



# **IT Automation**

## **All In One HA Synchronized Configuration**

### **Install Manual**

ITA ver1.7

※In this Document, "Exastro IT Automation" will be written as "ITA".

Ver 1.1

Exastro developer

# Introduction

## 1. About this Document

- This document describes the information necessary to configure an HA cluster by combining two ITA all-in-one installed servers and OSS products. By turning the ITA into an HA Cluster, users can construct an ITA Server with more reliability and availability than what would be possible with a single Linux server.
- HA clusters configured with the use of this document will have the following functions.
  - The HA Cluster is constructed on 2 server systems. They can be either physical servers or vertical servers.
  - One of the server will have an ITA Server running and will have ITA Web/AP, Backyard, DBMS, Ansible Driver and other ITA Functions
  - If the Server that provides the service being used gets turned off, the other server will start ITA a minute later and will continue the service. The database data does not get deleted if this happens.
- Please see the diagram in "4. System construction example" for Deploy example. Make sure to change the IP Addresss, Host Name , Directory name and File name to match the users environment.
- In this document, an environment with the following will be constructed.

Main software (e.x ITA)	Application	Vertification Version
CentOS	OS	7.8.2003
Pacemaker	Cluster control	1.1.23-1.el7_9.1
Corosync	Heartbeat	2.4.5
crmsh	Pacemaker Command	3.0.1
DRBD	Replication	8.4.11-1

## 2. Scope of responsibility

- This document is intended to provide reference information on precautions and configuration examples for clustering ITA servers, and does not guarantee the operation of said information.

## 3. Linux-HA Cluster stack

- Linux-HA Cluster stack is a software that aims to improve the server systems availability and prevent data loss from storage failure. It is constructed from the following 3 softwares and their respective packages.

### DRBD (Distributed Replicated Block Device)

Writes the same data to both of the server's storage (Partition or Logical volume) in real time (Real-time replication)

Hence, if one of the servers or the storage itself fails, the service can be continued on a normally functioning server without losing data.

### Corosync

A software that mutually monitors the normal operation of the servers that makes up the HA cluster.

### Pacemaker

Monitors the services provided by the HA Cluster and moves the service to another server if the first server shuts down.

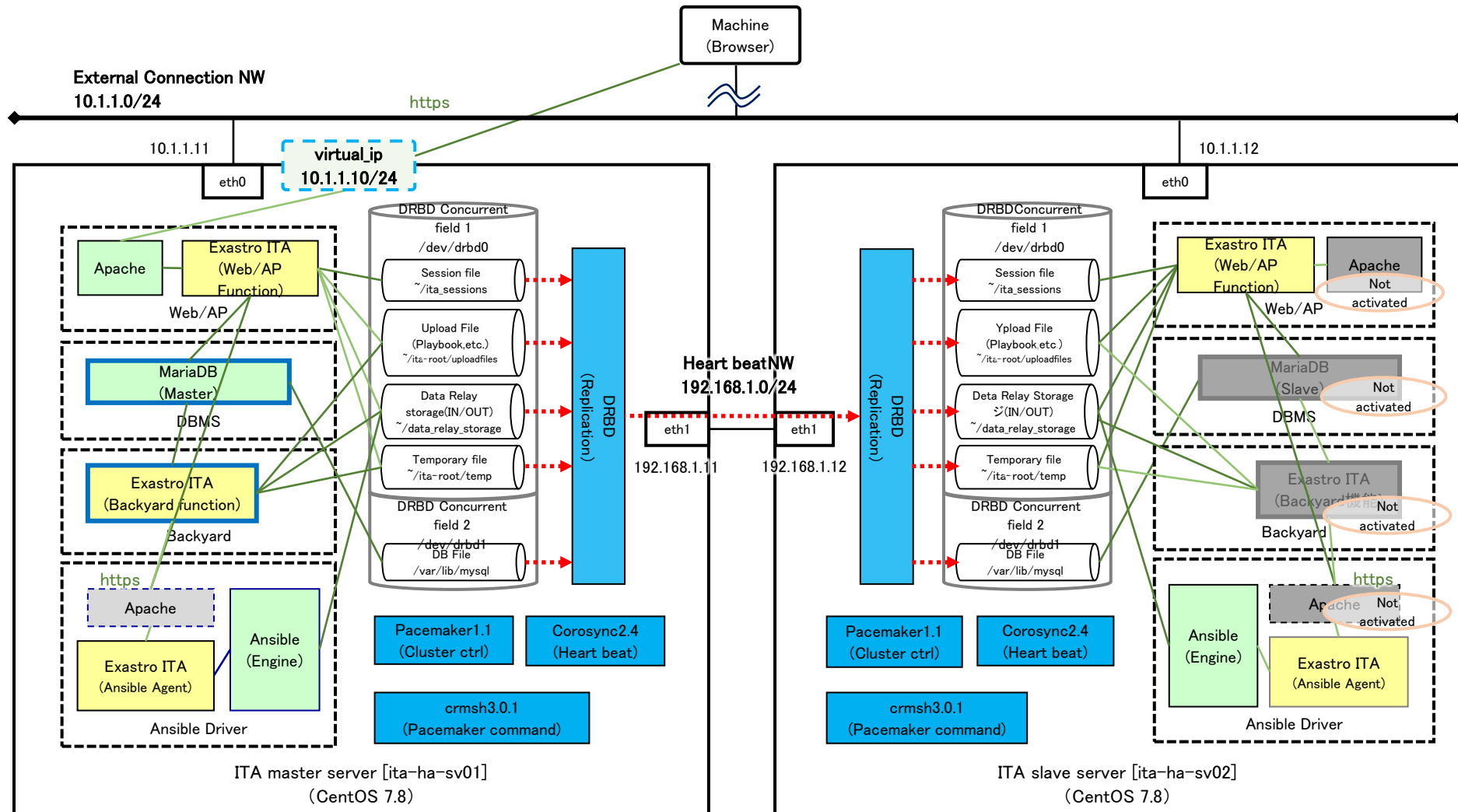
Reference URL:

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

# Introduction

## 4. System construction e)

- The disk and network configurations of the two servers that make up the ITA cluster, as well as the related programs, are shown in the figure below.



# Introduction

## ▪ Network Requirements

- Normally, each network interface has their own inherent IP Address, but in order to create an ITA Server HA Cluster, an additional IP Address (Virtual IP Address) will be needed. The virtual IP address will automatically be assigned to the master server by the Pacemaker after the cluster is constructed.

## ▪ Network settings used in this document

NetworkIF	Application	Master Server	Slave server
Virtual IP addr.	ITA Connection	10.1.1.10/24	
eth0	Extrenal Connection NW	10.1.1.11/24	10.1.1.12/24
eth1	Internal Connection NW	192.168.1.11/24	192.168.1.12/24

- The following communication ports must be available from the client to the server.

Connection type	Port number
http	80/tcp
https	443/tcp

- The cluster configuration servers must be configured on the same network, and the following communication ports must be available.

Connection type	Port Number
ssh(scp)	22/tcp
mariadb	3306/tcp
drbd	7788,7789/tcp ※1
corosync	5405/udp ※1

※1 The port number on the left can be changed from the construction procedure settings.

## ▪ Disc requirements

- Since the ITA Data is linked to an HA construction server different from the system disc, you must prepare a volume for DRBD synchronization and create a partition. This document describes the procedure for creating partitions and virtual block devices using “/dev/vdc” that has been created and attached in advance.
- The required volume size will vary depending on the ITA operation method and the number of tasks, so check the actual size and prepare a sizing with extra space.
- A minimum of 10GB is recommended. Be sure to prepare volumes of the same size for each of the servers that make up the HA cluster.
- DRBD Synchronization Partition construction example.

Partition	Virtual Block device name	Application	Size
/dev/vdc1	/dev/drbd0	ITA Synchronization file storage	5GB
/dev/vdc2	/dev/drbd1	MariiaDB Data file storage	5GB



# Introduction

## 5. Operation conditions

- This section assumes that the following construction procedure has already been done.
  - ① OS Settings (Network settings, ETC)
  - ② Creating and connecting volumes for DRBD synchronization \* For details, see "4.System construction example/Disc requirements."
- The installation for each software is written in assuming that they will be installed online in an online environment  
If you want to install them on an offline environment,  
please make sure that your software dependencies match those of the server you are building before you collect the installation libraries.
- Before the construction procedure, please configure the following basic OS Settings.  
Unless stated otherwise, you will need to perform the settings on both of the servers.
  - Deactivating SELinux  
In order to operate an HA Cluster while SELinux is active, the user must have experience and knowledge regarding SELinux.  
Since the procedure in this document has SELinux deactivated, If you want to activate SELinux, do it at your own risk.
  - Configuring Name resolution  
A Name resolution must be configured between the servers that creates the HA Cluster.  
In the procedure in this document, the server's Host name and IP Address will both be registered to /etc/hosts.  
If you are going to use an DNS server in the user environment, dont register to the /etc/hosts,  
but register the HA Server's Host information to the DNS and configure an appropriate name resolution.
  - Configuring NTP  
When managing HA Cluster operations,it is very important that the system clock is correct and synchronized.  
For this reason, we use the NTP protocol for time synchronization whenever possible.

# Introduction

- Installing ITA

- For information regarding ITA Requirements, please refer to the following documents.

<https://github.com/exastro-suite/it-automation-docs/raw/master/asset/Documents/Exastro-ITA System Configuration Environment Construction Guide Basic>

1.1 Server requirements

<https://github.com/exastro-suite/it-automation-docs/raw/master/asset/Documents/Exastro-ITA System Configuration Environment Construction Guide Ansible-driver>

Exastro-ITA\_System\_Configuration\_Environment\_Construction\_Guide\_Ansible-driver

3. System Requirements

- Installation requirements

Refer to the following URL for the All in one construction installation manual

ITA can be installed through either the online or offline installation procedure.

<https://exastro-suite.github.io/it-automation-docs/learn.html#deploy>

This document assumes that the following functions are activated.

ita\_answers.txt Function settings

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

The ITA startup service changes depending on the ITA Version and the functions the user is going to install. If you are using an environment different version or are installing functions different to what is listed above, the following service contents must be changed appropriately.

- The services found in onstruction procedure "4. ITA Stop settings"
  - The "ITA\_resource.conf" file's Primitive and Group setting services.



## Construction procedure

### User/Connection requirements

Run as root user  
the root user must be able to login through SSH

No.	Operation	Server		Contents (※When “Master” and “Slave” server configuration is required, please configure them in parallel)	Remarks
		master	slave		
1.Install ITA					
1-1	Install ITA	●	●	Please see the All in one installation manual below <a href="https://exastro-suite.github.io/it-automation-docs/learn_ja.html#deploy">https://exastro-suite.github.io/it-automation-docs/learn_ja.html#deploy</a>	
2.DRBD Synchronization settings					
2-1	Configure the environment settings for construction	●	●	【Setting line】 Change the red text to fit the user environment (In this manual, we will use the environment described in the previous section, “2. System Configuration Example”)  export ha1_name=ita-ha-sv01      ※ Specify the Master side's Host name export ha2_name=ita-ha-sv02      ※ Specify the Slave side's Host name export ha1_addr=192.168.1.11    ※ Specify the Master side's Heartbeat IP Address export ha2_addr=192.168.1.12    ※ Specify the Slave side's Heartbeat IP Address export virtual_ip_addr=10.1.1.10 ※ Specify the ITA Access virtual IP address	If you want to re-login in the middle of the procedure, please run this command again.
2-2	Add elrepo repository	●	●	rpm -Uvh http://www.elrepo.org/elrepo-release-7.0-2.el7.elrepo.noarch.rpm	
2-3	Install DRBD	●	●	yum -y install kmod-drbd84	
2-4	※Name resolution not necessary if DNS Server is used Allow HA configuration nodes to resolve each other's names	●	●	echo “\${ha1_addr} \${ha1_name}” >> /etc/hosts echo “\${ha2_addr} \${ha2_name}” >> /etc/hosts	
2-5	Create /dev/vdc1	●	●	echo -e “¥nn¥np¥n1¥n¥n+(Partition size)¥nw”   fdisk /dev/vdc	For the (Partition size), please set the value as such: Number+Measure unit. K=KibiByte, M=MebiByte, E.g) 5GibiByte= 5G
2-6	Create /dev/vdc2 ※If you want to use all the rest	●	●	echo -e “¥nn¥np¥n2¥n¥n¥nw”   fdisk /dev/vdc	
2-7	Fill the created Patition data with “0” and reset the file system	●	●	dd if=/dev/zero of=/dev/vdc1 bs=1M count=1 dd if=/dev/zero of=/dev/vdc2 bs=1M count=1	

## Construction procedure

### User/Connection requirements

Run as root user  
the root user must be able to login through SSH

No.	Operation	Server		Contents (※When "Master" and "Slave" server configuration is required, please configure them in parallel)	Remarks
		master	slave		
2-8	Create DRBD settings file	●	●	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; } }	Install the commands together
				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; } }	Install the commands together
2-9	※Only if using firewalld Allow communication to the used port.	●	●	firewall-cmd --add-port=7788/tcp --zone=public --permanent firewall-cmd --add-port=7789/tcp --zone=public --permanent firewall-cmd --reload	
2-10	Create metadata for DRBD resources	●	●	drbdadm create-md r0 drbdadm create-md r1	
2-11	Run DRBD service	●	●	systemctl start drbd	

## Construction procedure

### User/Connection requirements

Run as root user  
the root user must be able to login through SSH

No.	Operation	Server		Contents (※When “Master” and “Slave” server configuration is required, please configure them in parallel)	Remarks
		master	slave		
2-12	Configure the mount for ITA synchronization on the master side.				
	Initial Synchronize the DRBD synchronization devices from Master to Slave.	●		drbdadm primary --force all	
	Format the DRBD synchronization devices using xfs.	●		mkfs -t xfs /dev/drbd0 mkfs -t xfs /dev/drbd1	
	Configure ITA Synchronization file mount (drbd0)				
	Create a directory of ITA Synchronization and maunt the DRDB Synchronization device	●		mkdir -p /mnt/(ITA Install path) mount /dev/drbd0 /mnt/(ITA Install path)	Please change the red text to fit your environment.こと
	Store the ITA Installation directory in an array variable.	●		dirs=( "/(ITA Install path)/data relay storage/symphony" "/(ITA Install path)/data_relay_storage/conductor" "/(ITA Install path)/data_relay_storage/ansible_driver" "/(ITA Install path)/ita_sessions" "/(ITA Install path)/ita-root/temp" "/(ITA Install path)/ita-root/uploadfiles" "/(ITA Install path)/ita-root/webroot/uploadfiles" "/(ITA Install path)/ita-root/webroot/menus/sheets" "/(ITA Install path)/ita-root/webroot/menus/users" "/(ITA Install path)/ita-root/webconfs/sheets" "/(ITA Install path)/ita-root/webconfs/users" )	Install the commands together Please change the red text to fit your environment.こと
	Move the data from the ITA Synchronization directory to the DRBD Synchronization device and create a symbolic link to the ITA Installation path.	●		for dir in "\${dirs[@]}"; do ## directory backup if [ -d \${dir} ]; then mv \${dir}.org fi ## Create directory under the mounted directory. if [ ! -d `dirname /mnt\${dir}` ]; then mkdir -p `dirname /mnt\${dir}` fi ## Copy ITA Data to under the Mounted directory. if [ ! -d /mnt\${dir} ]; then cp -pr \${dir}.org /mnt\${dir}/ fi ##Create symbolic link if [ -d /mnt\${dir} ]; then ln -s "/mnt\${dir} \${dir} fi ## Comment in if you want to delete the backed up directory. # if [ -d \${dir}.org ]; then # rm -rf \${dir}.org # fi done	Install the commands together
	Configure MariaDB Datafile device (drbd1)				
	Save the source MariaDB data	●		cp -pr /var/lib/mysql /var/lib/mysql.org	
	Mount /var/lib/mysql to the DRBD Synchronization device	●		mount /dev/drbd1 /var/lib/mysql chown -R mysql:mysql /var/lib/mysql	
	Copy the MariaDB Data to the mount (Synchronize) and delete the saved source data.	●		rsync -a --delete /var/lib/mysql.org/ /var/lib/mysql/ rm -rf /var/lib/mysql.org	
	Unmount the DRBD Synchronization device.	●		umount /dev/drbd0 umount /dev/drbd1	
	Demote the DRBD Resource to Secondary.	●		drbdadm secondary all	

## Construction procedure

### User/Connection requirements

Run as root user  
the root user must be able to login through SSH

No.	Operation	Server		Contents (※When “Master” and “Slave” server configuration is required, please configure them in parallel)	Remarks
		master	slave		
2-13	Configure the Slave server's ITA Synchronization mount.				
	Initial synchronize the Slave storage to the Master storage.			● drbdadm primary --force all	
	Format the DRBD Synchronization partition using xfs			● mkfs -t xfs /dev/drbd0 mkfs -t xfs /dev/drbd1	
	Configure ITA Synchronization file mount (drbd0)				
	Create a directory for ITA Synchronization and mount it to DRBD Synchronization device.			● mkdir -p /mnt/(ITA Install path) mount /dev/drbd0 /mnt/(ITA Install path)	Please change the red text to fit your environment.
	Store the ITA Synchronization directory in an array variable.			● dirs=( "/(ITA Install path)/data relay storage/symphony" "/(ITA Install path)/data relay storage/conductor" "/(ITA Install path)/data relay storage/ansible_driver" "/(ITA Install path)/ita_sessions" "/(ITA Install path)/ita-root/temp " "/(ITA Install path)/ita-root/uploadfiles" "/(ITA Install path)/ita-root/webroot/uploadfiles" "/(ITA Install path)/ita-root/webroot/menus/sheets" "/(ITA Install path)/ita-root/webroot/menus/users" "/(ITA Install path)/ita-root/webconfs/sheets" "/(ITA Install path)/ita-root/webconfs/users" )	Install the commands together Please change the red text to fit your environment.
	Move the data from the ITA Synchronization directory to the DRBD Synchronization device and create a symbolic link to the ITA Installation path.			● for dir in "\${dirs[@]}"; do ## back-up directory if [ -d \${dir} ]; then mv \${dir}[_].org fi ## Create directory under the mounted directory if [ ! -d `dirname /mnt\${dir}` ]; then mkdir -p `dirname /mnt\${dir}` fi ## Copy ITA Data to under the mounted directory if [ ! -d /mnt\${dir} ]; then cp -pr \${dir}.org /mnt\${dir}/ fi ## Create symbolic link if [ -d /mnt\${dir} ]; then ln -s "/mnt\${dir} \${dir} fi ## Comment in if you want to delete the backed up directory. # if [ -d \${dir}.org ]; then # rm -rf \${dir}.org # fi done	Install the commands together
	Configure Device for MariaDB Data files (drbd1)				
	Mount /var/lib/mysql to DRBD Synchronization device.			● mount /dev/drbd1 /var/lib/mysql chown -R mysql:mysql /var/lib/mysql	
	Unmount the DRBD Synchronization partition.			● umount /dev/drbd0 umount /dev/drbd1	
	Demote the DRBD Resource to Secondary.			● drbdadm secondary all	
2-14	Stop DRBD Service	●	●	systemctl stop drbd	

## Construction procedure

### User/Connection requirements

Run as root user  
the root user must be able to login through SSH |

No.	Operation	Server		Contents (※When “Master” and “Slave” server configuration is required, please configure them in parallel)	Remarks
		master	slave		
3.Maria DB Stop settings					
3-1	Stops Maria DB	●	●	systemctl disable mariadb systemctl stop mariadb	
4.ITA Stop settings					
4-1	Stops KY-Service	●	●	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_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_varsautilistup-workflow.service systemctl stop ky_legacy_valautostup-workflow.service systemctl stop ky_legacy_varsautilistup-workflow.service systemctl stop ky_mail.service systemctl stop ky_pioneer_valautostup-workflow.service systemctl stop ky_pioneer_varsautilistup-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	Some services might not exist depending on the ITA Version. In that case, there will be no need to stop said service.
4-2	Deactivate ky-service Automatic startup service.	●	●	systemctl disable ky_activatedirectory_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_varsautilistup-workflow.service systemctl disable ky_legacy_valautostup-workflow.service systemctl disable ky_legacy_varsautilistup-workflow.service systemctl disable ky_mail.service systemctl disable ky_pioneer_valautostup-workflow.service systemctl disable ky_pioneer_varsautilistup-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	Some services might not exist depending on the ITA Version. In that case, there will be no need to deactivate said service.

## Construction procedure

User/Connection requirements
Run as root user
the root user must be able to login through SSH

No.	Operation	Server		Contents (※When “Master” and “Slave” server configuration is required, please configure them in parallel)	Remarks
		master	slave		
5.Configure Apache resources					
5-1	Create Apache Server status	●	●	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	Install the commands together
5-2	Stop/Deactivate Apache service.	●	●	systemctl disable httpd systemctl stop httpd	
5-3	Copy the https server certificate/secret key from the master server to the slave server.	●	●	scp /etc/pki/tls/certs/( <del>https</del> server certificate name).crt \${ha2_addr}:/etc/pki/tls/certs/ scp /etc/pki/tls/certs/( <del>https</del> server certificate name).key \${ha2_addr}:/etc/pki/tls/certs/	Make sure to change the https server certificate/secret key to the file name used when installing the ITA Server.
6.Configure Pacemaker					
6-1	Install HA Software	●	●	yum -y install pacemaker NetworkManager NetworkManager-config-server systemctl restart NetworkManager	
6-2	Install the python-parallax that the crmsh needs.	●	●	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	Install crmsh	●	●	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  yum install -y crmsh	Install the commands together
6-4	Configure public ssh key authentication between nodes so that crmsh on the master side can control the cluster construction nodes.	●		ssh-keygen -f /root/.ssh/id_rsa -N "" ssh-copy-id -oStrictHostKeyChecking=no -i /root/.ssh/id_rsa root@\${ha2_addr}	
6-5	Configure public ssh key authentication between nodes so that crmsh on the slave side can control the cluster construction nodes.		●	ssh-keygen -f /root/.ssh/id_rsa -N "" ssh-copy-id -oStrictHostKeyChecking=no -i /root/.ssh/id_rsa root@\${ha1_addr}	
6-6	※Do only if under a proxy. Add main machine/secondary machine IP to no_proxy	●	●	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	※do only if firewalld is used add firewalld settings	●	●	firewall-cmd --add-service=high-availability --permanent firewall-cmd --reload	
6-8	Configure corosync	●	●	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	

## Construction procedure

### User/Connection requirements

Run as root user  
the root user must be able to login through SSH

No.	Operation	Server		Contents (※When "Master" and "Slave" server configuration is required, please configure them in parallel)	Remarks
		master	slave		
6-9	Treat Pacemaker internal process failure as a node failure	●	●	sed -i 's/^#% PCMK_fail_fast=*/PCMK_fail_fast=yes/' /etc/sysconfig/pacemaker	
6-10	Configure corosync	●	●	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	Upload files to the Master server	●		Transfer the material file that accompanies the manual.  Files                      Destination corosync.conf          →    /etc/corosync/	
6-12	Edit /etc/corosync/corosync.conf	●		vi /etc/corosync/corosync.conf  ※If you are going to synchronize the data with the HeartBeatNW bindnetaddr: 192.168.1.11      ※ Specify Heartbeat NW  ring0 addr: 192.168.1.11      ※ Specify the Master server's Heartbeat IP Address ring0_addr: 192.168.1.12      ※ Specify the Slave server's Heartbeat IP Address  ※If you are going to synchronize data using an external NW Connection bindnetaddr: 10.1.1.10      ※ Specify External NW Connection  ring0_addr: 10.1.1.11      ※ Specify Master server's external connection IP Address ring0_addr: 10.1.1.12      ※ Specify Slave server's external connection IP Address  ※If you want to use multicast addresses in corosync, change the following settings. ① comment in mcastaddr line mcastaddr: 239.255.1.1  ②comment out transport: udpu #transport: udpu	Specify the red text to fit your environment
6-13	Copy material files to the slave server	●		scp -p /etc/corosync/corosync.conf root@[ha2_addr]:/etc/corosync/corosync.conf	
6-14	Create/deploy Cluster authentication setting file.	●		corosync-keygen -l scp -p /etc/corosync/authkey root@[ha2_addr]:/etc/corosync/authkey	
6-15	Reload settings file	●	●	systemctl daemon-reload	
6-16	Start HA program services	●	●	systemctl enable corosync systemctl enable pacemaker systemctl start corosync systemctl start pacemaker	

## Construction procedure

### User/Connection requirements

Run as root user  
the root user must be able to login through SSH

No.	Operation	Server		Contents (※When "Master" and "Slave" server configuration is required, please configure them in parallel)	Remarks
		master	slave		
6-17	Check that the Cluster is running properly	●		crm_mon -A  (Display example) ~~~~~Excerpt~~~~~ 2 nodes configured ←Make sure that this is displayed 32 resource instances configured  Online: [ ita-ha-sv01 ita-ha-sv02 ] ←Make sure that both Master and Slave are online  Active resources:  Resource Group: exastro fs_mysql (ocf:heartbeat:Filesystem): Started ita-ha-sv0 ←The Master server has started 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 ~~~~~Excerpt~~~~~	Please change the red text to fit your environment.
6-18	Upload material files to the Master side	●		Transfer the material file that accompanies the manual.  Files                      Destination ita_resource.crm      →    /Directory/	Please change the red text to fit your environment.
6-19	deploy the crm settings file to the master server's node.	●		sed -i 's/virtual_ip¥_addr/"\$virtual_ip_addr"/' /Directory/ita_resource.crm	Please change the red text to fit your environment.
6-20	Reflect the crm settings file.	●		crm configure load update /Directory/ita_resource.crm	Please change the red text to fit your environment.
6-21	Check ITA Connection	●		access the login screen with the URL below <a href="https://10.1.1.10">https://10.1.1.10</a>	Please change the red text to fit your environment.
Operation Complete					



## 【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 {
  # This is the IP Address of the host that joins the cluster. Please change it to fit your er
  node {
    ring0_addr: 192.168.1.11
    nodeid: 1
  }
  # This is the IP Address of the host that joins the cluster. Please change it to fit your er
  node {
    ring0_addr: 192.168.1.12
    nodeid: 2
  }
}

quorum {
  two_node: 1
  expected_votes: 2
  provider: corosync_votequorum
}
```

## 【Reference】ita\_resource.conf sample settings

```
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=xfstype=xfstype ¥
    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=xfstype=xfstype ¥
    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_activatedirectory_roleuser_replication-workflow ky_ansible_execute-workflow
ky_ansible_towermasterSync-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-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

```