

# サーバ UCS C240 M4 の欠陥のあるコンポーネントの置換- vEPC

## 目次

[はじめに](#)

[背景説明](#)

[省略形](#)

[Mop の作業の流れ](#)

[前提条件](#)

[バックアップ](#)

[コンポーネント RMA - Compute/OSD 計算ノード](#)

[Compute/OSD 計算ノードでホストされる VM を識別して下さい](#)

[電源遮断の段階的](#)

[1.計算ノード ホスト SF だけ VM を包装して下さい](#)

[2. Compute/OSD 計算ノード ホスト CF/ESC/EM/UAS を包装して下さい](#)

[Compute/OSD 計算ノードからの欠陥のあるコンポーネントを交換して下さい](#)

[VM を復元する](#)

[1.計算ノード ホスト SF だけ VM を包装して下さい](#)

[2. Compute/OSD 計算ノード ホスト CF、ESC、EM および UAS を包装して下さい](#)

[ESC リカバリ失敗を処理して下さい](#)

[自動導入 設定アップデート](#)

[コンポーネント RMA - Controller ノード](#)

[事前点検](#)

[メンテナンス モードにコントローラ クラスタを移動して下さい](#)

[Controller ノードからの欠陥のあるコンポーネントを交換して下さい](#)

[サーバの電源](#)

## 概要

ホスト StarOS バーチャルネットワークが機能することこの資料が設定される Ultra-M の Unified Computing System ( UCS ) サーバでここに述べられる欠陥のあるコンポーネントを交換するために必要なステップを記述したものです ( VNFs )。

- Dual In-Line Memory Module ( DIMM ) 置換 MOP
- FlexFlash コントローラ障害
- ソリッドステートドライブ ( SSD ) 失敗
- 信頼されたプラットフォーム モジュール ( TPM ) 失敗
- 侵略キャッシュ失敗
- 侵略コントローラ熱いバス アダプタ ( HBA ) 失敗
- PCI 暴徒失敗
- PCIe アダプタ Intel X520 10G 失敗
- マザーボード ( MLOM ) 失敗 LAN でモジュラ

- ファントレイ RMA
- CPU 失敗

## 背景説明

Ultra-M は事前包装され、検証された仮想化されたモバイル パケット コア ソリューションです VNFs の配備を簡素化するために設計されている。OpenStack は Ultra-M のための Virtualized Infrastructure Manager ( VIM ) で、これらのノードタイプで構成されています:

- 計算
- オブジェクト ストレージ ディスク-計算 ( OSD -計算 )
- コントローラ
- OpenStack プラットフォーム-ディレクター ( OSPD )

Ultra-M の高レベル アーキテクチャおよび含まれるコンポーネントはこのイメージで描写されます:

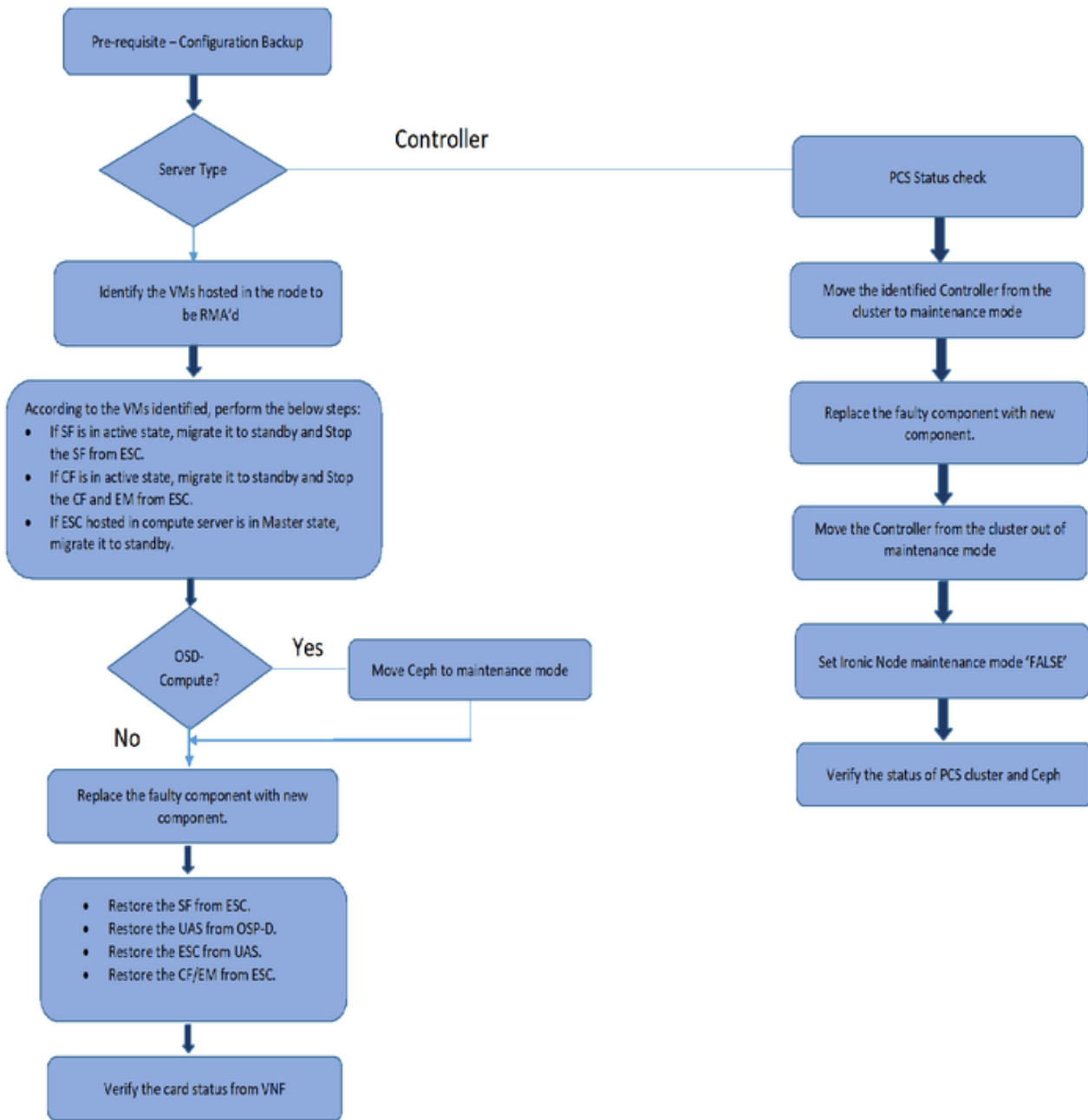
この資料は Cisco Ultra-M プラットフォームについて詳しく知っているサーバのコンポーネント置換の時に OpenStack および StarOS VNF レベルで遂行されることを必要なステップを詳述します Cisco社員のために意図され。

注: 超 M 5.1.x リリースはこの資料の手順を定義するために考慮されます。

## 省略形

VNF	バーチャルネットワーク 機能
CF	制御機能
SF	サービス 機能
ESC	伸縮性があるサービス コントローラ
MOP	プロシージャの方式
OSD	オブジェクト ストレージ ディスク
HDD	ハードディスク ドライブ
SSD	ソリッドステート ドライブ
VIM	仮想 な インフラストラクチャ マネージャ
VM	仮想マシン
EM	Element Manager
UAS	Ultra Automation Services
UUID	ユニバーサル固有の識別番号

## Mop の作業の流れ



## 前提条件

### バックアップ

欠陥のあるコンポーネントを交換する前に、Red Hat OpenStack プラットフォーム 環境の現在のステータスをチェックすることは重要です。交換の手順がオンになっているとき複雑な状況を避けるためにチェック 現在のステータス推奨します。それを置換のこのフローによって達成することができます。

リカバリの場合には、Cisco はこれらのステップの使用の OSPD データベースのバックアップを奪取することを推奨します:

```
[root@director ~]# mysqldump --opt --all-databases > /root/undercloud-all-databases.sql
[root@director ~]# tar --xattrs -czf undercloud-backup-`date +%F`.tar.gz /root/undercloud-all-databases.sql
/etc/my.cnf.d/server.cnf /var/lib/glance/images /srv/node /home/stack
tar: Removing leading `/' from member names
```

このプロセスはノードが例ことをの Availability に影響を与えないで取り替えることができるようにします。また特に取り替えられるべき compute/OSD 計算ノードが制御機能 ( CF ) Virtual Machine ( VM ) をホストすれば、StarOS 設定をバックアップすることを推奨します。

注: サーバが Controller ノードである場合、セクションに進んで下さい、次のセクションと他では続けて下さい。

## コンポーネント RMA - Compute/OSD 計算ノード

### Compute/OSD 計算ノードでホストされる VM を識別して下さい

サーバでホストされる VM を識別して下さい。2つの可能性がある場合もあります:

- サーバはサービス機能だけ ( SF ) VM 含まれています:

```
[stack@director ~]$ nova list --field name,host | grep compute-10
| 49ac5f22-469e-4b84-badc-031083db0533 | VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-
10e75d0e134d |
pod1-compute-10.localdomain |
```

- サーバ含んでいます制御機能 ( CF ) /Elastic サービス コントローラ ( ) /Element Manager ( EM ) 超 ESC/VM のオートメーション サービス ( UAS ) 組み合わせ:

```
[stack@director ~]$ nova list --field name,host | grep compute-8
| 507d67c2-1d00-4321-b9d1-da879af524f8 | VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-
88a2d6fa82ea | pod1-compute-8.localdomain |
| f9c0763a-4a4f-4bbd-af51-bc7545774be2 | VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-
3812653ee229 | pod1-compute-8.localdomain |
| 75528898-ef4b-4d68-b05d-882014708694 | VNF2-ESC-ESC-
0 | pod1-compute-8.localdomain |
| f5bd7b9c-476a-4679-83e5-303f0aae9309 | VNF2-UAS-uas-
0 | pod1-compute-8.localdomain |
```

注: ここに示されている出力では最初のカラムは固有の識別番号 ( UUID ) に、2番目のカラムです VM 名前ユニバーサル対応し、第3カラムは VM があるホスト名です。この出力からのパラメータはそれに続くセクションで使用されます。

## 電源遮断の段階的

### 1. 計算ノード ホスト SF だけ VM を包装して下さい

## STANDBY 状態に SF カードを移行して下さい

- SF VM に対応する StarOS VNF へのログインはカードを識別し。セクションから「識別される SF VM の UUID を識別し UUID に対応する Compute/OSD 計算ノードで」ホストされる VM を識別しますカードを使用して下さい:

```
[local]VNF2# show card hardware
Tuesday night 08 16:49:42 UTC 2018
<snip>
```

### Card 8:

```
Card Type           : 4-Port Service Function Virtual Card
CPU Packages        : 26 [#0, #1, #2, #3, #4, #5, #6, #7, #8, #9, #10, #11, #12, #13, #14,
#15, #16, #17, #18, #19, #20, #21, #22, #23, #24, #25]
CPU Nodes           : 2
CPU Cores/Threads   : 26
Memory              : 98304M (qvpc-di-large)
UUID/Serial Number  : 49AC5F22-469E-4B84-BADC-031083DB0533
```

- カードのステータスをチェックして下さい:

```
[local]VNF2# show card table
Tuesday night 08 16:52:53 UTC 2018
```

Slot	Card Type	Oper State	SPOF	Attach
1: CFC	Control Function Virtual Card	Active	No	
2: CFC	Control Function Virtual Card	Standby	-	
3: FC	4-Port Service Function Virtual Card	Active	No	
4: FC	4-Port Service Function Virtual Card	Active	No	
5: FC	4-Port Service Function Virtual Card	Active	No	
6: FC	4-Port Service Function Virtual Card	Active	No	
7: FC	4-Port Service Function Virtual Card	Active	No	
<b>8: FC</b>	<b>4-Port Service Function Virtual Card</b>	<b>Active</b>	<b>No</b>	
9: FC	4-Port Service Function Virtual Card	Active	No	
10: FC	4-Port Service Function Virtual Card	Standby	-	

- カードが ACTIVE 状態にある場合、STANDBY 状態にカードを移動して下さい:

```
[local]VNF2# show card table
Tuesday night 08 16:52:53 UTC 2018
```

Slot	Card Type	Oper State	SPOF	Attach
1: CFC	Control Function Virtual Card	Active	No	
2: CFC	Control Function Virtual Card	Standby	-	
3: FC	4-Port Service Function Virtual Card	Active	No	
4: FC	4-Port Service Function Virtual Card	Active	No	
5: FC	4-Port Service Function Virtual Card	Active	No	
6: FC	4-Port Service Function Virtual Card	Active	No	
7: FC	4-Port Service Function Virtual Card	Active	No	
<b>8: FC</b>	<b>4-Port Service Function Virtual Card</b>	<b>Active</b>	<b>No</b>	
9: FC	4-Port Service Function Virtual Card	Active	No	
10: FC	4-Port Service Function Virtual Card	Standby	-	

## ESC SF VM

- VNF に対応する SF VM のステータスをチェックし、ESC ノードへのログインは:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
    VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
    VM_ALIVE_STATE
    VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
    VM_ALIVE_STATE</state>
<snip>
```

- VM 名前の使用の SF VM を停止して下さい。（セクションからの VM ネーム注目される「Compute/OSD 計算ノードで」ホストされる VM を識別します）:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
    VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
    VM_ALIVE_STATE
    VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
    VM_ALIVE_STATE</state>
<snip>
```

- それが停止すれば、VM は切断状態を入力する必要があります:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
    VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
    VM_ALIVE_STATE
    VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
    VM_ALIVE_STATE
    VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
    VM_SHUTOFF_STATE</state>
```

## ケース 2. Compute/OSD 計算ノード ホスト CF/ESC/EM/UAS

### STANDBY 状態への移行する CF カード

- CF VM に対応する StarOS VNF へのログインはカードを識別し。セクションから「識別される CF VM の UUID を識別し UUID に対応するノードで」ホストされる VM を見つけますカードを使用して下さい:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
    VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
    VM_ALIVE_STATE
    VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
```

```
VM_ALIVE_STATE
VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
  VM_SHUTOFF_STATE</state>
```

- カードのステータスをチェックして下さい:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
  VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
  VM_ALIVE_STATE
  VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
  VM_ALIVE_STATE
  VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
  VM_SHUTOFF_STATE</state>
```

- カードが ACTIVE 状態にある場合、STANDBY 状態にカードを移動して下さい:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
  VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
  VM_ALIVE_STATE
  VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
  VM_ALIVE_STATE
  VNF2-DEPLOYM_s9_0_8bc6cc60-15d6-4ead-8b6a-10e75d0e134d
  VM_SHUTOFF_STATE</state>
```

## ESC からの CF および EM VM をシャットダウンして下さい

- VNF に対応する VM のステータスをチェックし、ESC ノードへのログインは:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
  VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
  VM_ALIVE_STATE</state>
  VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
  VM_ALIVE_STATE
<deployment_name>VNF2-DEPLOYMENT-em</deployment_name>
  507d67c2-1d00-4321-b9d1-da879af524f8
  dc168a6a-4aeb-4e81-abd9-91d7568b5f7c
  9ffec58b-4b9d-4072-b944-5413bf7fcf07
  SERVICE_ACTIVE_STATE
  VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea
  VM_ALIVE_STATE</state>
<snip>
```

- VM 名前の使用と CF および EM VM を順次に停止して下さい。（セクションからの VM ネーム注目される「Compute/OSD 計算ノードで」ホストされる VM を識別します）:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
    VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229
    VM_ALIVE_STATE</state>
    VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
    VM_ALIVE_STATE
<deployment_name>VNF2-DEPLOYMENT-em</deployment_name>
    507d67c2-1d00-4321-b9d1-da879af524f8
    dc168a6a-4aeb-4e81-abd9-91d7568b5f7c
    9ffec58b-4b9d-4072-b944-5413bf7fcf07
    SERVICE_ACTIVE_STATE
    VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea
    VM_ALIVE_STATE</state>
<snip>
```

- それが停止した後、VM は切断状態を入力する必要があります:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
    VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229</vm_name>
    VM_SHUTOFF_STATE</state>
    VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
    VM_ALIVE_STATE
<deployment_name>VNF2-DEPLOYMENT-em</deployment_name>
    507d67c2-1d00-4321-b9d1-da879af524f8
    dc168a6a-4aeb-4e81-abd9-91d7568b5f7c
    9ffec58b-4b9d-4072-b944-5413bf7fcf07
    SERVICE_ACTIVE_STATE
    VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea</vm_name>
    <state>VM_SHUTOFF_STATE</state>
<snip>
```

## スタンバイ モードへの移行する ESC

- マスター状態にある場合ノードおよびチェックでホストされる ESC へのログイン。Yes の場合は、スタンバイ モードに ESC を切り替えて下さい:

```
[admin@VNF2-esc-esc-0 ~]$ cd /opt/cisco/esc/esc-confd/esc-cli
[admin@VNF2-esc-esc-0 esc-cli]$ ./esc_nc_cli get esc_datamodel | egrep --color
"<state>|<vm_name>|<vm_id>|<deployment_name>"
<snip>
<state>SERVICE_ACTIVE_STATE</state>
    VNF2-DEPLOYM_c1_0_df4be88d-b4bf-4456-945a-3812653ee229</vm_name>
    VM_SHUTOFF_STATE</state>
```



```
VNF2-DEPLOYM_c3_0_3e0db133-c13b-4e3d-ac14-
VM_ALIVE_STATE
<deployment_name>VNF2-DEPLOYMENT-em</deployment_name>
507d67c2-1d00-4321-b9d1-da879af524f8
dc168a6a-4aeb-4e81-abd9-91d7568b5f7c
9ffec58b-4b9d-4072-b944-5413bf7fcf07
SERVICE_ACTIVE_STATE
VNF2-DEPLOYM_XXXX_0_c8d98f0f-d874-45d0-af75-88a2d6fa82ea</vm_name>
<state>VM_SHUTOFF_STATE</state>
<snip>
```

**注:** 欠陥のあるコンポーネントが OSD 計算ノードで取り替えられるべきならコンポーネント置換を続行する前にサーバでメンテナンスに **Ceph** を入れて下さい。

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
```

```
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
```

```
[admin@osd-compute-0 ~]$ sudo ceph status
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_WARN
  noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
  monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-
controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
  election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-
controller-1,tb3-ultram-pod1-controller-2
  osdmap e194: 12 osds: 12 up, 12 in
  flags noout,norebalance,sortbitwise,require_jewel_osds
  pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
  1585 GB used, 11808 GB / 13393 GB avail
  704 active+clean
  client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

## Compute/OSD 計算ノードからの欠陥のあるコンポーネントを交換して下さい

電源遮断の指定されたサーバー。ステップは UCS C240 M4 サーバの欠陥のあるコンポーネントを交換するためにから参照することができます:

### [サーバコンポーネントの交換](#)

## VM を復元する

### 1.計算ノード ホスト SF だけ VM を包装して下さい

#### ESC からの SF VM リカバリ

- SF VM は新星リストのエラー状態にあります:

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
```

```
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
```

```
[admin@osd-compute-0 ~]$ sudo ceph status
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_WARN
  noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
  monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-
controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
  election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-
controller-1,tb3-ultram-pod1-controller-2
  osdmap e194: 12 osds: 12 up, 12 in
  flags noout,norebalance,sortbitwise,require_jewel_osds
  pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
  1585 GB used, 11808 GB / 13393 GB avail
  704 active+clean
client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

- ESC からの SF VM を回復して下さい:

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
```

```
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
```

```
[admin@osd-compute-0 ~]$ sudo ceph status
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_WARN
  noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
  monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-
controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
  election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-
controller-1,tb3-ultram-pod1-controller-2
  osdmap e194: 12 osds: 12 up, 12 in
  flags noout,norebalance,sortbitwise,require_jewel_osds
  pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
  1585 GB used, 11808 GB / 13393 GB avail
  704 active+clean
client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

- yangesc.log を監視して下さい:

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
```

```
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
```

```
[admin@osd-compute-0 ~]$ sudo ceph status
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_WARN
  noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
  monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-
controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
  election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-
controller-1,tb3-ultram-pod1-controller-2
  osdmap e194: 12 osds: 12 up, 12 in
  flags noout,norebalance,sortbitwise,require_jewel_osds
```

```
pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
1585 GB used, 11808 GB / 13393 GB avail
704 active+clean
client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

- SF カードが VNF のスタンバイ SF として搭載されるようにして下さい

## 2. Compute/OSD 計算ノード ホスト CF、ESC、EM および UAS を包装して下さい

### UAS VM のリカバリ

- 新星リストの UAS VM のステータスをチェックし、削除して下さい:

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
```

```
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
```

```
[admin@osd-compute-0 ~]$ sudo ceph status
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_WARN
noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-
controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-
controller-1,tb3-ultram-pod1-controller-2
osdmap e194: 12 osds: 12 up, 12 in
flags noout,norebalance,sortbitwise,require_jewel_osds
pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
1585 GB used, 11808 GB / 13393 GB avail
704 active+clean
client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

- autovnf-uas VM を回復するために、uas チェックスクリプト チェック状態を実行して下さい。それはエラーを報告する必要があります。それから再度実行して下さい ---抜けた UAS VM を作り直すためにオプションを固定して下さい:

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
```

```
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
```

```
[admin@osd-compute-0 ~]$ sudo ceph status
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_WARN
noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-
controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-
controller-1,tb3-ultram-pod1-controller-2
osdmap e194: 12 osds: 12 up, 12 in
flags noout,norebalance,sortbitwise,require_jewel_osds
pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
1585 GB used, 11808 GB / 13393 GB avail
704 active+clean
client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

- autovnf-uas へのログイン。数分間待てば UAS はよい状態に戻る必要があります:

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
```

```
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
```

```
[admin@osd-compute-0 ~]$ sudo ceph status
  cluster eb2bb192-b1c9-11e6-9205-525400330666
  health HEALTH_WARN
    noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
  monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-
controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
    election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-
controller-1,tb3-ultram-pod1-controller-2
  osdmap e194: 12 osds: 12 up, 12 in
    flags noout,norebalance,sortbitwise,require_jewel_osds
  pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
    1585 GB used, 11808 GB / 13393 GB avail
      704 active+clean
  client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

**注: uas-check.py なら ---修正は失敗します、このファイルをコピーし、再度動作する必要  
があるかもしれません。**

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
```

```
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
```

```
[admin@osd-compute-0 ~]$ sudo ceph status
  cluster eb2bb192-b1c9-11e6-9205-525400330666
  health HEALTH_WARN
    noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
  monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-
controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
    election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-
controller-1,tb3-ultram-pod1-controller-2
  osdmap e194: 12 osds: 12 up, 12 in
    flags noout,norebalance,sortbitwise,require_jewel_osds
  pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
    1585 GB used, 11808 GB / 13393 GB avail
      704 active+clean
  client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

## ESC VM のリカバリ

- 新星リストからの ESC VM のステータスをチェックし、削除して下さい:

```
[admin@osd-compute-0 ~]$ sudo ceph osd set norebalance
set norebalance
```

```
[admin@osd-compute-0 ~]$ sudo ceph osd set noout
set noout
```

```
[admin@osd-compute-0 ~]$ sudo ceph status
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_WARN
  noout,norebalance,sortbitwise,require_jewel_osds flag(s) set
monmap e1: 3 mons at {tb3-ultram-pod1-controller-0=11.118.0.40:6789/0,tb3-ultram-pod1-controller-1=11.118.0.41:6789/0,tb3-ultram-pod1-controller-2=11.118.0.42:6789/0}
election epoch 58, quorum 0,1,2 tb3-ultram-pod1-controller-0,tb3-ultram-pod1-controller-1,tb3-ultram-pod1-controller-2
osdmap e194: 12 osds: 12 up, 12 in
  flags noout,norebalance,sortbitwise,require_jewel_osds
pgmap v584865: 704 pgs, 6 pools, 531 GB data, 344 kobjects
  1585 GB used, 11808 GB / 13393 GB avail
    704 active+clean
client io 463 kB/s rd, 14903 kB/s wr, 263 op/s rd, 542 op/s wr
```

- AutoVNF UAS から、boot\_vm.py コマンド・ラインを ESC 例を作成するために見つけるとトランザクションのためのログの ESC 配備トランザクションが見つけば:

```
ubuntu@VNF2-uas-uas-0:~$ sudo -i
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show transaction
TX ID                                TX TYPE                                DEPLOYMENT ID
TIMESTAMP                            STATUS
-----
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b  vnf-deployment                         VNF2-DEPLOYMENT  2017-11-29T02:01:27.750692-00:00  deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b  vnf-deployment                         VNF2-ESC         2017-11-29T01:56:02.133663-00:00  deployment-success
```

```
VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
    <tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNF Deployment RPC triggered for deployment: VNF2-ESC, deactivate: 0
2017-11-29 01:56:02,179 - Notify deployment
..
2017-11-29 01:57:30,385 - Creating VNF 'VNF2-ESC-ESC-1' with [python //opt/cisco/vnf-staging/bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-dfd910436689 --net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-orchestration --os_auth_url http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username ***** --os_password ***** --bs_os_auth_url http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --bs_os_username ***** --bs_os_password ***** --esc_ui_startup false --esc_params_file /tmp/esc_params.cfg --encrypt_key ***** --user_pass ***** --user_confid_pass ***** --kad_vif eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3 172.168.10.6 --file root:0755:/opt/cisco/esc/esc-scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_volume_em_staging.sh --file root:0755:/opt/cisco/esc/esc-scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py --file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh
```

boot\_vm.py 行をシエル スクリプト ファイル ( esc.sh ) に保存し、正しい情報 ( 一般的に core/<PASSWORD> ) のすべてのユーザ名\*\*\*\*\*およびパスワード\*\*\*\*\*行をアップデートして下さい。 - encrypt\_key オプションを同様に取除く必要があります。 user\_pass および user\_confid\_pass に関しては、-ユーザ名 形式を使用する必要があります: パスワード ( 例-

admin: <PASSWORD> ) 。

- autovnf-uas VM に running-config および wget からの URL bootvm.py を bootvm.pyfile 見つけて下さい。 この場合、10.1.2.3 は自動 IT VM IP です:

```
ubuntu@VNF2-uas-uas-0:~$ sudo -i
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show transaction
TX ID                                TX TYPE                                DEPLOYMENT ID
TIMESTAMP                            STATUS
-----
-----
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b vnf-deployment                          VNF2-DEPLOYMENT 2017-11-
29T02:01:27.750692-00:00 deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b vnf-deployment                          VNF2-ESC        2017-11-
29T01:56:02.133663-00:00 deployment-success

VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
    <tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNF Deployment RPC triggered for deployment: VNF2-ESC,
deactivate: 0
2017-11-29 01:56:02,179 - Notify deployment
..
2017-11-29 01:57:30,385 - Creating VNF 'VNF2-ESC-ESC-1' with [python //opt/cisco/vnf-
staging/bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-
dfd910436689 --net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-
orchestration --os_auth_url http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username *****
--os_password ***** --bs_os_auth_url http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --
bs_os_username ***** --bs_os_password ***** --esc_ui_startup false --esc_params_file
/tmp/esc_params.cfg --encrypt_key ***** --user_pass ***** --user_confid_pass ***** --kad_vif
eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3 172.168.10.6
--file root:0755:/opt/cisco/esc/esc-
scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-
scripts/esc_volume_em_staging.sh --file root:0755:/opt/cisco/esc/esc-
scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-
keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh
```

```
ubuntu@VNF2-uas-uas-0:~$ sudo -i
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show transaction
TX ID                                TX TYPE                                DEPLOYMENT ID
TIMESTAMP                            STATUS
-----
-----
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b vnf-deployment                          VNF2-DEPLOYMENT 2017-11-
29T02:01:27.750692-00:00 deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b vnf-deployment                          VNF2-ESC        2017-11-
29T01:56:02.133663-00:00 deployment-success
```

```
VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
```

```

<tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
<log>2017-11-29 01:56:02,142 - VNFM Deployment RPC triggered for deployment: VNF2-ESC,
deactivate: 0
2017-11-29 01:56:02,179 - Notify deployment
..
2017-11-29 01:57:30,385 - Creating VNFM 'VNF2-ESC-ESC-1' with [python //opt/cisco/vnf-
staging/bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-
dfd910436689 --net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-
orchestration --os_auth_url http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username *****
--os_password ***** --bs_os_auth_url http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --
bs_os_username ***** --bs_os_password ***** --esc_ui_startup false --esc_params_file
/tmp/esc_params.cfg --encrypt_key ***** --user_pass ***** --user_confid_pass ***** --kad_vif
eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3 172.168.10.6
--file root:0755:/opt/cisco/esc/esc-
scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-
scripts/esc_volume_em_staging.sh --file root:0755:/opt/cisco/esc/esc-
scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-
keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh

```

- /tmp/esc\_params.cfg ファイルを作成して下さい:

```

ubuntu@VNF2-uas-uas-0:~$ sudo -i
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show transaction

```

TX ID	TX TYPE	DEPLOYMENT ID	TIMESTAMP	STATUS
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b	vnf-deployment	VNF2-DEPLOYMENT	2017-11-29T02:01:27.750692-00:00	deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b	vnfm-deployment	VNF2-ESC	2017-11-29T01:56:02.133663-00:00	deployment-success

```

VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml

```

```

<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
    <tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNFM Deployment RPC triggered for deployment: VNF2-ESC,
deactivate: 0
2017-11-29 01:56:02,179 - Notify deployment
..

```

```

2017-11-29 01:57:30,385 - Creating VNFM 'VNF2-ESC-ESC-1' with [python //opt/cisco/vnf-
staging/bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-
dfd910436689 --net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-
orchestration --os_auth_url http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username *****
--os_password ***** --bs_os_auth_url http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --
bs_os_username ***** --bs_os_password ***** --esc_ui_startup false --esc_params_file
/tmp/esc_params.cfg --encrypt_key ***** --user_pass ***** --user_confid_pass ***** --kad_vif
eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3 172.168.10.6
--file root:0755:/opt/cisco/esc/esc-
scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-
scripts/esc_volume_em_staging.sh --file root:0755:/opt/cisco/esc/esc-
scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-
keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh

```

- UAS ノードからの ESC を展開するためにシェル スクリプトを実行して下さい:

```
ubuntu@VNF2-uas-uas-0:~$ sudo -i
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show transaction
```

TX ID	TX TYPE	DEPLOYMENT ID	STATUS
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b	vnf-deployment	VNF2-DEPLOYMENT	2017-11-29T02:01:27.750692-00:00 deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b	vnfm-deployment	VNF2-ESC	2017-11-29T01:56:02.133663-00:00 deployment-success

```
VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
    <tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNF Deployment RPC triggered for deployment: VNF2-ESC,
deactivate: 0
2017-11-29 01:56:02,179 - Notify deployment
..
2017-11-29 01:57:30,385 - Creating VNF 'VNF2-ESC-ESC-1' with [python //opt/cisco/vnf-
staging/bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-
dfd910436689 --net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-
orchestration --os_auth_url http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username *****
--os_password ***** --bs_os_auth_url http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --
bs_os_username ***** --bs_os_password ***** --esc_ui_startup false --esc_params_file
/tmp/esc_params.cfg --encrypt_key ***** --user_pass ***** --user_confid_pass ***** --kad_vif
eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3 172.168.10.6
--file root:0755:/opt/cisco/esc/esc-
scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-
scripts/esc_volume_em_staging.sh --file root:0755:/opt/cisco/esc/esc-
scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-
keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh
```

- 新しい ESC へのログインはバックアップ状態を確認し、:

```
ubuntu@VNF2-uas-uas-0:~$ sudo -i
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show transaction
```

TX ID	TX TYPE	DEPLOYMENT ID	STATUS
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b	vnf-deployment	VNF2-DEPLOYMENT	2017-11-29T02:01:27.750692-00:00 deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b	vnfm-deployment	VNF2-ESC	2017-11-29T01:56:02.133663-00:00 deployment-success

```
VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
    <tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNF Deployment RPC triggered for deployment: VNF2-ESC,
```



```

deactivate: 0
2017-11-29 01:56:02,179 - Notify deployment
..
2017-11-29 01:57:30,385 - Creating VNF 'VNF2-ESC-ESC-1' with [python //opt/cisco/vnf-
staging/bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-
dfd910436689 --net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-
orchestration --os_auth_url http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username *****
--os_password ***** --bs_os_auth_url http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --
bs_os_username ***** --bs_os_password ***** --esc_ui_startup false --esc_params_file
/tmp/esc_params.cfg --encrypt_key ***** --user_pass ***** --user_confid_pass ***** --kad_vif
eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3 172.168.10.6
--file root:0755:/opt/cisco/esc/esc-
scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-
scripts/esc_volume_em_staging.sh --file root:0755:/opt/cisco/esc/esc-
scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-
keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh

```

## ESC からの CF および EM VM を回復して下さい

- 新星リストからの CF および EM VM のステータスをチェックして下さい。それらはエラー状態にある必要があります:

```

ubuntu@VNF2-uas-uas-0:~$ sudo -i
root@VNF2-uas-uas-0:~# confd_cli -u admin -C
Welcome to the ConfD CLI
admin connected from 127.0.0.1 using console on VNF2-uas-uas-0
VNF2-uas-uas-0#show transaction

```

TX ID	TX TYPE	DEPLOYMENT ID	STATUS
35eefc4a-d4a9-11e7-bb72-fa163ef8df2b	vnf-deployment	VNF2-DEPLOYMENT	2017-11-29T02:01:27.750692-00:00 deployment-success
73d9c540-d4a8-11e7-bb72-fa163ef8df2b	vnfm-deployment	VNF2-ESC	2017-11-29T01:56:02.133663-00:00 deployment-success

```

VNF2-uas-uas-0#show logs 73d9c540-d4a8-11e7-bb72-fa163ef8df2b | display xml
<config xmlns="http://tail-f.com/ns/config/1.0">
  <logs xmlns="http://www.cisco.com/usp/nfv/usp-autovnf-oper">
    <tx-id>73d9c540-d4a8-11e7-bb72-fa163ef8df2b</tx-id>
    <log>2017-11-29 01:56:02,142 - VNF Deployment RPC triggered for deployment: VNF2-ESC,
deactivate: 0
2017-11-29 01:56:02,179 - Notify deployment
..
2017-11-29 01:57:30,385 - Creating VNF 'VNF2-ESC-ESC-1' with [python //opt/cisco/vnf-
staging/bootvm.py VNF2-ESC-ESC-1 --flavor VNF2-ESC-ESC-flavor --image 3fe6b197-961b-4651-af22-
dfd910436689 --net VNF2-UAS-uas-management --gateway_ip 172.168.10.1 --net VNF2-UAS-uas-
orchestration --os_auth_url http://10.1.2.5:5000/v2.0 --os_tenant_name core --os_username *****
--os_password ***** --bs_os_auth_url http://10.1.2.5:5000/v2.0 --bs_os_tenant_name core --
bs_os_username ***** --bs_os_password ***** --esc_ui_startup false --esc_params_file
/tmp/esc_params.cfg --encrypt_key ***** --user_pass ***** --user_confid_pass ***** --kad_vif
eth0 --kad_vip 172.168.10.7 --ipaddr 172.168.10.6 dhcp --ha_node_list 172.168.10.3 172.168.10.6
--file root:0755:/opt/cisco/esc/esc-
scripts/esc_volume_em_staging.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-
scripts/esc_volume_em_staging.sh --file root:0755:/opt/cisco/esc/esc-
scripts/esc_vpc_chassis_id.py:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc_vpc_chassis_id.py
--file root:0755:/opt/cisco/esc/esc-scripts/esc-vpc-di-internal-
keys.sh:/opt/cisco/usp/uas/autovnf/vnfms/esc-scripts/esc-vpc-di-internal-keys.sh

```

- ESC マスターへのログインは、各々の影響を与えられた EM および CF VM のためのリカバリ vm 処理を実行します。忍耐強くであって下さい。ESC はリカバリ処理をスケジュールし、数分間起こらないかもしれませんが。yangesc.log を監視して下さい:

```
sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO <VM-Name>
```

```
[admin@VNF2-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO
VNF2-DEPLOYMENT-_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8
[sudo] password for admin:
```

Recovery VM Action

```
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=admin --
privKeyFile=/root/.ssh/confd_id_dsa --privKeyType=dsa --rpc=/tmp/esc_nc_cli.ZpRCGiieuW
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <ok/>
</rpc-reply>
```

```
[admin@VNF2-esc-esc-0 ~]$ tail -f /var/log/esc/yangesc.log
```

```
...
14:59:50,112 07-Nov-2017 WARN Type: VM_RECOVERY_COMPLETE
14:59:50,112 07-Nov-2017 WARN Status: SUCCESS
14:59:50,112 07-Nov-2017 WARN Status Code: 200
14:59:50,112 07-Nov-2017 WARN Status Msg: Recovery: Successfully recovered VM [VNF2-DEPLOYMENT-
_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8]
```

- 新しい EM へのログインは EM が状態稼働していることを確認し、:

```
sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO <VM-Name>
```

```
[admin@VNF2-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO
VNF2-DEPLOYMENT-_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8
[sudo] password for admin:
```

Recovery VM Action

```
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=admin --
privKeyFile=/root/.ssh/confd_id_dsa --privKeyType=dsa --rpc=/tmp/esc_nc_cli.ZpRCGiieuW
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <ok/>
</rpc-reply>
```

```
[admin@VNF2-esc-esc-0 ~]$ tail -f /var/log/esc/yangesc.log
```

```
...
14:59:50,112 07-Nov-2017 WARN Type: VM_RECOVERY_COMPLETE
14:59:50,112 07-Nov-2017 WARN Status: SUCCESS
14:59:50,112 07-Nov-2017 WARN Status Code: 200
14:59:50,112 07-Nov-2017 WARN Status Msg: Recovery: Successfully recovered VM [VNF2-DEPLOYMENT-
_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8]
```

- StarOS VNF にログインし、CF カードが STANDBY 状態にあることを確認して下さい

## ESC リカバリ失敗を処理して下さい

ESC が予想外状態による VM を開始しなければマスター ESC のリポートによって ESC スイッチオーバーを行う方法を Cisco は推奨します。ESC スイッチオーバーは約少しだけ時間がかかります。ステータスが稼働しているかどうか確認するために新しいマスター ESC のスクリプト「

health.sh」を実行して下さい。VMを開始し、VM状態を修復するためにESCをマスターして下さい。このリカバリタスクは5分程完了するためにかかります。

/var/log/esc/yangesc.log および /var/log/esc/escmanager.log を監視できます。5-7分後に回復されたVM gets はユーザ影響を与えられたVMの手動リカバリをすることを行く必要があることがわからなければ。

## 自動導入 設定アップデート

- AutoDeploy VM から、autodeploy.cfg を編集し、新しいものと古い計算サーバを取り替えて下さい。それからロードは confd\_cli で取り替えます。正常な配置非アクティブ化以降にこのステップが必要となります:

```
sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO <VM-Name>
```

```
[admin@VNF2-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO
VNF2-DEPLOYMENT-_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8
[sudo] password for admin:
```

Recovery VM Action

```
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=admin --
privKeyFile=/root/.ssh/confd_id_dsa --privKeyType=dsa --rpc=/tmp/esc_nc_cli.ZpRCGiieuW
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <ok/>
</rpc-reply>
```

```
[admin@VNF2-esc-esc-0 ~]$ tail -f /var/log/esc/yangesc.log
```

```
...
14:59:50,112 07-Nov-2017 WARN Type: VM_RECOVERY_COMPLETE
14:59:50,112 07-Nov-2017 WARN Status: SUCCESS
14:59:50,112 07-Nov-2017 WARN Status Code: 200
14:59:50,112 07-Nov-2017 WARN Status Msg: Recovery: Successfully recovered VM [VNF2-DEPLOYMENT-
_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8]
```

- コンフィギュレーション変更の後で uas-confd および autodeploy サービスを再開して下さい:

```
sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO <VM-Name>
```

```
[admin@VNF2-esc-esc-0 ~]$ sudo /opt/cisco/esc/esc-confd/esc-cli/esc_nc_cli recovery-vm-action DO
VNF2-DEPLOYMENT-_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8
[sudo] password for admin:
```

Recovery VM Action

```
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=127.0.0.1 --user=admin --
privKeyFile=/root/.ssh/confd_id_dsa --privKeyType=dsa --rpc=/tmp/esc_nc_cli.ZpRCGiieuW
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <ok/>
</rpc-reply>
```

```
[admin@VNF2-esc-esc-0 ~]$ tail -f /var/log/esc/yangesc.log
```

```
...
14:59:50,112 07-Nov-2017 WARN Type: VM_RECOVERY_COMPLETE
14:59:50,112 07-Nov-2017 WARN Status: SUCCESS
14:59:50,112 07-Nov-2017 WARN Status Code: 200
14:59:50,112 07-Nov-2017 WARN Status Msg: Recovery: Successfully recovered VM [VNF2-DEPLOYMENT-
_VNF2-D_0_a6843886-77b4-4f38-b941-74eb527113a8]
```

# コンポーネント RMA - Controller ノード

## 事前点検

- OSPD から、コントローラへのログオンはパソコンがよい州に 3 コントローラ オンラインすべてであり、ガレラ川がマスターとしてすべての 3 人のコントローラを示すことを確認し。

**注:** 健全なクラスタは 2 つのアクティブ コントローラを必要とします従って残る 2 人のコントローラがオンラインおよびアクティブであることを確認して下さい。

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs status
Cluster name: tripleo_cluster
Stack: corosync
Current DC: pod1-controller-2 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum
Last updated: Mon Dec  4 00:46:10 2017                Last change: Wed Nov 29 01:20:52
2017 by hacluster via crmd on pod1-controller-0
3 nodes and 22 resources configured
```

**Online:** [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]

Full list of resources:

```
ip-11.118.0.42 (ocf::heartbeat:IPAddr2):          Started pod1-controller-1
ip-11.119.0.47 (ocf::heartbeat:IPAddr2):          Started pod1-controller-2
ip-11.120.0.49 (ocf::heartbeat:IPAddr2):          Started pod1-controller-1
ip-192.200.0.102 (ocf::heartbeat:IPAddr2):        Started pod1-controller-2
Clone Set: haproxy-clone [haproxy]
  Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: galera-master [galera]
  Masters: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
ip-11.120.0.47 (ocf::heartbeat:IPAddr2):          Started pod1-controller-2
Clone Set: rabbitmq-clone [rabbitmq]
  Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: redis-master [redis]
  Masters: [ pod1-controller-2 ]
  Slaves: [ pod1-controller-0 pod1-controller-1 ]
ip-10.84.123.35 (ocf::heartbeat:IPAddr2):          Started pod1-controller-1
openstack-cinder-volume (systemd:openstack-cinder-volume): Started pod1-
controller-2
my-ipmilan-for-pod1-controller-0 (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-pod1-controller-1 (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-pod1-controller-2 (stonith:fence_ipmilan): Started pod1-controller-0
```

Daemon Status:

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

## メンテナンス モードにコントローラ クラスタを移動して下さい

- スタンバイでアップデートされるコントローラのパソコン クラスタを使用して下さい:

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs cluster standby
```

- パソコン ステータスを再度チェックし、パソコンがこのノードで停止されてクラスタ化する

ようにして下さい:

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs status
Cluster name: tripleo_cluster
Stack: corosync
Current DC: pod1-controller-2 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum
Last updated: Mon Dec  4 00:48:24 2017                Last change: Mon Dec  4 00:48:18
2017 by root via crm_attribute on pod1-controller-0
3 nodes and 22 resources configured
```

**Node pod1-controller-0: standby**

**Online: [ pod1-controller-1 pod1-controller-2 ]**

Full list of resources:

```
ip-11.118.0.42 (ocf::heartbeat:IPAddr2): Started pod1-controller-1
ip-11.119.0.47 (ocf::heartbeat:IPAddr2): Started pod1-controller-2
ip-11.120.0.49 (ocf::heartbeat:IPAddr2): Started pod1-controller-1
ip-192.200.0.102 (ocf::heartbeat:IPAddr2): Started pod1-controller-2
Clone Set: haproxy-clone [haproxy]
  Started: [ pod1-controller-1 pod1-controller-2 ]
  Stopped: [ pod1-controller-0 ]
Master/Slave Set: galera-master [galera]
  Masters: [ pod1-controller-1 pod1-controller-2 ]
  Slaves: [ pod1-controller-0 ]
ip-11.120.0.47 (ocf::heartbeat:IPAddr2): Started pod1-controller-2
Clone Set: rabbitmq-clone [rabbitmq]
  Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: redis-master [redis]
  Masters: [ pod1-controller-2 ]
  Slaves: [ pod1-controller-1 ]
  Stopped: [ pod1-controller-0 ]
ip-10.84.123.35 (ocf::heartbeat:IPAddr2): Started pod1-controller-1
openstack-cinder-volume (systemd:openstack-cinder-volume): Started pod1-
controller-2
my-ipmilan-for-pod1-controller-0 (stonith:fence_ipmilan): Started pod1-controller-1
my-ipmilan-for-pod1-controller-1 (stonith:fence_ipmilan): Started pod1-controller-1
my-ipmilan-for-pod1-controller-2 (stonith:fence_ipmilan): Started pod1-controller-2
```

Daemon Status:

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

また、他の2人のコントローラのパソコンステータスはスタンバイとしてノードを示す必要があります。

## Controller ノードからの欠陥のあるコンポーネントを交換して下さい

電源遮断の指定されたサーバー。ステップは UCS C240 M4 サーバの欠陥のあるコンポーネントを交換するためにから参照することができます:

### [サーバコンポーネントの交換](#)

## サーバの電源

- サーバの電源はサーバがアップすることを確認し、:

```
[stack@tb5-ospd ~]$ source stackrc
[stack@tb5-ospd ~]$ nova list |grep pod1-controller-0
| 1ca946b8-52e5-4add-b94c-4d4b8a15a975 | pod1-controller-0 | ACTIVE | - | Running
| ctlplane=192.200.0.112 |
```

- 影響を与えられたコントローラにログインして下さい、**unstandby** の使用とスタンバイ モードを取除いて下さい。コントローラがオンライン クラスタの来、ガレラ川がマスターとしてすべての 3 人のコントローラを示すことを確認して下さい。これは数分かかるかもしれません:

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs cluster unstandby
```

```
[heat-admin@pod1-controller-0 ~]$ sudo pcs status
```

```
Cluster name: tripleo_cluster
Stack: corosync
Current DC: pod1-controller-2 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum
Last updated: Mon Dec 4 01:08:10 2017 Last change: Mon Dec 4 01:04:21
2017 by root via crm_attribute on pod1-controller-0
3 nodes and 22 resources configured
```

```
Online: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
```

Full list of resources:

```
ip-11.118.0.42 (ocf::heartbeat:IPAddr2): Started pod1-controller-1
ip-11.119.0.47 (ocf::heartbeat:IPAddr2): Started pod1-controller-2
ip-11.120.0.49 (ocf::heartbeat:IPAddr2): Started pod1-controller-1
ip-192.200.0.102 (ocf::heartbeat:IPAddr2): Started pod1-controller-2
Clone Set: haproxy-clone [haproxy]
Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: galera-master [galera]
Masters: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
ip-11.120.0.47 (ocf::heartbeat:IPAddr2): Started pod1-controller-2
Clone Set: rabbitmq-clone [rabbitmq]
Started: [ pod1-controller-0 pod1-controller-1 pod1-controller-2 ]
Master/Slave Set: redis-master [redis]
Masters: [ pod1-controller-2 ]
Slaves: [ pod1-controller-0 pod1-controller-1 ]
ip-10.84.123.35 (ocf::heartbeat:IPAddr2): Started pod1-controller-1
openstack-cinder-volume (systemd:openstack-cinder-volume): Started pod1-
controller-2
my-ipmilan-for-pod1-controller-0 (stonith:fence_ipmilan): Started pod1-controller-1
my-ipmilan-for-pod1-controller-1 (stonith:fence_ipmilan): Started pod1-controller-1
my-ipmilan-for-pod1-controller-2 (stonith:fence_ipmilan): Started pod1-controller-2
```

Daemon Status:

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

- 健全な状態にあることいくつかの **ceph** のようなモニタ サービスをチェックできます:

```
[heat-admin@pod1-controller-0 ~]$ sudo ceph -s
cluster eb2bb192-b1c9-11e6-9205-525400330666
health HEALTH_OK
monmap e1: 3 mons at {pod1-controller-0=11.118.0.10:6789/0,pod1-controller-
1=11.118.0.11:6789/0,pod1-controller-2=11.118.0.12:6789/0}
election epoch 70, quorum 0,1,2 pod1-controller-0,pod1-controller-1,pod1-controller-2
osdmap e218: 12 osds: 12 up, 12 in
flags sortbitwise,require_jewel_osds
```

pgmap v2080888: 704 pgs, 6 pools, 714 GB data, 237 kobjects  
2142 GB used, 11251 GB / 13393 GB avail  
704 active+clean  
client io 11797 kB/s wr, 0 op/s rd, 57 op/s wr