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

ITA ver1.7

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

第1.1版

Exastro developer

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て、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、以下の環境で構築することを前提に作成しています。

主要ソフトウェア(ITAを除く)	用途	検証バージョン
CentOS	OS	7.8.2003
Pacemaker	クラスタ制御	1.1.23-1.el7_9.1
Corosync	ハートビート	2.4.5
crmsh	Pacemakerコマンド	3.0.1
DRBD	レプリケーション	8.4.11-1

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。

はじめに

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。

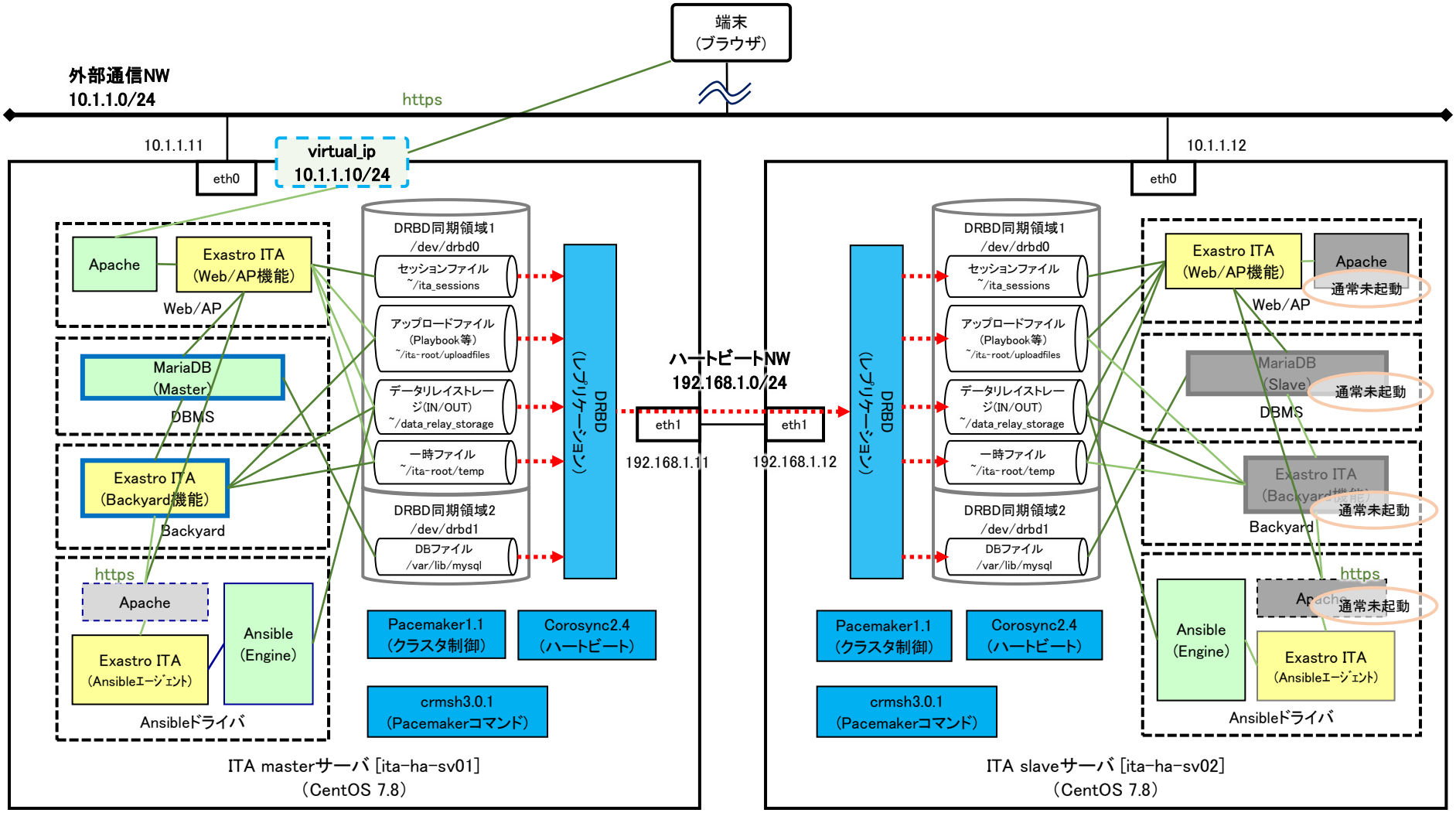
Pacemaker
HAクラスタが提供するサービスを監視し、サービスを提供していたサーバがダウンした場合に他のサーバでサービスを継続させるなどして、クラスタ全体の可用性を保ちます。

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アドレスを持つが、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
corosync	5405/udp ※1

※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

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

・ 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ます。

ita_answers.txtの機能設定例
ita_base:yes
material:no
createparam:yes
hostgroup:yes
ansible_driver:yes
cobbler_driver:no
terraform_driver:no

    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
    - 資材「ita_resource.conf」のPrimitive、Group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「2. .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	elrepoリポジトリを追加する	●	●	rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm	
2-3	DRBDをインストールする	●	●	yum -y install kmod-drbd84	
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	/dev/vdc1を作成する	●	●	echo -e "¥nn¥np¥n1¥n¥n+(パーティションのサイズ)¥nw" | fdisk /dev/vdc	(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
2-6	/dev/vdc2を作成する ※残り全部とする場合	●	●	echo -e "¥nn¥np¥n2¥n¥n¥nw" | fdisk /dev/vdc	
2-7	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	●	●	dd if=/dev/zero of=/dev/vdc1 bs=1M count=1 dd if=/dev/zero of=/dev/vdc2 bs=1M count=1	

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		

2-8	DRBD設定ファイルを作成する	●	●	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
					コマンドをまとめて実施する

				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
					コマンドをまとめて実施する

2-9	※firewalld使用時のみ実施 使用したポート宛の通信を許可する	●	●	firewall-cmd --add-port=7788/tcp --zone=public --permanent firewall-cmd --add-port=7789/tcp --zone=public --permanent firewall-cmd --reload	
2-10	DRBDリソースのメタデータを作成する	●	●	drbdadm create-md r0 drbdadm create-md r1	
2-11	DRBDサービスを起動する	●	●	systemctl start 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		
2-12	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同期用デバイスにマウントする	●		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、
	ITAインストールパス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）				
	移出元のMaiaDBデータを退避する	●		cp -pr /var/lib/mysql /var/lib/mysql,.org	
	/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	
	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		
2-13	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同期用デバイスにマウントする		●	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、
	ITAインストールパス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同期用のパーティション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	
2-14	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	MaiaDB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	ky-serviceを停止する	●	●	systemctl stop ky_activedirectory_roleuser_replication-workflow.service systemctl stop ky_ansible_execute-workflow.service systemctl stop ky_ansible_towermasterSync-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-symphony.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。
4-2	ky-service 自動起動設定を無効にする	●	●	systemctl disable ky_activedirectory_roleuser_replication-workflow.service systemctl disable ky_ansible_execute-workflow.service systemctl disable ky_ansible_towermasterSync-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	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。

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		
5.Apacheリソース設定					
5-1	apacheのserver statusを作成する	●	●	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	●	●	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	●	●	scp /etc/pki/tls/certs/(httpsサーバ証明書名).crt ${ha2 addr}:/etc/pki/tls/certs/ scp /etc/pki/tls/certs/(httpsサーバ秘密鍵名).key ${ha2_addr}:/etc/pki/tls/certs/	httpsサーバ証明書／秘密鍵は、ITAサーバインストール時に作成した、ファイル名に読み替えること
6.Pacemaker設定					
6-1	HAソフト各種をインストールする	●	●	yum -y install pacemaker NetworkManager NetworkManager-config-server systemctl restart NetworkManager	
6-2	crmshで要求されるpython-parallaxをインストールする	●	●	yum install -y http://repo.okay.com.mx/centos/7/x86_64/release/okay-release-1-1.noarch.rpm yum install -y http://repo.okay.com.mx/centos/7/x86_64/release/python-parallax-1.0.0a1-7.1.noarch.rpm	
6-3	crmshをインストールする	●	●	cat > /etc/yum.repos.d/crmsh.repo << EOF [network_ha-clustering_Stable] name=Stable High Availability/Clustering packages (CentOS_CentOS-7) type=rpm-md baseurl=https://download.opensuse.org/repositories/network:/ha-clustering:/Stable/CentOS_CentOS-7/ gpgcheck=1 gpgkey=https://download.opensuse.org/repositories/network:/ha-clustering:/Stable/CentOS_CentOS-7/repodata/repomd.xml.key enabled=1 EOF	コマンドをまとめて実施する
				yum install -y crmsh	
6-4	master側のcrmshがクラスタ構成ノードを制御できるように、ノード間でssh公開鍵認証設定をする	●		ssh-keygen -f /root/.ssh/id_rsa -N '' ssh-copy-id -oStrictHostKeyChecking=no -i /root/.ssh/id_rsa root@${ha2_addr}	
6-5	slave側のcrmshがクラスタ構成ノードを制御できるように、ノード間でssh公開鍵認証設定をする		●	ssh-keygen -f /root/.ssh/id_rsa -N '' ssh-copy-id -oStrictHostKeyChecking=no -i /root/.ssh/id_rsa root@${ha1_addr}	
6-6	※プロキシ環境下の場合に実施 主機/副機のIPをno_proxyに追記する	●	●	sed -i 's/^setenv¥ NO_PROXY/setenv no_proxy=¥$no_proxy,'$ha1_addr','$ha2_addr'¥nsetenv NO_PROXY/' /etc/profile.d/proxy.csh sed -i 's/^export¥ NO_PROXY/export no_proxy=¥$no_proxy,'$ha1_addr','$ha2_addr'¥nexport NO_PROXY/' /etc/profile.d/proxy.sh	
6-7	※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	
6-8	corosync設定をする	●	●	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	

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		
6-9	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	
6-10	corosync設定をする	●	●	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	
6-11	資材ファイルをmaster側にアップロードする	●		マニュアルに同伴されている資材ファイルを転送する 対象資材 → 転送先 corosync.conf → /etc/corosync/	
6-12	/etc/corosync/corosync.confを編集する	●		vi /etc/corosync/corosync.conf ※ハートビートNWでデータ同期を行う場合 bindnetaddr: 192.168.1.11 ※ ハートビートNWを指定 ring0 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 ring0_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 ※外部通信NWでデータ同期を行う場合 bindnetaddr: 10.1.1.10 ※ 外部通信NWを指定 ring0_addr: 10.1.1.11 ※ masterサーバの外部通信用IPアドレスを指定 ring0_addr: 10.1.1.12 ※ slaveサーバの外部通信用IPアドレスを指定 ※corosyncでマルチキャストアドレスを使用する場合は、以下の設定を変更します。 ① mcastaddr行をコメントイン mcastaddr: 239.255.1.1 ②transport: udpuをコメントアウト #transport: udpu	赤字は、環境に合わせて定義する
6-13	資材ファイルをslave側にコピーする	●		scp -p /etc/corosync/corosync.conf root@${ha2_addr}:/etc/corosync/corosync.conf	
6-14	クラスタ認証設定ファイルの生成及び配置する	●		corosync-keygen -l scp -p /etc/corosync/authkey root@${ha2_addr}:/etc/corosync/authkey	
6-15	設定ファイルを再読込する	●	●	systemctl daemon-reload	
6-16	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		
6-17	クラスタの正常起動を確認する	●		crm_mon -A (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: Resource Group: exastro fs_mysql (ocf::heartbeat:Filesystem): Started ita-ha-sv0 ←masterがStartedであること fs_httpd (ocf::heartbeat:Filesystem): Started ita-ha-sv01 mariadb (systemd:mariadb): Started ita-ha-sv01 virtual_ip (ocf::heartbeat:IPaddr2): Started ita-ha-sv01 httpd (systemd:httpd): Started ita-ha-sv01 ～～～～～～～～～～～～～～～～～～～～～～抜粋～～～～～～～～～～～～～～～～～～～～～～	赤字は、環境に合わせて読み替える
6-18	資材ファイルをmaster側にアップロードする	●		マニュアルに同伴されている資材ファイルを転送する 対象資材　転送先 ita_resource.crm　→　/任意のディレクトリ/	赤字は、環境に合わせて読み替える
6-19	crm設定ファイルをmaster側のノードへ配置する	●		sed -i 's/virtual¥_ip¥_addr/'$virtual_ip_addr'/' /任意のディレクトリ/ita_resource.crm	赤字は、環境に合わせて読み替える
6-20	crm設定ファイルを反映させる	●		crm configure load update /任意のディレクトリ/ita_resource.crm	赤字は、環境に合わせて読み替える
6-21	ITAの接続を確認する	●		以下のURLより、ログイン画面にアクセスする https://10.1.1.10	赤字は、環境に合わせて読み替える
作業終了					

【参考】corosync.confのsample設定

※ハートビートNWでデータ同期(ユニキャスト通信)を行う場合の設定例

```
totem {
    version: 2
    crypto_cipher: aes128
    crypto_hash: sha256
    transport: udpu
    interface {
        ringnumber: 0
        # 各ノードが属するNWアドレスを設定します。環境に合わせて書き替えてください。
        bindnetaddr: 192.168.1.0
        #mcastaddr: 239.255.1.1
        mcastport: 5405
        ttl: 1
    }
}

service {
    name: pacemaker
    ver: 0
    use_mgmtd: yes
}

logging {
    fileline: off
    to_stderr: no
    to_logfile: yes
    logfile: /var/log/cluster/corosync.log
    to_syslog: yes
    debug: off
    timestamp: on
    logger_subsys {
        subsys: QUORUM
        debug: off
    }
}

nodelist {
    # クラスタに参加するホストのIPアドレスです。環境に合わせて書き替えてください。
    node {
        ring0_addr: 192.168.1.11
        nodeid: 1
    }
    # クラスタに参加するホストのIPアドレスです。環境に合わせて書き替えてください。
    node {
        ring0_addr: 192.168.1.12
        nodeid: 2
    }
}
quorum {
    two_node: 1
    expected_votes: 2
    provider: corosync_votequorum
}
```

【参考】ita_resource.confのsample設定

```
primitive virtual_ip IPaddr2 ¥
    params cidr_netmask=24 ip=virtual_ip_addr ¥
    op monitor interval=10s timeout=20s ¥
    op start interval=0s timeout=20s ¥
    op stop interval=0s timeout=20s
primitive drbd_r0 ocf:linbit:drbd ¥
    params drbd_resource=r0 ¥
    op demote interval=0s timeout=90 ¥
    op monitor interval=10s role=Master ¥
    op monitor interval=30s role=Slave ¥
    op notify interval=0s timeout=90 ¥
    op promote interval=0s timeout=90 ¥
    op reload interval=0s timeout=30 ¥
    op start interval=0s timeout=240 ¥
    op stop interval=0s timeout=100
primitive drbd_r1 ocf:linbit:drbd ¥
    params drbd_resource=r1 ¥
    op demote interval=0s timeout=90 ¥
    op monitor interval=10s role=Master ¥
    op monitor interval=30s role=Slave ¥
    op notify interval=0s timeout=90 ¥
    op promote interval=0s timeout=90 ¥
    op reload interval=0s timeout=30 ¥
    op start interval=0s timeout=240 ¥
    op stop interval=0s timeout=100
primitive fs_httpd Filesystem ¥
    params device="/dev/drbd0" directory="/mnt/exastro" fstype=xfs ¥
    op monitor interval=20s timeout=40s ¥
    op notify interval=0s timeout=60s ¥
    op start interval=0s timeout=60s ¥
    op stop interval=0s timeout=60s
primitive fs_mysql Filesystem ¥
    params device="/dev/drbd1" directory="/var/lib/mysql" fstype=xfs ¥
    op monitor interval=20s timeout=40s ¥
    op notify interval=0s timeout=60s ¥
    op start interval=0s timeout=60s ¥
    op stop interval=0s timeout=60s
primitive httpd systemd:httpd ¥
    op monitor interval=60 timeout=100 ¥
    op start interval=0s timeout=100 ¥
    op stop interval=0s timeout=100
primitive ky_activedirectory_roleuser_replication-workflow
systemd:ky_activedirectory_roleuser_replication-workflow ¥
    op monitor interval=60 timeout=100 ¥
    op start interval=0s timeout=100 ¥
    op stop interval=0s timeout=100
primitive ky_ansible_execute-workflow systemd:ky_ansible_execute-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_ansible_towermasterSync-workflow systemd:ky_ansible_towermasterSync-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_change_col_to_row systemd:ky_change_col_to_row ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
```

```
primitive ky_cmdbmenuanalysis-workflow systemd:ky_cmdbmenuanalysis-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_create_er-workflow systemd:ky_create_er-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_create_param_menu_execute systemd:ky_create_param_menu_execute ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_data_portability_execute-workflow systemd:ky_data_portability_execute-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_hostgroup_check_loop systemd:ky_hostgroup_check_loop ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_hostgroup_split systemd:ky_hostgroup_split ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_legacy_role_valautostup-workflow systemd:ky_legacy_role_valautostup-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_legacy_role_varsautolistup-workflow systemd:ky_legacy_role_varsautolistup-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_legacy_valautostup-workflow systemd:ky_legacy_valautostup-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_legacy_varsautolistup-workflow systemd:ky_legacy_varsautolistup-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_pioneer_valautostup-workflow systemd:ky_pioneer_valautostup-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_pioneer_varsautolistup-workflow systemd:ky_pioneer_varsautolistup-workflow ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_std_checkcondition-linklist systemd:ky_std_checkcondition-linklist ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_std_synchronize-Conductor systemd:ky_std_synchronize-Conductor ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_std_synchronize-regularly systemd:ky_std_synchronize-regularly ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
```

```
primitive ky_std_synchronize-regularly2 systemd:ky_std_synchronize-regularly2 ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive ky_std_synchronize-symphony systemd:ky_std_synchronize-symphony ¥
    op monitor interval=30 timeout=60 ¥
    op start interval=0s timeout=60 ¥
    op stop interval=0s timeout=60
primitive mariadb systemd:mariadb ¥
    op monitor interval=60 timeout=100 ¥
    op start interval=0s timeout=100 ¥
    op stop interval=0s timeout=100
group exastro fs_mysql fs_httpd mariadb virtual_ip httpd
group ky_services ky_activedirectory_roleuser_replication-workflow ky_ansible_execute-workflow
ky_ansible_towermasterSync-workflow ky_change_col_to_row ky_cmdbmenuanalysis-workflow
ky_create_param_menu_execute ky_data_portability_execute-workflow ky_hostgroup_check_loop
ky_hostgroup_make_var ky_hostgroup_regist_var_legacy ky_hostgroup_regist_var_legacy_role
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-regularly ky_std_synchronize-regularly2 ky_std_synchronize-symphony
ms ms_drbd_r0 drbd_r0 ¥
    meta master-node-max=1 clone-max=2 notify=true master-max=1 clone-node-max=1
ms ms_drbd_r1 drbd_r1 ¥
    meta master-node-max=1 clone-max=2 notify=true master-max=1 clone-node-max=1
colocation colocation-exastro-ms_drbd_r0-INFINITY inf: exastro ms_drbd_r0:Master
colocation colocation-exastro-ms_drbd_r1-INFINITY inf: exastro ms_drbd_r1:Master
colocation colocation_set_eoks inf: _rsc_set_ exastro ky_services
order order-ms_drbd_r1-exastro-mandatory ms_drbd_r1:promote exastro:start
order order_set_eoks _rsc_set_ exastro ky_services
property cib-bootstrap-options: ¥
    stonith-enabled=false ¥
    no-quorum-policy=ignore
```