://exastro-suite.github.io/it-automation-docs/learn_ja.html#deploy	
2.DRBD同期設定					
2-1	構築用の環境変数を設定する	●	●	【設定例】赤字の値は、ユーザ環境に合わせて設定します。 (本手順では前項「4.システム構成例」環境を例に記載) export ha1_name=ita-ha-sv01 ※ masterサーバのホスト名を指定 export ha2_name=ita-ha-sv02 ※ slaveサーバのホスト名を指定 export ha1_addr=192.168.1.11 ※ masterサーバのハートビート用IPアドレスを指定 export ha2_addr=192.168.1.12 ※ slaveサーバのハートビート用IPアドレスを指定 export virtual_ip_addr=10.1.1.10 ※ ITAアクセス用のVirtualIPアドレスを指定	本手順の途中で、再ログインする場合は、本コマンドを再実行します。
2-2	DRBDインストール用のelrepoリポジトリを追加する	●	●	yum -y install http://www.elrepo.org/elrepo-release-8.1-1.el8.elrepo.noarch.rpm	
2-3	DRBDをインストールする	●	●	yum -y install drbd kmod-drbd90	
2-4	※DNSサーバ利用による名前解決を行う場合は不要 HA構成ノードが互いに名前解決できるようにする	●	●	echo "\${ha1_addr} \${ha1_name}" >> /etc/hosts echo "\${ha2_addr} \${ha2_name}" >> /etc/hosts	
2-5	DRBD init スクリプトでシステム起動時のDRBDの動作を定義する	●	●	vi /etc/drbd.d/global_common.conf <pre> startup { # wfc-timeout degr-wfc-timeout outdated-wfc-timeout wait-after-sb wfc-timeout 5; ※startupセクションに"wfc-timeout"設定を追記 } </pre>	

構築手順

実行ユーザ / 接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
2-6	DBRDのsystemd設定ファイルから、バージョン9.0以降では存在しないコマンドの実行をコメントアウト	●	●	<pre>vi /usr/lib/systemd/system/drbd.service</pre> <p>[Service] (表示例) ~~~~~抜粋~~~~~ #ExecStart=/usr/sbin/drbdadm sh-b-pri all ※サブコマンド「sh-b-pri」をコメントアウト # disconnect and detach all resources ExecStop=/usr/sbin/drbdadm down all</p>	
2-7	/dev/vdc1を作成する	●	●	<pre>echo -e "\$nn\$np\$nn1\$nn\$nn+(/パーティションのサイズ)\$nnw" fdisk /dev/vdc</pre>	(パーティションのサイズ)は、数字+単位をK(KibiByte)、M(MebiByte)、G(GibiByte)指定する 例 5GibiByteの場合は、5G
2-8	/dev/vdc2を作成する ※DRBD同期用ボリュームの残り空き容量すべてとする場合	●	●	<pre>echo -e "\$nn\$np\$nn2\$nn\$nnw" fdisk /dev/vdc</pre>	
2-9	作成したパーティションのデータを「0」で埋めて、ファイルシステムを初期化する	●	●	<pre>dd if=/dev/zero of=/dev/vdc1 bs=1M count=1 dd if=/dev/zero of=/dev/vdc2 bs=1M count=1</pre>	
2-10	DRBD設定ファイルを作成する	●	●	<pre>cat > /etc/drbd.d/r0.res << DRBD resource r0 { device /dev/drbd0; disk /dev/vdc1; meta-disk internal; on \$ha1_name { address \$ha1_addr:7788; } on \$ha2_name { address \$ha2_addr:7788; } } DRBD</pre>	コマンドをまとめて実施する
				<pre>cat > /etc/drbd.d/r1.res << DRBD resource r1 { device /dev/drbd1; disk /dev/vdc2; meta-disk internal; on \$ha1_name { address \$ha1_addr:7789; } on \$ha2_name { address \$ha2_addr:7789; } } DRBD</pre>	コマンドをまとめて実施する
2-11	※firewalld使用時のみ実施 使用したポート宛の通信を許可する	●	●	<pre>firewall-cmd --add-port=7788/tcp --zone=public --permanent firewall-cmd --add-port=7789/tcp --zone=public --permanent firewall-cmd --reload</pre>	
2-12	DRBDリソースのメタデータを作成する	●	●	<pre>drbdadm create-md r0 drbdadm create-md r1</pre>	
2-13	DRBDサービスを起動する	●	●	<pre>systemctl start drbd</pre>	

構築手順

実行ユーザ/接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
2-14	master側のITA同期用マウントを設定する				
		masterからslaveにDRBD同期用デバイスの初期同期をする	●	drbdadm primary --force all	
		DRBD同期用デバイスをxfsでフォーマットする	●	mkfs -t xfs /dev/drbd0 mkfs -t xfs /dev/drbd1	
		ITA同期対象ファイルのマウント設定 (drbd0)			
		ITA同期用ディレクトリを作成し、DRBD同期用デバイスをマウントする	●	mkdir -p /mnt/(ITAのインストールパス) mount /dev/drbd0 /mnt/(ITAのインストールパス)	赤字は、環境に合わせて読み替えること
		ITA同期対象のディレクトリを配列変数に格納する	●	dirs=("/(ITAのインストールパス)/data_relay_storage/symphony" "/(ITAのインストールパス)/data_relay_storage/conductor" "/(ITAのインストールパス)/data_relay_storage/ansible_driver" "/(ITAのインストールパス)/ita_sessions" "/(ITAのインストールパス)/ita-root/temp" "/(ITAのインストールパス)/ita-root/uploadfiles" "/(ITAのインストールパス)/ita-root/webroot/uploadfiles" "/(ITAのインストールパス)/ita-root/webroot/menus/sheets" "/(ITAのインストールパス)/ita-root/webroot/menus/users" "/(ITAのインストールパス)/ita-root/webconfs/sheets" "/(ITAのインストールパス)/ita-root/webconfs/users")	コマンドをまとめて実施する 赤字は、環境に合わせて読み替えること
		ITA同期対象ディレクトリのデータをDRBD同期用デバイスに移出し、DRBD同期用デバイスをマウントしたディレクトリへのシンボリックリンクを作成する	●	for dir in "\${dirs[@]"; do ## directory回避 if [-d \$dir]; then mv \$dir{,.org} fi ## マウントしたディレクトリ直下にディレクトリ作成 if [! -d `dirname /mnt\${dir}`]; then mkdir -p `dirname /mnt\${dir}` fi ## マウントしたディレクトリ直下にITAデータをコピー if [! -d /mnt\${dir}]; then cp -pr \$dir.org /mnt\${dir}/ fi ## シンボリックリンクを作成 if [-d /mnt\${dir}]; then ln -s "/mnt/\${dir} \$dir" fi ## 回避ディレクトリを削除する場合はコメントイン # if [-d \$dir.org]; then # rm -rf \$dir.org # fi done	コマンドをまとめて実施する
		MaiaDBデータファイル用デバイス設定 (drbd1)			
		移出元のMaiaDBデータを退避する	●	cp -pr /var/lib/mysql /var/lib/mysql.org	
		DRBD同期用デバイスを /var/lib/mysql にマウントする	●	mount /dev/drbd1 /var/lib/mysql chown -R mysql:mysql /var/lib/mysql	
	マウント先にMaiaDBデータをコピー(同期)し、退避元データを削除する	●	rsync -a --delete /var/lib/mysql.org/ /var/lib/mysql/ rm -rf /var/lib/mysql.org		
	DRBD同期用デバイスをアンマウントする	●	umount /dev/drbd0 umount /dev/drbd1		
	DRBDリソースをセカンダリに降格する	●	drbdadm secondary all		

構築手順

実行ユーザ / 接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
2-15	slave側のITA同期用マウントを設定する				
	slaveからmasterにストレージの初期同期をする			● drbdadm primary --force all	
	DRBD同期用のパーティションをxfsでフォーマットする			● mkfs -t xfs /dev/drbd0 mkfs -t xfs /dev/drbd1	
	ITA同期対象ファイルのマウント設定 (drbd0)				
	ITA同期用ディレクトリを作成し、DRBD同期用デバイスをマウントする			● mkdir -p /mnt/(ITAのインストールパス) mount /dev/drbd0 /mnt/(ITAのインストールパス)	赤字は、環境に合わせて読み替えること
	ITA同期対象のディレクトリを配列変数に格納する			● dirs=("/(ITAのインストールパス)/data_relay_storage/symphony" "/(ITAのインストールパス)/data_relay_storage/conductor" "/(ITAのインストールパス)/data_relay_storage/ansible_driver" "/(ITAのインストールパス)/ita_sessions" "/(ITAのインストールパス)/ita-root/temp" "/(ITAのインストールパス)/ita-root/uploadfiles" "/(ITAのインストールパス)/ita-root/webroot/uploadfiles" "/(ITAのインストールパス)/ita-root/webroot/menus/sheets" "/(ITAのインストールパス)/ita-root/webroot/menus/users" "/(ITAのインストールパス)/ita-root/webconfs/sheets" "/(ITAのインストールパス)/ita-root/webconfs/users")	コマンドをまとめて実施する 赤字は、環境に合わせて読み替えること
	ITA同期対象ディレクトリのデータをDRBD同期用デバイスに移出し、DBRD同期用デバイスをマウントしたディレクトリへのシンボリックリンクを作成する			● for dir in "\${dirs[@]"; do ## directory退避 if [-d \$dir]; then mv \$dir{,.org} fi ## マウントしたディレクトリ直下にディレクトリ作成 if [! -d `dirname /mnt\${dir}`]; then mkdir -p `dirname /mnt\${dir}` fi ## マウントしたディレクトリ直下にITAデータをコピー if [! -d /mnt\${dir}]; then cp -pr \$dir.org /mnt\${dir}/ fi ## シンボリックリンクを作成 if [-d /mnt\${dir}]; then ln -s "/mnt" \${dir} \${dir} fi ## 退避ディレクトリを削除する場合はコメントイン # if [-d \$dir.org]; then # rm -rf \$dir.org # fi done	コマンドをまとめて実施する
MaiaDBデータファイル用デバイス設定 (drbd1)					
/var/lib/mysql をDRBD同期用デバイスにマウントする			● mount /dev/drbd1 /var/lib/mysql chown -R mysql:mysql /var/lib/mysql		
DRBD同期用デバイスをアンマウントする			● umount /dev/drbd0 umount /dev/drbd1		
リソースをセカンダリに降格する			● drbdadm secondary all		
2-16	DRBDサービスを停止する			● systemctl stop drbd	

構築手順

実行ユーザ/接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
3.MariaDB停止設定					
3-1	MariaDBサービスを停止する	●	●	systemctl disable mariadb systemctl stop mariadb	
4.ITAサービス停止設定					
4-1	各ITAサービス(ky.*)を停止する	●	●	systemctl stop ky_activatedirectory_roleuser_replication-workflow.service systemctl stop ky_ansible_execute-workflow.service systemctl stop ky_ansible_towermasterSync-workflow.service systemctl stop ky_bulk_excel-workflow.service systemctl stop ky_change_col_to_row.service systemctl stop ky_cmdbmenuanalysis-workflow.service systemctl stop ky_create_er-workflow.service systemctl stop ky_create_param_menu_execute.service systemctl stop ky_data_portability_execute-workflow.service systemctl stop ky_hostgroup_check_loop.service systemctl stop ky_hostgroup_split.service systemctl stop ky_legacy_role_valautostup-workflow.service systemctl stop ky_legacy_role_varsautolistup-workflow.service systemctl stop ky_legacy_valautostup-workflow.service systemctl stop ky_legacy_varsautolistup-workflow.service systemctl stop ky_mail.service systemctl stop ky_pioneer_valautostup-workflow.service systemctl stop ky_pioneer_varsautolistup-workflow.service systemctl stop ky_std_checkcondition-linklist.service systemctl stop ky_std_synchronize-Collector.service systemctl stop ky_std_synchronize-Conductor.service systemctl stop ky_std_synchronize-regularly2.service systemctl stop ky_std_synchronize-regularly.service systemctl stop ky_std_synchronize-regularly.service systemctl stop ky_terraform_execute-workflow.service systemctl stop ky_terraform_checkcondition-workflow.service systemctl stop ky_terraform_varsautolistup-workflow.service systemctl stop ky_terraform_valautosetup-workflow.service	ITAのバージョンによって、存在しないサービスがあります。その場合は、サービス停止不要です。

構築手順

実行ユーザ/接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
4-2	各ITAサービス(ky_*) 自動起動設定を無効にする			<ul style="list-style-type: none"> ● systemctl disable ky_active_directory_roleuser_replication-workflow.service ● systemctl disable ky_ansible_execute-workflow.service ● systemctl disable ky_ansible_towermasterSync-workflow.service ● systemctl disable ky_bulk_excel-workflow.service ● systemctl disable ky_change_col_to_row.service ● systemctl disable ky_cmdbmenuanalysis-workflow.service ● systemctl disable ky_create_er-workflow.service ● systemctl disable ky_create_param_menu_execute.service ● systemctl disable ky_data_portability_execute-workflow.service ● systemctl disable ky_hostgroup_check_loop.service ● systemctl disable ky_hostgroup_split.service ● systemctl disable ky_legacy_role_valautostup-workflow.service ● systemctl disable ky_legacy_role_varsautolistup-workflow.service ● systemctl disable ky_legacy_valautostup-workflow.service ● systemctl disable ky_legacy_varsautolistup-workflow.service ● systemctl disable ky_mail.service ● systemctl disable ky_pioneer_valautostup-workflow.service ● systemctl disable ky_pioneer_varsautolistup-workflow.service ● systemctl disable ky_std_checkcondition-linklist.service ● systemctl disable ky_std_synchronize-Collector.service ● systemctl disable ky_std_synchronize-Conductor.service ● systemctl disable ky_std_synchronize-regularly2.service ● systemctl disable ky_std_synchronize-regularly.service ● systemctl disable ky_std_synchronize-symphony.service ● systemctl disable ky_terraform_execute-workflow.service ● systemctl disable ky_terraform_checkcondition-workflow.service ● systemctl disable ky_terraform_varsautolistup-workflow.service ● systemctl disable ky_terraform_valautosetup-workflow.service 	ITAのバージョンによって、存在しないサービスがあります。その場合は、設定の無効は不要です。
5.Apacheリソース設定					
5-1	Apacheにヘルスチェック用のステータスレポートを設定する			<ul style="list-style-type: none"> ● cat > /etc/httpd/conf.d/server_status.conf << STAT ExtendedStatus On <Location /server-status> SetHandler server-status Order deny,allow Deny from all Allow from localhost </Location> STAT 	コマンドをまとめて実施する
5-2	Apacheサービス停止・無効化する			<ul style="list-style-type: none"> ● systemctl disable httpd ● systemctl stop httpd 	
5-3	httpsサーバ証明書/秘密鍵をmasterからslaveへコピーする			<ul style="list-style-type: none"> ● scp /etc/pki/tls/certs/(httpsサーバ証明書).crt \${ha2_addr}:/etc/pki/tls/certs/ ● scp /etc/pki/tls/certs/(httpsサーバ秘密鍵名).key \${ha2_addr}:/etc/pki/tls/certs/ 	httpsサーバ証明書/秘密鍵は、ITAサーバインストール時に作成した、ファイル名に読み替えること

構築手順

実行ユーザ / 接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
6.Pacemaker設定					
6-1	HAソフト各種をインストールする	●	●	yum --enablerepo=rhel-8-for-x86_64-highavailability-rpms install pacemaker pcs drbd-pacemaker NetworkManager NetworkManager-config-server systemctl restart NetworkManager	
6-2	※プロキシ環境下の場合に実施 主機/副機のIPをno_proxyに追加する	●	●	sed -i 's/^export NO_PROXY/export no_proxy=%\$no_proxy,\$ha1_name,\$ha2_name\nexport NO_PROXY/' /etc/profile.d/proxy.sh sed -i 's/^setenv NO_PROXY/setenv no_proxy=%\$no_proxy,\$ha1_name,\$ha2_name\nsetenv NO_PROXY/' /etc/profile.d/proxy.csh source /etc/profile.d/proxy.sh source /etc/profile.d/proxy.csh	※一度、テキストに張り付けてから実行
6-3	※firewalld使用時のみ実施 firewalld設定を追加する	●	●	firewall-cmd --add-service=high-availability --permanent firewall-cmd --reload	
6-4	corosync.serviceの設定を変更する	●	●	cp -p /usr/lib/systemd/system/corosync.service /etc/systemd/system/ sed -i 's/^#Restart=on.*\/Restart=on-failure/' /etc/systemd/system/corosync.service sed -i 's/^#RestartSec=.*\/RestartSec=70/' /etc/systemd/system/corosync.service	
6-5	Pacemakerの内部プロセスが異常になった場合もノード故障として取り扱う	●	●	sed -i 's/^# PCMK_fail_fast.*\/PCMK_fail_fast=yes/' /etc/sysconfig/pacemaker	
6-6	pacemaker.serviceの設定を変更する	●	●	cp -p /usr/lib/systemd/system/pacemaker.service /etc/systemd/system sed -i "s/^# ExecStopPost=.*\/bin\/sh.*\/ExecStopPost=.*\/bin\/sh -c 'pidof crmd % killall -TERM corosync'/" /etc/systemd/system/pacemaker.service	
6-7	クラスター認証設定ファイルを生成及び配置する	●	●	corosync-keygen -l scp -p /etc/corosync/authkey root@[ha2_addr]:/etc/corosync/authkey	
6-8	pcsの管理アカウントとなるユーザー「hacluster」のパスワードを設定する	●	●	passwd hacluster 新しいパスワード:xxxxxx 新しいパスワードを再入力してください:xxxxxx passwd: すべての認証トークンが正しく更新できました。	赤字は、環境に合わせて読み替えること master、slaveの両方にて同一のパスワードを設定すること
6-9	pcsdサービスを起動し、システムの起動時にpcsdが有効になるよう設定する	●	●	systemctl enable pcsd systemctl start pcsd systemctl status pcsd	

構築手順

実行ユーザ/接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
6-10	クラスターを認証する	●		pcs host auth \${ha1_name} \${ha2_name} Username: hacluster Password: passwd haclusterで設定したパスワード	赤字は、環境に合わせて読み替えること
6-11	クラスターをセットアップする	●		pcs cluster setup hacluster --start \${ha1_name} addr=\${ha1_addr} \${ha2_name} addr=\${ha2_addr}	
6-12	STONITH (Shoot The Other Node In The Head) オプションを無効にする	●		pcs property set stonith-enabled=false	
6-13	クラスターが必要最低限の数に達していない場合でも特別な動作は行わず全ノードの管理を続	●		pcs property set no-quorum-policy=ignore	
6-14	属性待ち時間を設定する	●		pcs property set transition-delay="0s"	
6-15	自動フェイルバックを無効にする	●		pcs resource defaults resource-stickiness="INFINITY" migration-threshold="1"	
6-16	クラスターを自動起動するように設定をする	●		pcs cluster enable --all	
6-17	vipリソース設定をする	●		pcs resource create virtual_ip ocf:heartbeat:IPAddr2 ip=\${virtual_ip_addr} cidr_netmask=24 ¥ op monitor interval=10s timeout=30s ¥ op start interval=0s timeout=20s ¥ op stop interval=0s timeout=20s	
6-18	DRBDデバイスr0をリソースとして登録する	●		pcs resource create drbd_r0 ocf:linbit:drbd drbd_resource=r0 ¥ op monitor interval=10s role=Master ¥ op monitor interval=30s role=Slave ¥ op notify interval=0s timeout=90s ¥ op promote interval=0s timeout=90s ¥ op reload interval=0s timeout=30s ¥ op start interval=0s timeout=240s ¥ op stop interval=0s timeout=100s	
6-19	DRBDデバイスr1をリソースとして登録する	●		pcs resource create drbd_r1 ocf:linbit:drbd drbd_resource=r1 ¥ op monitor interval=10s role=Master ¥ op monitor interval=30s role=Slave ¥ op notify interval=0s timeout=90s ¥ op promote interval=0s timeout=90s ¥ op reload interval=0s timeout=30s ¥ op start interval=0s timeout=240s ¥ op stop interval=0s timeout=100s	
6-20	DRBDブロックデバイスのリソース設定を行う	●		pcs resource create fs_exastro ocf:heartbeat:Filesystem device=/dev/drbd0 directory=/mnt/exastro fstype=xfs pcs resource create fs_mariadb ocf:heartbeat:Filesystem device=/dev/drbd1 directory=/var/lib/mysql fstype=xfs	
6-21	httpdリソース設定をする	●		pcs resource create httpd systemd:httpd ¥ op monitor interval=60s timeout=100s ¥ op start interval=0s timeout=100s ¥ op stop interval=0s timeout=100s	

構築手順

実行ユーザ / 接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
6-22	各ITAサービス(ky_*)のリソース設定をする	●		<pre> pcs resource create ky_activatedirectory_roleuser_replication-workflow systemd:ky_activatedirectory_roleuser_replication-workflow ¥ op monitor interval=60s timeout=100s ¥ op start interval=0s timeout=100s ¥ op stop interval=0s timeout=100s pcs resource create ky_ansible_execute-workflow systemd:ky_ansible_execute-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_ansible_towermasterSync-workflow systemd:ky_ansible_towermasterSync-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_bulk_excel-workflow systemd:ky_bulk_excel-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_change_col_to_row systemd:ky_change_col_to_row ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_cmdbmenuanalysis-workflow systemd:ky_cmdbmenuanalysis-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_create_er-workflow systemd:ky_create_er-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_create_param_menu_execute systemd:ky_create_param_menu_execute ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_data_portability_execute-workflow systemd:ky_data_portability_execute-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s </pre>	

構築手順

実行ユーザ / 接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
6-22	各ITAサービス(ky_*)のリソース設定をする	●		<pre> pcs resource create ky_hostgroup_check_loop systemd:ky_hostgroup_check_loop ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_hostgroup_split systemd:ky_hostgroup_split ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_legacy_role_valautostup-workflow systemd:ky_legacy_role_valautostup-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_legacy_role_varsautolistup-workflow systemd:ky_legacy_role_varsautolistup-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_legacy_valautostup-workflow systemd:ky_legacy_valautostup-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_legacy_varsautolistup-workflow systemd:ky_legacy_varsautolistup-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_pioneer_valautostup-workflow systemd:ky_pioneer_valautostup-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_pioneer_varsautolistup-workflow systemd:ky_pioneer_varsautolistup-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_std_checkcondition-linklist systemd:ky_std_checkcondition-linklist ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s </pre>	

構築手順

実行ユーザ / 接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
6-22	各ITAサービス(ky_*)のリソース設定をする	●		<pre> pcs resource create ky_std_synchronize-Conductor systemd:ky_std_synchronize-Conductor ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_std_synchronize-regularly2 systemd:ky_std_synchronize-regularly2 ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_std_synchronize-regularly systemd:ky_std_synchronize-regularly ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_std_synchronize-symphony systemd:ky_std_synchronize-symphony ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_terraform_execute-workflow systemd:ky_terraform_execute-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_terraform_checkcondition-workflow systemd:ky_terraform_checkcondition-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_terraform_varsautolistup-workflow systemd:ky_terraform_varsautolistup-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s pcs resource create ky_terraform_valautosetup-workflow systemd:ky_terraform_valautosetup-workflow ¥ op monitor interval=30s timeout=60s ¥ op start interval=0s timeout=60s ¥ op stop interval=0s timeout=60s </pre>	
6-23	mariadbリソース設定をする	●		<pre> pcs resource create mariadb systemd:mariadb ¥ op monitor interval=60s timeout=100s ¥ op start interval=0s timeout=100s ¥ op stop interval=0s timeout=100s </pre>	

構築手順

実行ユーザ/接続条件

rootユーザで実行すること
rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
6-24	fsリソースグループ設定をする	●		pcs resource group add fs_mount_group fs_exastro fs_mariadb	
6-25	vip、httpdリソースグループ設定をする	●		pcs resource group add exastro virtual_ip httpd	
6-26	各ITAサービス(ky_*)リソースのグループ設定をする	●		pcs resource group add ky_services ky_activedirectory_roleuser_replication-workflow ky_ansiible_execute-workflow ky_ansiible_towermasterSync-workflow ky_bulk_excel-workflow ky_change_col_to_row ky_cmdbmenuanalysis-workflow ky_create_er-workflow ky_create_param_menu_execute ky_data_portability_execute-workflow ky_hostgroup_check_loop ky_hostgroup_split ky_legacy_role_valautostup-workflow ky_legacy_role_varsautolistup-workflow ky_legacy_valautostup-workflow ky_legacy_varsautolistup-workflow ky_pioneer_valautostup-workflow ky_pioneer_varsautolistup-workflow ky_std_checkcondition-linklist ky_std_synchronize-Conductor ky_std_synchronize-regularly2 ky_std_synchronize-regularly ky_std_synchronize-symphony ky_terraform_execute-workflow ky_terraform_checkcondition-workflow ky_terraform_varsautolistup-workflow ky_terraform_valautostup-	
6-27	追加したDRBDリソースをMaster/Slave形式とし、2ノードで立ち上げ、うち1ノードがMaster(DRBDにおけるPriamry)となるように指定する	●		pcs resource promotable drbd_r0 master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true pcs resource promotable drbd_r1 master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true	
6-28	fsリソースとms_drbdが同じノードで起動する制約を付加する	●		pcs constraint colocation add fs_exastro with drbd_r0-clone INFINITY with-rsc-role=Master pcs constraint colocation add fs_mariadb with drbd_r1-clone INFINITY with-rsc-role=Master	
6-29	リソースms_drbd⇒fs_exastroの順で起動する制約を付加する	●		pcs constraint order promote drbd_r0-clone then start fs_exastro pcs constraint order promote drbd_r1-clone then start fs_mariadb	
6-30	これまでに作成したリソースを指定し、リソースグループ起動/停止の順序を指定する	●		pcs constraint order set fs_mount_group mariadb ky_services exastro	
6-31	これまでに設定したリソースが同ノードで起動する制限を指定する	●		pcs constraint colocation set fs_mount_group mariadb ky_services exastro	
6-32	リソースの再読み込みを実行する	●		pcs resource cleanup	
6-33	設定ファイルを再読み込みする	●	●	systemctl daemon-reload	
6-34	HAソフトのサービスを開始する	●	●	systemctl enable corosync systemctl enable pacemaker systemctl start corosync systemctl start pacemaker	

構築手順

実行ユーザ / 接続条件
 rootユーザで実行すること
 rootユーザがsshログイン可能であること

No.	作業項目	サーバ		設定内容 (※master、slaveの両方が対象の場合は、並行して設定してください。)	備考
		master	slave		
6-35	クラスタの正常起動を確認する	●		<pre>pcs status (表示例) ~~~~~ 抜粋 ~~~~~ 2 nodes configured ←左記の表示であること 32 resource instances configured Online: [ita-ha-sv01 ita-ha-sv02] ←master,slaveがOnlineであること Active resources: mariadb (systemd:mariadb): Started ita-ha-sv01 ←masterがStartedであること Resource Group: fs_mount_group fs_mysql (ocf:heartbeat:Filesystem): Started ita-ha-sv01 fs_httpd (ocf:heartbeat:Filesystem): Started ita-ha-sv01 Resource Group: ky_services kv xxx (systemd:ky xxx): Started ita-ha-sv01 Resource Group: exastro virtual_ip (ocf:heartbeat:IPAddr2): Started ita-ha-sv01 httpd (systemd:httpd): Started ita-ha-sv01 ~~~~~ 抜粋 ~~~~~</pre>	赤字は、環境に合わせて読み替える
6-36	ITAの接続を確認する	●		<p>以下のURLにアクセスし、正常にITAのログイン画面が表示されることを確認する https://10.1.1.10</p>	赤字は、環境に合わせて読み替える
作業終了					