

統合 NMS サーバー 環境構築手順書

もくじ

もくじ.....	1
前提条件.....	2
第1章 CentOS のインストール.....	3
1.1 CentOS のインストール方法.....	3
1.2 CentOS の初期セットアップ.....	5
第2章 NMS サーバー設定.....	6
2.1 NMS サーバー設定手順.....	6
第3章 冗長化設定.....	7
3.1 冗長化構成について.....	7
3.2 パーティション追加～物理・論理ボリューム作成(※冗長化用の領域に、新しく追加した物理ディスクを使用する場合).....	9
3.3 パーティション追加～物理・論理ボリューム作成(※冗長化用の領域に、既存の物理ディスクにある未割当領域を使用する場合).....	16
3.4 hostname 設定.....	23
3.5 DRBD インストール.....	24
3.6 DRBD 設定.....	26
3.7 DRBD リソースの設定/起動.....	28
3.8 DRBD の同期およびファイルシステム作成/マウント.....	30
3.9 Pacemaker のインストール/起動/パスワード設定.....	31
3.10 Pacemaker クラスタの設定.....	32
3.11 Pacemaker リソースの設定.....	33
3.12 同期監視用 LAN ケーブル切断後の復旧手順.....	37
第4章 サーバー証明書について.....	38
4.1 Web アクセス用サーバー証明書更新手順.....	39
4.2 5G-GW 接続用サーバー証明書更新手順.....	40

前提条件

本手順書は、以下の設定値やバージョンで作業を説明するものです

設定値

パラメータ	設定値
Root パスワード	P@ssword
ユーザアカウント/パスワード	nmsdev/nmsdev
プロキシアドレス : ポート番号	proxy.co.jp:8080
サーバーホスト名	nmsserver

バージョン

プログラム	バージョン
CentOS	7.7.1908
mariaDB	10.4.4
Nginx	1.15.12
Node.js	8.15.0
Express	4.16.3
Python	3.6.8

第1章 CentOS のインストール

1.1 CentOS のインストール方法

1. インストール対象装置にインストールディスクをセットして電源ボタンを押下する。
※インストールディスクは CentOS のサイトより ISO イメージを取得して作成してください。
CentOS のバージョンについては、操作説明書を参照ください。
2. 起動中にブートメニューの選択画面へ遷移させるための操作を行う。この操作は、インストール対象装置により異なるため、事前に確認しておくこと。
3. ブートメニュー画面が表示されたら、インストールデバイスが表示されたメニューを選択する。
DVD の場合は、「Embedded ~ CD/DVD ROM」のように DVD と書かれたメニューを選択する。
4. 「CentOS 7」と表示されたメニュー画面が表示されるので、「Install CentOS 7」メニューを選択しエンターキーを押下する。
5. 「WELCOME TO CENTOS 7.」と表示された画面が出てくるのを待ち、表示されたら左側の言語メニューから「日本語」を選択し、「続行」ボタンを押下する。
6. 「インストールの概要」画面が表示されるので、「ソフトウェアの選択」ボタンを押下する。
7. 「ソフトウェアの選択」画面が表示される。ここでは使用目的に合わせてインストールするソフトウェアを選択できる。
GUI 環境で使用する為「サーバー(GUI 使用)」を選択し「完了」ボタンを押下する。
8. 続いて「インストール先」ボタンを押下して、インストールするデバイスにチェックを入れた状態で「パーティションを自分で構成する」を選択し「完了」ボタンを押下する。
9. 「インストール先」画面が表示される。既にマウントポイントが割り当てられていた場合は、一ボタンを押下してすべて削除すること。削除が完了したら、LVM を選択して、新規マウントポイントとサイズを以下のように設定する。新規の割り当ては+ボタン押下から行う。割り当てる領域サイズはインストール先のディスク容量に従いサイズを指定すること。

表 1.1-1 インストール先ディスク構成例

マウントポイント	割り当てる領域サイズ	ファイルシステム種別
/	300G	xfs
Swap	8G	swap
/boot	1G	xfs
/boot/efi	1G	EFI System Partition

★冗長化用の 8GB パーティションが別途必要

10. 設定後は、「完了」ボタンを押下する。
11. 続いて「ネットワークとホスト名」ボタンを押下する。
12. ネットワークの設定を行う。以下を利用環境に応じて設定する。

- Web インタフェース IP アドレス : WebUI で NMS と接続する IP アドレス
 - AP インタフェース IP アドレス : AP と接続する IP アドレス
 - 5GGW インタフェース IP アドレス : 5G-GW と接続する IP アドレス
13. 「インストールの概要」画面に戻るので、三角の注意マークが表示されていない事を確認して「インストールの開始」ボタンを押下する。
 14. インストール中に root パスワードとユーザの作成を行う。root パスワードを作成するために「root パスワード」ボタンを押下する。
 15. 「root パスワード」画面が表示される。「root パスワード」と「確認」にそれぞれパスワードを入力し（今回は「P@ssword」として説明する）パスワードを設定する。
入力後「完了」ボタンを2回押下しインストール画面に戻る。（画面下部にパスワードが弱い旨のメッセージが表示されるが無視する）
 16. 次に「ユーザーの作成」ボタンを押下し、「フルネーム」、「ユーザー名」、「パスワード」、「パスワードの確認」にそれぞれ「ユーザー名」（今回は「nmsdev」として説明する）を入力してパスワードを設定し、「このユーザーを管理者にする」にチェックを入れる。
※ここで作成したユーザを以降「管理者ユーザ」とする。
 17. 入力後「完了」ボタンを2回押下する。（画面下部にパスワードが弱い旨のメッセージが表示された場合は、適宜見直すか判断する）
 18. インストール画面に戻って「root パスワード」と「ユーザーの作成」ボタンにそれぞれ三角の注意マークが表示されていない事を確認してインストール終了まで待つ。
 19. 「インストールが完了しました」とメッセージが表示されるので「再起動」ボタンを押下して DVD がイジェクトされるのを待つ。
 20. DVD がイジェクトされたら取り出す。
 21. 更に待つと POST 画面が表示された後に CentOS のブートローダーが表示されるので、一番上の CentOS を選択してエンターキーを押下する。
以下初期セットアップに進む。

1.2 CentOS の初期セットアップ

1. OS が起動すると初期セットアップ画面が表示されるので、「LICENSE INFORMATION」 ボタンを押下する。
2. ライセンスが表示されるのでチェックボックスにチェックをいれ、「完了」 ボタンを押下する。
3. 初期セットアップ画面に戻るので三角の注意マークが表示されていない事を確認して「設定の完了」 ボタンを押下する。
4. ログイン画面が表示されるので、管理者ユーザでログインする。
5. ようこそ画面が表示されるので日本語を選択し「次へ」 ボタンを押下する。
6. 入力画面が表示されるので日本語を選択し「次へ」 ボタンを押下する。
7. 位置情報画面が表示されるので「オフ」を選択し「次へ」 ボタンを押下する。
8. オンラインアカウント画面が表示されるが何も選択せずに「スキップ」 ボタンを押下する。
9. 準備完了画面が表示されるので「CentOS を使い始める」 ボタンを押下する。
10. ホーム画面が表示されれば初期セットアップ完了です。

第2章 NMS サーバー設定

2.1 NMS サーバー設定手順

1. CentOS をインストールした PC の任意の場所に NMS_install.zip をコピーする。
※NMS_install.zip については、操作説明書に記載の Web URL にアクセスいただき、自動環境構築スクリプトよりダウンロードして入手してください。
2. コピーした zip ファイルを展開して、nms_install.sh を実行する。(フォルダパスに日本語が含まれていると実行できないことがあります)

```
unzip NMS_install.zip  
./NMS_install/nms_install.sh
```

3. 実行後、「1.1 CentOS のインストール方法」の 12 で設定した値と同様に、以下の情報を入力する。テナント名には任意の文字列を入力する。
 - 管理者ユーザのパスワード
 - Web インタフェース IP アドレス
 - AP インタフェース IP アドレス
 - 5GGW インタフェース IP アドレス
 - テナント名
4. 10 分程度待つと設定が完了します。途中で 2 分ほど止まるところがあります。
5. "Installation completed."というメッセージが出力されたら完了です。
PC を再起動してください。

※ファイアウォールのポート開放

本スクリプトを実行すると以下のポートが自動で開放されます

https(443/tcp)ポート

snmptrap(162/udp)ポート

snmp(161/udp)ポート

7547/tcp ポート

7557/tcp ポート

7567/tcp ポート

60001/udp ポート

8080/tcp ポート

7070/tcp ポート

10051/tcp ポート

2224/tcp ポート

5405/udp ポート

7789/tcp ポート

第3章 冗長化設定

本章では冗長化設定手順について説明します。冗長化設定を行うとマスター側で故障が発生したとしても、スレーブ側の機器を速やかにあてがうことでシステムの停止時間を縮小できます。また、スレーブ側が可動している間にマスター側の機器を修理・復旧し、再度組み戻すことができます。

本ソフトウェアは、冗長化設定を行わず、単一のシステムで利用することも可能です。

3.1 冗長化構成について

1. ネットワークの構成について

本手順書では、図 3.1-1 のような構成の場合の冗長化手順を記載する。特に記載がない節の設定は、マスターとスレーブの両ホストで実施するものとする。

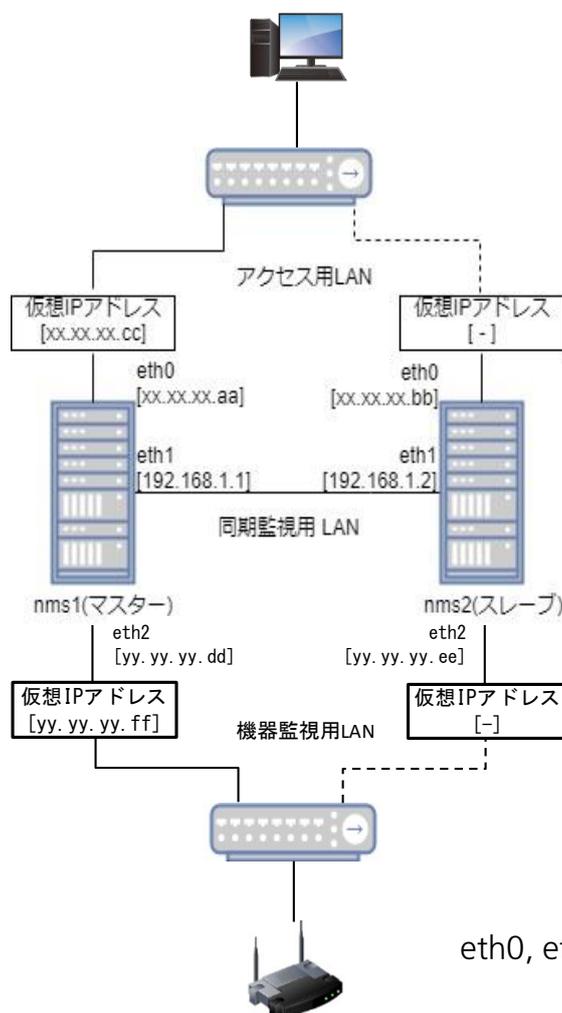


図 3.1-1 冗長化構成

2. パーティションの構成について

冗長化構成をとる場合は、DBを同期するための領域を残す必要がある。

本手順書では8G[Byte]のサイズを確保している。(実際に必要なサイズについては別途確認が必要)

3. Pacemakerによる監視対象と問題検出時の動作について

本手順書では下記を監視対象としている。

- DRBD リソース
- ファイルシステム
- 各種サービス
 - ① mariadb
 - ② nginx
 - ③ php-fpm
 - ④ nms-ap(node)
 - ⑤ zabbix-server
 - ⑥ zabbix-agent
 - ⑦ snmptrapd
 - ⑧ snmpd
 - ⑨ snmptt
 - ⑩ mongodb
 - ⑪ genieacs-cwmp
 - ⑫ genieacs-nbi
 - ⑬ genieacs-fs
 - ⑭ acsdbmng
 - ⑮ supervisor
 - ⑯ crond
- VIP

問題検出時の動作は図 3.1-2 のようになる

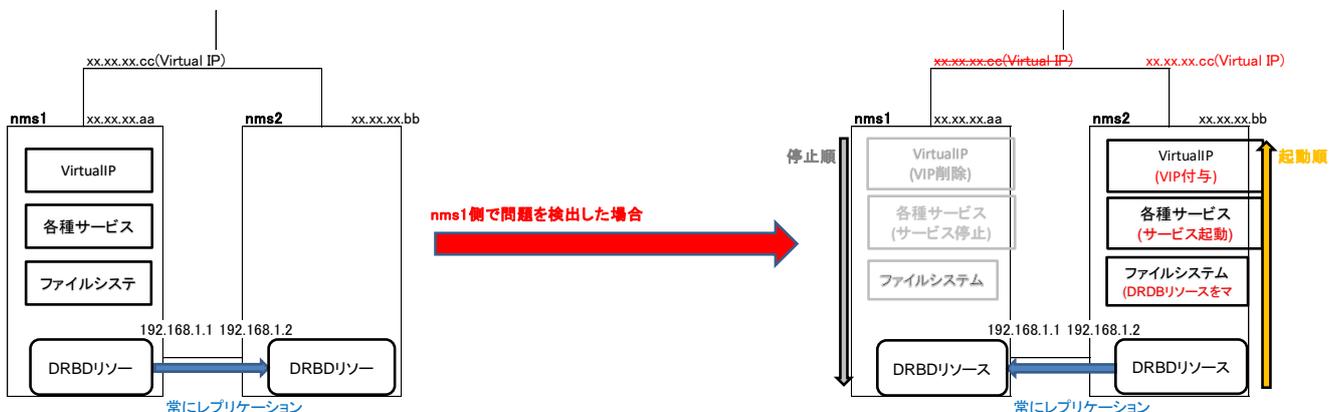


図 3.1-2 問題検出時の動作

3.2 パーティション追加～物理・論理ボリューム作成(※ 冗長化用の領域に、新しく追加した物理ディスクを使用す る場合)

本節では、DB を同期するための領域に、新しく追加した物理ディスクを使用する場合の構築方法を記載する。領域は、図 3.2-1 のような構成とする。

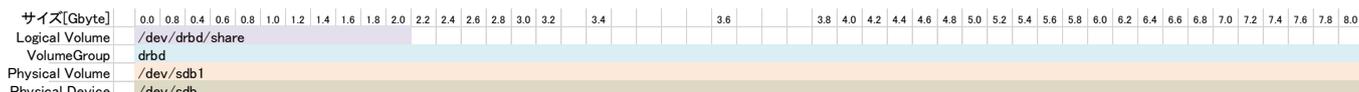


図 3.2-1 冗長化用領域の構成

Command	備考
\$ sudo fdisk -l	デバイス名の確認

```

Disk /dev/sda: 8589 MB, 8589934592 bytes, 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト
Disk label type: dos
ディスク識別子: 0x0006e679

デバイス ブート      始点        終点      ブロック  Id システム
/dev/sda1  *            2048     2099199    1048576  83  Linux
/dev/sda2                2099200   16777215    7339008  8e  Linux LVM

Disk /dev/sdb: 8589 MB, 8589934592 bytes, 16777216 sectors ←DB 同期用に確保している領域
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト

Disk /dev/mapper/centos-root: 6652 MB, 6652166144 bytes, 12992512 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト
  
```

Disk /dev/mapper/centos-swap: 859 MB, 859832320 bytes, 1679360 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト

Command	備考
\$ sudo fdisk /dev/sdb	パーティションの作成

Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x33307c88.

コマンド (m でヘルプ): n ← 「n」を入力して Enter

Partition type:

- p primary (0 primary, 0 extended, 4 free)
- e extended

Select (default p):

Using default response p

パーティション番号 (1-4, default 1): ←何も入力せずに Enter

最初 sector (2048-16777215, 初期値 2048): ←何も入力せずに Enter

初期値 2048 を使います

Last sector, +sectors or +size{K,M,G} (2048-16777215, 初期値 16777215): ←何も入力せずに Enter

初期値 16777215 を使います

Partition 1 of type Linux and of size 8 GiB is set

コマンド (m でヘルプ): w ← 「w」を入力して Enter

パーティションテーブルは変更されました！

ioctl() を呼び出してパーティションテーブルを再読み込みします。
ディスクを同期しています。

Command	備考
\$ sudo fdisk -l	追加したパーティションの確認

Disk /dev/sda: 8589 MB, 8589934592 bytes, 16777216 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト

Disk label type: dos

ディスク識別子: 0x0006e679

デバイス	ブート	始点	終点	ブロック	Id	システム
/dev/sda1	*	2048	2099199	1048576	83	Linux
/dev/sda2		2099200	16777215	7339008	8e	Linux LVM

Disk /dev/sdb: 8589 MB, 8589934592 bytes, 16777216 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト

Disk label type: dos

ディスク識別子: 0x33307c88

デバイス	ブート	始点	終点	ブロック	Id	システム
/dev/sdb1		2048	16777215	8387584	8e	Linux LVM

←追加したパーティションの情報

Disk /dev/mapper/centos-root: 6652 MB, 6652166144 bytes, 12992512 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト

Disk /dev/mapper/centos-swap: 859 MB, 859832320 bytes, 1679360 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト

Command	備考
\$ sudo pvcreate /dev/sdb1	物理ボリュームの作成
\$ sudo pvdisplay	追加した物理ボリュームの確認

```

-- Physical volume --
PV Name           /dev/sda2
VG Name           centos
PV Size           <7.00 GiB / not usable 3.00 MiB
Allocatable       yes (but full)
PE Size           4.00 MiB
Total PE          1791
Free PE           0
Allocated PE      1791
PV UUID           6rzeyK-LQz0-glbG-iypu-l4Sx-gvCf-SU7d7P

"/dev/sdb1" is a new physical volume of "<8.00 GiB"
-- NEW Physical volume --
PV Name           /dev/sdb1
VG Name
PV Size           <8.00 GiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           acZiP5-PtmM-mK01-mQah-caTd-KnGd-E3grWe

```

←追加した物理ボリュームの領域

Command	備考
\$ sudo vgcreate drbd /dev/sdb1	ボリュームグループの作成
\$ sudo vgdisplay	追加したボリュームグループの確認

```

-- Volume group --
VG Name          centos
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No  3
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          2
Open LV          2
Max PV           0
Cur PV          1
Act PV           1
VG Size          <7.00 GiB
PE Size          4.00 MiB
Total PE         1791
Alloc PE / Size  1791 / <7.00 GiB
Free PE / Size   0 / 0
VG UUID          eH85Xh-uMvH-Y0fb-Cgiq-falP-WKTE-6bJluT

-- Volume group --
VG Name          drbd
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No  1
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          0
Open LV          0
Max PV           0
Cur PV          1
Act PV           1

```

←追加したボリュームグループ

VG Size	<8.00 GiB
PE Size	4.00 MiB
Total PE	2047
Alloc PE / Size	0 / 0
Free PE / Size	2047 / <8.00 GiB
VG UUID	JVXwdO-J1or-g4xF-4wgC-QLBX-TPiC-5Pkx3G

Command	備考
\$ sudo lvcreate --name share --size 2GB drbd	論理ボリュームの作成
\$ sudo lvdisplay	追加した論理ボリュームの確認

```

-- Logical volume --
LV Path                /dev/centos/swap
LV Name                swap
VG Name                centos
LV UUID                Y4aG04-gpZV-5Nxl-w5oQ-JYCu-7EcX-yHkh4l
LV Write Access        read/write
LV Creation host, time localhost, 2019-04-30 17:54:26 +0900
LV Status              available
# open                 2
LV Size                820.00 MiB
Current LE             205
Segments               1
Allocation              inherit
Read ahead sectors     auto
- currently set to     8192
Block device           253:1

-- Logical volume --
LV Path                /dev/centos/root
LV Name                root
VG Name                centos
LV UUID                IVX1Fm-p0Wt-OLCe-UO3q-2Yoi-CC0t-9tSDHf
LV Write Access        read/write
LV Creation host, time localhost, 2019-04-30 17:54:27 +0900
LV Status              available
# open                 1
LV Size                <6.20 GiB

```

```
Current LE          1586
Segments           1
Allocation         inherit
Read ahead sectors auto
- currently set to 8192
Block device       253:0

-- Logical volume -- ←追加した論理ボリューム
LV Path            /dev/drbd/share
LV Name            share
VG Name            drbd
LV UUID            FmFmwY-WNDm-TATQ-KUTC-4xdg-3fP0-ni2f9f
LV Write Access    read/write
LV Creation host, time localhost.localdomain, 2019-08-08 10:12:49 +0900
LV Status          available
# open             0
LV Size            2.00 GiB
Current LE         512
Segments           1
Allocation         inherit
Read ahead sectors auto
- currently set to 8192
Block device       253:2
```

3.3 パーティション追加～物理・論理ボリューム作成(※冗長化用の領域に、既存の物理ディスクにある未割当領域を使用する場合)

本節では、DB を同期するための領域として、既存の物理ディスクにある未割当領域を使用する場合の構築方法を記載する。領域は、下記のような構成とする。

```
$ lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0 465.8G  0 disk
├─sda1                8:1    0   953M  0 part /boot
├─sda2                8:2    0 286.9G  0 part
│ └─centos-root       253:0    0 279.4G  0 lvm  /
│ └─centos-swap       253:1    0    7.5G  0 lvm  [SWAP]
└─sda3                8:3    0   953M  0 part /boot/efi
└─sda4                8:4    0     1K  0 part
└─sda5                8:5    0     8G  0 part
    └─drbd-share       253:2    0     2G  0 lvm
        └─drbd1       147:1    0     2G  0 disk /var/lib/mysql
sr0                  11:0    1 1024M  0 rom
```

←ここを追加する

Command	備考
\$ sudo fdisk /dev/sda	パーティションの作成

```

Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

コマンド (m でヘルプ): n ← 「n」を入力して Enter
Partition type:
  p  primary (3 primary, 0 extended, 1 free)
  e  extended
Select (default e): ←何も入力せずに Enter
Using default response e
Selected partition 4
最初 sector (650135552-976773167, 初期値 650135552): ←何も入力せずに Enter
初期値 650135552 を使います
Last sector, +sectors or +size{K,M,G} (650135552-976773167, 初期値 976773167): ←
何も入力せずに Enter
初期値 976773167 を使います
Partition 4 of type Extended and of size 155.8 GiB is set

コマンド (m でヘルプ): n ← 「n」を入力して Enter
All primary partitions are in use
Adding logical partition 5
最初 sector (650137600-976773167, 初期値 650137600): ←何も入力せずに Enter
初期値 650137600 を使います
Last sector, +sectors or +size{K,M,G} (650137600-976773167, 初期値 976773167):
+8G ← 「+8G」を入力して Enter
Partition 5 of type Linux and of size 8 GiB is set

コマンド (m でヘルプ): t ← 「t」を入力して Enter
パーティション番号 (1-5, default 5): ←何も入力せずに Enter
Hex code (type L to list all codes): 8e ← 「8e」を入力して Enter
Changed type of partition 'Linux' to 'Linux LVM'

コマンド (m でヘルプ): p ← 「p」を入力して Enter

```

Disk /dev/sda: 500.1 GB, 500107862016 bytes, 976773168 sectors

Units = sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O サイズ (最小 / 推奨): 512 バイト / 512 バイト

Disk label type: dos

ディスク識別子: 0x0002ccfe

デバイス	ブート	始点	終点	ブロック	Id	システム
/dev/sda1	*	2048	2099199	1048576	83	Linux
/dev/sda2		2099200	648038399	322969600	8e	Linux LVM
/dev/sda3		648038400	650135551	1048576	6	FAT16
/dev/sda4		650135552	976773167	163318808	5	Extended
/dev/sda5		650137600	666914815	8388608	8e	Linux LVM

←追加したパーティションの情報

コマンド (m でヘルプ): w ← 「w」を入力して Enter

パーティションテーブルは変更されました！

ioctl() を呼び出してパーティションテーブルを再読み込みします。

WARNING: Re-reading the partition table failed with error 16: デバイスもしくはリソースがビジー状態です。

The kernel still uses the old table. The new table will be used at the next reboot or after you run partprobe(8) or kpartx(8)

ディスクを同期しています。

Command	備考
\$ sudo reboot	デバイスがビジー状態のため再起動で反映する

Command	備考
\$ sudo pvcreate /dev/sda5	物理ボリュームの作成
\$ sudo pvdisplay	追加した物理ボリュームの確認

```

-- Physical volume --
PV Name           /dev/sda2
VG Name           centos
PV Size           <308.01 GiB / not usable 4.00 MiB
Allocatable       yes
PE Size           4.00 MiB
Total PE          78849
Free PE           1
Allocated PE      78848
PV UUID           5cPbty-Epio-KnAw-UQBu-NBMm-583Q-LE23Uw

"/dev/sda5" is a new physical volume of "8.00 GiB"
-- NEW Physical volume --
PV Name           /dev/sda5
VG Name
PV Size           8.00 GiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           SJFGvo-4B5a-xcJO-8iiv-xxGI-yii1-K9f0Ar
←追加した物理ボリュームの領域

```

Command	備考
\$ sudo vgcreate drbd /dev/sda5	ボリュームグループの作成
\$ sudo vgsdisplay	追加したボリュームグループの確認

```

-- Volume group --
VG Name          centos
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No  3
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          2
Open LV          2
Max PV           0
Cur PV          1
Act PV           1
VG Size          308.00 GiB
PE Size          4.00 MiB
Total PE         78849
Alloc PE / Size  78848 / 308.00 GiB
Free PE / Size   1 / 4.00 MiB
VG UUID          yandrL-CCtk-zANy-w88T-eJ0A-0n6R-Zwyga3

-- Volume group --
VG Name          drbd
System ID
Format           lvm2
Metadata Areas   1
Metadata Sequence No  1
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          0
Open LV          0
Max PV           0
Cur PV          1
Act PV           1

```

←追加したボリュームグループ

VG Size	<8.00 GiB
PE Size	4.00 MiB
Total PE	2047
Alloc PE / Size	0 / 0
Free PE / Size	2047 / <8.00 GiB
VG UUID	B6aPqN-gMla-nO99-Nck6-6fSy-e5tV-hf0Cl6

Command	備考
\$ sudo lvcreate --name share --size 2GB drbd	論理ボリュームの作成
\$ sudo lvdisplay	追加した論理ボリュームの確認

```

-- Logical volume --
LV Path          /dev/centos/root
LV Name          root
VG Name          centos
LV UUID          AeaEbA-bSxz-1j7L-MHzs-zrrQ-lmQB-s107JJ
LV Write Access  read/write
LV Creation host, time localhost, 2019-12-23 17:40:47 +0900
LV Status        available
# open          1
LV Size          300.00 GiB
Current LE       76800
Segments        1
Allocation       inherit
Read ahead sectors auto
                 - currently set to 256
Block device     253:0

-- Logical volume --
LV Path          /dev/centos/swap
LV Name          swap
VG Name          centos
LV UUID          Dohvih-pLAA-FKaZ-nzkC-28wT-GStU-vpcTpl
LV Write Access  read/write
LV Creation host, time localhost, 2019-12-23 17:40:50 +0900
LV Status        available
# open          2
LV Size          8.00 GiB
Current LE       2048

```

Segments 1
Allocation inherit
Read ahead sectors auto
- currently set to 256
Block device 253:1

-- Logical volume -- ←追加した論理ボリューム
LV Path /dev/drbd/share
LV Name share
VG Name drbd
LV UUID nTT5R1-09s0-kCZe-4Eku-I3Ud-cbbs-GzdaAm
LV Write Access read/write
LV Creation host, time nms2, 2020-06-03 11:44:26 +0900
LV Status available
open 0
LV Size 2.00 GiB
Current LE 512
Segments 1
Allocation inherit
Read ahead sectors auto
- currently set to 256
Block device 253:2

3.4 hostname 設定

Command	備考
\$ hostnamectl set-hostname nms1(or 2)	hostname を設定

3.5 DRBD インストール

Command	備考
\$ cd	ホームディレクトリで作業を想定
\$ sudo yum -y install gcc gcc-c++ rpm-build flex automake po4a	必要なパッケージのインストール
\$ uname -r	現 OS のカーネルバージョンを確認する
\$ sudo yum -y install http://vault.centos.org/7.5.1804/os/x86_64/Packages/kernel-devel-3.10.0-862.el7.x86_64.rpm	現 OS のカーネルバージョンが 3.10.0-862.el7.x86_64(「uname -r」の出力結果に対応したパッケージをインストールする)
\$ sudo yum -y install http://vault.centos.org/7.6.1810/os/x86_64/Packages/kernel-devel-3.10.0-957.el7.x86_64.rpm	現 OS のカーネルバージョンが 3.10.0-957.el7.x86_64(「uname -r」の出力結果に対応したパッケージをインストールする)
\$ sudo yum -y install http://vault.centos.org/7.7.1908/os/x86_64/Packages/kernel-devel-3.10.0-1062.el7.x86_64.rpm	現 OS のカーネルバージョンが 3.10.0-1062.el7.x86_64(「uname -r」の出力結果に対応したパッケージをインストールする)
\$ sudo yum -y install kernel-devel	現 OS のカーネルバージョンが 3.10.0-1127.el7.x86_64(「uname -r」の出力結果に対応したパッケージをインストールする)
\$ wget www.linbit.com/downloads/drbd/9.0/drbd-9.0.19-1.tar.gz	DRBD をダウンロード
\$ wget www.linbit.com/downloads/drbd/utills/archive/drbd-utills-9.9.0.tar.gz	DRBD Utilities をダウンロード
\$ mkdir -p rpmbuild/{SOURCES,SPECS}	DRBD のビルド
\$ tar zxvf drbd-9.0.19-1.tar.gz	
\$ cd drbd-9.0.19-1	
\$ make kmp-rpm	
\$ cd	drbd-unit のビルド
\$ tar zxvf drbd-utills-9.9.0.tar.gz	
\$ cd drbd-utills-9.9.0	
\$./configure	

\$ make rpm	
\$ cd ~/rpmbuild/RPMS/x86_64/	インストール
\$ sudo rpm -Uvh drbd-utils-9.9.0-1.el7.x86_64.rpm	競合している場合は <code>-replacefiles</code> をつける。ファイル名が「 <code>drbd-utils-9.9.0-1.el7.centos.x86_64.rpm</code> 」になっている場合あり
\$ sudo rpm -Uvh drbd-udev-9.9.0-1.el7.x86_64.rpm	ファイル名が「 <code>drbd-udev-9.9.0-1.el7.centos.x86_64.rpm</code> 」になっている場合あり
\$ sudo rpm -Uvh drbd-bash-completion-9.9.0-1.el7.x86_64.rpm	ファイル名が「 <code>drbd-bash-completion-9.9.0-1.el7.centos.x86_64.rpm</code> 」になっている場合あり
\$ sudo rpm -Uvh kmod-drbd-9.0.19_3.10.0_862-1.x86_64.rpm	現 OS のカーネルバージョンによりファイル名が変わる
\$ sudo rpm -Uvh drbd-pacemaker-9.9.0-1.el7.x86_64.rpm	ファイル名が「 <code>drbd-pacemaker-9.9.0-1.el7.centos.x86_64.rpm</code> 」になっている場合あり

「make kmp-rpm」でエラーが出た場合

■エラー1

以下のエラーメッセージが表示された場合

```
"/usr/src/kernels/3.10.0-XXX.el7.x86_64.x86_64: No such file or directory. "
```

【解決策】

現 OS のカーネルバージョンを調べて一致するカーネルソースをインストールする
(Command 表の 3~7 行目の手順を参照)

■エラー2

以下のエラーメッセージが表示された場合

```
"SORRY, kernel makefile not found. You need to tell me a correct KDIR!"
```

【解決策】

カーネルソースのパスが間違っているので、正しいパスを `export KDIR=<正しいパス>` で与える
例えば、現 OS のカーネルバージョンが「`3.10.0-862.el7.x86_64`」の場合は、
以下のコマンドを実行した後に「make kmp-rpm」を実行する

```
$ export KDIR=/usr/src/kernels/3.10.0-862.el7.x86_64.x86_64
```

3.6 DRBD 設定

※SELinux は Disable を想定

※Firewall が有効な場合は、7789 ポートを許可しておく（他のサービスが TCP ポート 7789 のポートを使用していないことを想定）

Command	備考
\$ sudo firewall-cmd --add-port=7789/tcp --zone=public --permanent	TCP7789 ポートの許可
\$ sudo firewall-cmd --reload	設定の反映

Command	備考
\$ cd /etc/drbd.d	DRBD のリソースファイルを置くフォルダに移動
\$ sudo touch r0.res	リソースファイルの作成
\$ sudo vim r0.res	リソースファイルに DRBD の設定

```
resource r0 {
  net {
    protocol      C;
  }
  volume 0 {
    device        /dev/drbd1;
    disk          /dev/drbd/share;
    meta-disk     internal;
  }
  on nms1 {
    address 192.168.1.1:7789;
  }
  on nms2 {
    address 192.168.1.2:7789;
  }
}
```

← r0.res ファイルに左記の設定値を書き込み

Command	備考
\$ sudo vim /etc/hosts	hosts ファイルに同期監視用 LAN の IP アドレスを設定

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.1.1 nms1 ←左記の内容を設定
192.168.1.2 nms2
```

Command	備考
\$ sudo reboot	DRBD のリソース作成前に再起動が必要

3.7 DRBD リソースの設定/起動

Command	備考
\$ sudo systemctl stop mariadb.service	NMS に必要な各種プロセスは、クラスタソフトウェアから起動するため停止しておく
\$ sudo systemctl disable mariadb.service	
\$ sudo systemctl stop nginx	
\$ sudo systemctl disable nginx	
\$ sudo systemctl stop php-fpm	
\$ sudo systemctl disable php-fpm	
\$ sudo systemctl stop nms-ap	
\$ sudo systemctl disable nms-ap	
\$ sudo systemctl stop zabbix-server	
\$ sudo systemctl disable zabbix-server	
\$ sudo systemctl stop zabbix-agent	
\$ sudo systemctl disable zabbix-agent	
\$ sudo systemctl stop snmptrapd	
\$ sudo systemctl disable snmptrapd	
\$ sudo systemctl stop snmpd	
\$ sudo systemctl disable snmpd	
\$ sudo systemctl stop snmptt	
\$ sudo systemctl disable snmptt	
\$ sudo systemctl stop rh-mongodb34-mongod	
\$ sudo systemctl disable rh-mongodb34-mongod	
\$ sudo systemctl stop genieacs-cwmp	
\$ sudo systemctl disable genieacs-cwmp	
\$ sudo systemctl stop genieacs-nbi	
\$ sudo systemctl disable genieacs-nbi	
\$ sudo systemctl stop genieacs-fs	
\$ sudo systemctl disable genieacs-fs	
\$ sudo systemctl stop acsdbmng	
\$ sudo systemctl disable acsdbmng	
\$ sudo systemctl stop supervisord	
\$ sudo systemctl disable supervisord	
\$ sudo systemctl stop crond	
\$ sudo systemctl disable crond	
\$ sudo drbdadm create-md r0	DRBD のリソース作成
\$ sudo systemctl start drbd	DRBD の起動。片方ホストの DRBD を起動

	すると、もう片方の DRBD が起動するまで待つため、プロンプトは対向の DRBD が起動するまで戻ってこない
--	---

3.8 DRBD の同期およびファイルシステム作成/マウン

ト

マスター側のみ

Command	備考
\$ sudo drbdadm primary --force r0	nms1 を Primary に昇格および初期同期

両方ホストで実施

Command	備考
\$ sudo drbdadm status r0	ステータス確認

• nms1

```
[nmsdev@nms1 ~]$ drbdadm status r0
```

```
r0 role:Primary ←Primary になっていることを確認
```

```
disk:UpToDate ←ディスクが同期状態(UpToDate)になっていることを確認
```

```
nms2 role:Secondary
```

```
peer-disk:UpToDate ←ディスクが同期状態(UpToDate)になっていることを確認
```

• nms2

```
[nmsdev@nms2 ~]$ drbdadm status r0
```

```
r0 role:Secondary ←Secondary になっていることを確認
```

```
disk:UpToDate
```

```
nms1 role:Primary ←ディスクが同期状態(UpToDate)になっていることを確認
```

```
peer-disk:UpToDate ←ディスクが同期状態(UpToDate)になっていることを確認
```

マスター側のみ

Command	備考
\$ sudo mkdir -p /usr/local/mysql/backup	現在のデータベースの情報をマウント前にコピー
\$ sudo cp -rp /var/lib/mysql /usr/local/mysql/backup	
\$ sudo mkfs.xfs /dev/drbd1 -f	/dev/drbd1 にファイルシステムを作成
\$ sudo mount /dev/drbd1 /var/lib/mysql	/dev/drbd1 を/var/lib/mysql にマウント
\$ sudo cp -rp /usr/local/mysql/backup/mysql/* /var/lib/mysql/	コピーしておいたデータベースの情報を戻す
\$ sudo chown -R mysql:mysql /var/lib/mysql	
\$ sudo chmod -R 755 /var/lib/mysql	

3.9 Pacemaker のインストール/起動/パスワード設定

Command	備考
\$ sudo systemctl stop drbd.service	DRDB はクラスタソフトウェアから起動するため停止しておく
\$ sudo systemctl disable drbd	
\$ sudo umount /dev/drbd1	
\$ sudo yum -y install pacemaker pcs corosync	Pacemaker のインストール
\$ sudo systemctl start pcsd	Pacemaker の起動
\$ sudo systemctl enable pcsd	
\$ sudo passwd hacluster	hacluster ユーザにマスタースレーブ間の通信の認証で使用するパスワードを設定する

※Firewall が有効な場合は、Pacemaker 起動前にクラスタ関連サービスを許可しておく

Command	備考
\$ sudo firewall-cmd --permanent --add-service=high-availability	クラスタ関連サービスの許可
\$ sudo firewall-cmd --reload	設定の反映

3.10 Pacemaker クラスタの設定

マスター側のみ

Command	備考
\$ sudo pcs cluster auth nms1 nms2 -u hacluster	本コマンド実行後にパスワード入力を求められるので、上の手順で設定した hacluster ユーザのパスワードを入力する
\$ sudo pcs cluster setup --force --name database_cluster nms1 nms2	クラスタの同期
\$ sudo pcs cluster start --all	クラスタの起動
\$ sudo pcs cluster enable --all	
\$ sudo pcs status	クラスタのステータス確認

```
Cluster name: database_cluster
```

```
WARNINGS:
```

```
No stonith devices and stonith-enabled is not false
```

```
Stack: corosync
```

```
Current DC: nms2 (version 1.1.19-8.el7_6.4-c3c624ea3d) - partition with quorum
```

```
Last updated: Fri Aug 16 13:09:58 2019
```

```
Last change: Fri Aug 16 13:07:56 2019 by hacluster via crmd on nms2
```

```
2 nodes configured
```

```
0 resources configured
```

```
Online: [ nms1 nms2 ] ←Online になっていることを確認
```

```
No resources
```

```
Daemon Status:
```

```
corosync: active/disabled
```

```
pacemaker: active/disabled
```

```
pcsd: active/enabled
```

3.11 Pacemaker リソースの設定

マスター側のみ

Command	備考
\$ sudo pcs property set stonith-enabled=false	Pacemaker のパラメータ設定
\$ sudo pcs property set no-quorum-policy=ignore	
\$ sudo pcs property set default-action-timeout=240	
\$ sudo pcs property set default-resource-stickiness=200	
\$ sudo pcs property set cluster-recheck-interval=60s	
\$ sudo pcs resource defaults resource-stickiness="INFINITY"	
\$ sudo pcs resource defaults migration-threshold="1"	
\$ sudo pcs resource create drbd_r0 ocf:linbit:drbd drbd_resource=r0 op monitor interval=20s	DRBD リソース定義
\$ sudo pcs resource master ms_drbd_r0 drbd_r0 \ master-max=1 master-node-max=1 clone-max=2 clone-node-max=1 notify=true	リソース drbd_r0 を Master/Slave 形式とし、2 ノードで立ち上げ、うち 1 ノードが Master となるように指定
\$ sudo pcs resource create cluster ocf:heartbeat:Filesystem \ device=/dev/drbd1 directory=/var/lib/mysql fstype=xfs --group rggroup	DRBD ブロックデバイス /var/lib/mysql を /mnt/data へマウントおよびリソースグループ(rggroup)の設定
\$ sudo pcs constraint order promote ms_drbd_r0 then start cluster	起動順序設定(ms_drbd_r0 ->cluster)
\$ sudo pcs constraint colocation add cluster with ms_drbd_r0 INFINITY with-rsc-role=Master	強制配置の設定(cluster, ms_drbd_r0)
\$ sudo pcs resource create mariadb systemd:mariadb --group rggroup	mariadb の設定
\$ sudo pcs resource create nginx systemd:nginx --group rggroup	nginx の設定
\$ sudo pcs resource create php-fpm systemd:php-fpm --group rggroup	php-fpm の設定
\$ sudo pcs resource create nms-ap systemd:nms-ap --group rggroup	nms-ap(node)の設定
\$ sudo pcs resource create zabbix-server systemd:zabbix-server --group rggroup	zabbix-server の設定
\$ sudo pcs resource create zabbix-agent systemd:zabbix-agent --group rggroup	zabbix-agent の設定
\$ sudo pcs resource create snmptrapd systemd:snmptrapd --group rggroup	snmptrapd の設定

\$ sudo pcs resource create snmpd systemd:snmpd --group rgroup	snmpd の設定
\$ sudo pcs resource create snmptt systemd:snmptt --group rgroup	snmptt の設定
\$ sudo pcs resource create rh-mongodb34-mongod systemd:rh-mongodb34-mongod --group rgroup	mongodb の設定
\$ sudo pcs resource create genieacs-cwmp systemd:genieacs-cwmp --group rgroup	genieacs-cwmp の設定
\$ sudo pcs resource create genieacs-nbi systemd:genieacs-nbi --group rgroup	genieacs-nbi の設定
\$ sudo pcs resource create genieacs-fs systemd:genieacs-fs --group rgroup	genieacs-fs の設定
\$ sudo pcs resource create acsdbmng systemd:acsdbmng --group rgroup	acsdbmng の設定
\$ sudo pcs resource create supervisord systemd:supervisord --group rgroup	supervisord の設定
\$ sudo pcs resource create crond systemd:crond --group rgroup	crond の設定
\$ sudo pcs resource create VIP1 ocf:heartbeat:IPaddr2 nic="XXX" ip=xx.xx.xx.cc cidr_netmask=24 op monitor interval=20s --group rgroup	Web インタフェース側の仮想 IP 設定
\$ sudo pcs resource create VIP2 ocf:heartbeat:IPaddr2 ip=xx.xx.xx.dd cidr_netmask=24 op monitor interval=20s --group rgroup"	AP インタフェース側の仮想 IP 設定 ※他のインタフェースと共用の場合は不要
\$ sudo pcs resource create VIP3 ocf:heartbeat:IPaddr2 ip=xx.xx.xx.ee cidr_netmask=24 op monitor interval=20s --group rgroup"	5GGW インタフェース側の仮想 IP 設定 ※他のインタフェースの共用の場合は不要
\$ sudo pcs resource cleanup	念のため検出しているエラーをクリーンアップしておく
\$ sudo pcs status	状態確認

```

Cluster name: database_cluster
Stack: corosync
Current DC: nms2 (version 1.1.21-4.el7-f14e36fd43) - partition with quorum
Last updated: Tue Jun 30 20:16:58 2020
Last change: Tue Jun 30 19:59:27 2020 by hacluster via crmd on nms2

2 nodes configured
21 resources configured

```

Online: [nms1 nms2]

Full list of resources:

Master/Slave Set: ms_drbd_r0 [drbd_r0] ←設定したリソース

Masters: [nms1] ←nms1 がマスターになっていることを確認

Slaves: [nms2]

Resource Group: rgroup

```
cluster (ocf::heartbeat:Filesystem): Started nms1
mariadb (systemd:mariadb): Started nms1
nginx (systemd:nginx): Started nms1
php-fpm (systemd:php-fpm): Started nms1
nms-ap (systemd:nms-ap): Started nms1
zabbix-server (systemd:zabbix-server): Started nms1
zabbix-agent (systemd:zabbix-agent): Started nms1
snmptrapd (systemd:snmptrapd): Started nms1
snmpd (systemd:snmpd): Started nms1
snmptt (systemd:snmptt): Started nms1
rh-mongodb34-mongod (systemd:rh-mongodb34-mongod): Started nms1
genieacs-cwmp (systemd:genieacs-cwmp): Started nms
genieacs-nbi (systemd:genieacs-nbi): Started nms1
genieacs-fs (systemd:genieacs-fs): Started nms1
acsdbmng (systemd:acsdbmng): Started nms1
supervisord (systemd:supervisord): Started nms1
crond (systemd:crond): Started nms1
VIP (ocf::heartbeat:IPAddr2): Started nms1
VIP2 (ocf::heartbeat:IPAddr2): Started nms1
```

←nms1 が動作していることを確認

Daemon Status:

```
corosync: active/enabled
pacemaker: active/enabled
pcsd: active/enabled
```

※仮想 IP 設定の注意事項

■マスター側とスレーブ側で仮想対象にする NIC の名前が異なる場合

- ・ コマンドパラメータ「nic="XXX"」を使用しないこと

- ・ マスター側とスレーブ側の 実 IP を、仮想 IP と同じネットワークアドレスにすること

例えば、仮想 IP に 192.168.0.*/24 を設定する場合は、

マスター側とスレーブ側で仮想対象にする NIC の実 IP を 192.168.0.*/24 にしておく

コマンドパラメータ「nic="XXX"」は、マスター側とスレーブ側で仮想対象にする NIC を特定するために用いられるため、「nic="XXX"」を使用する場合はマスター側とスレーブ側で NIC の名前が同じである必要がある

一方、「nic="XXX"」を使用しない場合は、マスター側とスレーブ側で仮想 IP と同じネットワークアドレスを持つ NIC を仮想対象とするため、これらのネットワークアドレスを同じにしておく必要がある

3.12 同期監視用 LAN ケーブル切断後の復旧手順

冗長化の同期ケーブル切断等のエラー発生時の復旧手段を以下に記載する。

切断要因	同期監視用 LAN 復旧後の状態		復旧手順 ※
	nms1(元マスター)の状態	nms2(元スレーブ)の状態	
マスター側の LAN を HUB から抜く	■ 「sudo pcs status」 コマンドを実行した状態 Masters: [nms1] Slaves: [nms2]	■ 「sudo pcs status」 コマンドを実行した状態 Masters: [nms1] Slaves: [nms2]	1. nms2(元スレーブ)にて以下のコマンドを実行 \$ sudo drbdadm disconnect r0 \$ sudo drbdadm secondary r0 \$ sudo drbdadm connect --discard-my-data r0 2. nms1(元マスター)にて以下のコマンドを実行 \$ sudo drbdadm primary r0 \$ sudo drbdadm connect r0
スレーブ側の LAN を HUB から抜く	■ 「sudo drbdadm status r0」 コマンドを実行した状態 nms2 connection:StandAlone	■ 「sudo drbdadm status r0」 コマンドを実行した状態 nms1 connection:StandAlone または nms1 connection:Connecting	
両方の LAN を HUB から抜く			
HUB の電源断			

※コマンド実行は必ず nms2(元スレーブ)側から実行すること

元マスター、元スレーブの確認は、マスターの"/var/log/cluster/corosync.log"ファイルには "Master/Slave Set: ms_drbd_r0 [drbd_r0]"が出ることから判別できます

第4章 サーバー証明書について

本ソフトウェアは、クライアント PC からの Web アクセス及び 5G-GW 機器との通信はデフォルトでは自己証明書を使った SSL 通信を行っていますが、お客様側で準備されたサーバー証明書をお使いになりたい場合、本項記載の手順により入れ替えが可能です。本項ではサーバー証明書の更新を行う手順について説明します。

重要

- 閉域網以外で使う場合には認証局で署名されたサーバー証明書をご利用ください。

4.1 Web アクセス用サーバー証明書更新手順

本項ではクライアント PC からの Web アクセス用のサーバー証明書の更新を行う手順について説明します。

操作手順

手順1 NMS をインストールしたサーバーにログインします。

手順2 以下フォルダの nginx.pem と nginx.key を使用するサーバー証明書ファイルで上書きします。
/etc/nginx/ssl/

手順3 以下のコマンドで NMS の再起動を行う事で、証明書の更新が反映されます。

```
$sudo systemctl restart nginx.service
```

```
[user@localhost ~]$ ll /etc/nginx/ssl/
合計 8
-rw-r--r--. 1 root root 1704 7月 13 11:33 nginx.key
-rw-r--r--. 1 root root 1220 7月 13 11:33 nginx.pem
[user@localhost ~]$
```

証明書の更新日時を確認してください

図 4.1-1 サーバー証明書ファイル

4.2 5G-GW 接続用サーバー証明書更新手順

本項では 5G-GW 接続用のサーバー証明書の更新を行う手順について説明します。

操作手順

手順1 NMS をインストールしたサーバーにログインします。

手順2 以下フォルダの server.crt と server.key を使用するサーバー証明書ファイルで上書きします。

/home/管理者ユーザ名/.nvm/versions/node/v8.15.0/lib/node_modules/genieacs/config/

手順3 以下のコマンドで NMS の再起動を行う事で、証明書の更新が反映されます。

```
$sudo systemctl stop genieacs-cwmp.service genieacs-fs.service genieacs-nbi.service
```

```
$sudo systemctl start genieacs-cwmp.service genieacs-fs.service genieacs-nbi.service
```

```
[user@localhost /]$ cd home/user/.nvm/versions/node/v8.15.0/lib/node_modules/genieacs/config
[user@localhost config]$ ll
合計 44
-rwxrwxr--. 1 user user 196 10月 26 1985 auth-sample.js
lrwxrwxrwx. 1 user user 6 10月 20 14:42 bk.cwmp.ca-bundle -> ca.crt
-rw-r--r--. 1 user user 591 10月 20 14:42 ca.crt
-rwxrwxr--. 1 user user 15603 10月 20 14:42 client-auth-info.js
-rwxrwxr--. 1 user user 298 10月 26 1985 config-sample.json
-rwxrwxr--. 1 user user 991 10月 20 14:42 config.json
lrwxrwxrwx. 1 user user 10 10月 20 14:42 cwmp.crt -> server.crt
lrwxrwxrwx. 1 user user 10 10月 20 14:42 cwmp.key -> server.key
drwxrwxr-x. 2 user user 19 10月 20 14:42 ext
-rwxrwxr--. 1 user user 937 10月 26 1985 fs.crt -> server.crt
lrwxrwxrwx. 1 user user 10 10月 20 14:42 fs.key -> server.key
-rw-r--r--. 1 user user 611 1月 1 1980 server.crt
-rw-r--r--. 1 user user 302 1月 1 1980 server.key
[user@localhost config]$
```

証明書の更新日時を確認してください

図 4.2-1 サーバー証明書ファイル

重要

■本ソフトウェアでサーバー証明書を入れ替えた場合、5G-GW 機器側にも対応したルート証明書をインストールする必要があります(手順は 5G-GW 機器側のマニュアル XC-WN930J-01_取扱説明書(設定編)の「5.4 証明書ファイルの管理をご確認ください)」