

Ultra 6.3 릴리스 이후 요소 관리자 - 아키텍처 변경 사항 및 EM 문제 해결 방법

목차

[소개](#)

[사전 요구 사항](#)

[요구 사항](#)

[사용되는 구성 요소](#)

[배경 정보](#)

[약어](#)

[Ultra 6.3 릴리스 이후 Ultra Element Manager - 아키텍처 변경](#)

[6.3 릴리스로 구성 가능한 UEM VM 인스턴스 수](#)

[동물원키퍼 분포](#)

[HA용 Keepalived 소개](#)

[NCS 프로세스에서 SCM 분리](#)

[EM 서비스는 마스터 노드에서만 실행됩니다.](#)

[요소 관리자 관련 문제 해결 단계](#)

[VM 2개가 포함된 UEM 설정 - 프로세스 시작 시퀀스 및 로그 위치](#)

[3개의 VM을 사용하는 UEM 설정 - 프로세스 시작 시퀀스 및 로그 위치](#)

[UEM 프로세스 요약](#)

소개

이 문서에서는 6.3 UltraM 릴리스의 일부로 도입된 EM(Element Manager) 아키텍처의 변경 사항에 대해 설명합니다.

사전 요구 사항

요구 사항

다음 주제에 대한 지식을 보유하고 있으면 유용합니다.

- STARO
- Ultra-M 기본 아키텍처

사용되는 구성 요소

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다. 이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다. 네트워크가 작동 중인 경우 모든 명령의 잠재적인 영향을 이해해야 합니다.

배경 정보

Ultra 6.3 릴리스 이전에는 Ultra Element Manager가 작동하려면 3개의 UEM VM을 생성해야 했습니다. 3번째 동물은 사용하지 않았으며 동물원키퍼 클러스터를 구성하는 것을 돕기 위해 그곳에 있었다. 6.3 릴리스부터 이 설계가 변경되었습니다.

약어

약어:

VNF	가상 네트워크 기능
CF	제어 기능
SF	서비스 기능
ESC	Elastic Service Controller
VIM	가상 인프라 관리자
VM	가상 머신
EM	요소 관리자
UAS	Ultra Automation 서비스
UUID	보편적으로 고유한 Identifier
ZK	동물원 키퍼

Ultra 6.3 릴리스 이후 Ultra Element Manager - 아키텍처 변경

이 문서에서는 6.3 UltraM 릴리스의 일부로 도입된 다음 5가지 변경 사항에 대해 설명합니다.

6.3 릴리스로 구성 가능한 UEM VM 인스턴스 수

6.3 릴리스 이전에는 UEM VM 3개가 필수입니다. 코어 테넌트 파일이 소싱된 후 nova 목록을 사용할 수 있습니다.

```
[root@POD]# openstack server list --all
+-----+-----+-----+-----+-----+
| ID | Name | Status | Networks | Image Name |
+-----+-----+-----+-----+-----+
.....
| fae2d54a-96c7-4199-a412-155e6c029082 | vpc-LAASmme-em-3 | ACTIVE | orch=192.168.12.53;
mgmt=192.168.11.53 | ultra-em |
| c89a3716-9028-4835-9237-759166b5b7fb | vpc-LAASmme-em-2 | ACTIVE | orch=192.168.12.52;
mgmt=192.168.11.52 | ultra-em |
| 5f8cda2c-657a-4ba1-850c-805518e4bc18 | vpc-LAASmme-em-1 | ACTIVE | orch=192.168.12.51;
mgmt=192.168.11.51 | ultra-em |
이 컨피그레이션 스냅샷(vnf.conf 파일에서)이 사용되었습니다.
```

```
vnfc em
health-check enabled
health-check probe-frequency 10
health-check probe-max-miss 6
health-check retry-count 6
health-check recovery-type restart-then-redeploy
health-check boot-time 300
vdu vdu-id em
number-of-instances 1 --> HERE, this value was previously ignored in pre 6.3 releases
connection-point eth0
...
```

이 명령에 지정된 인스턴스 수에 관계없이 스핀된 VM 수는 항상 3입니다. 즉, 인스턴스 수 값이 무시되었습니다.

6.3부터 이 값이 변경됩니다. 구성된 값은 2 또는 3입니다.

2를 구성하면 2개의 UEM VM이 생성됩니다.

3을 구성하면 3개의 UEM VM이 생성됩니다.

```
vnfc em
health-check enabled
health-check probe-frequency 10
health-check probe-max-miss 6
health-check retry-count 3
health-check recovery-type restart
health-check boot-time 300
vdu vdu-id vdu-em
vdu image ultra-em
vdu flavor em-flavor
number-of-instances 2 --> HERE
connection-point eth0
....
```

이 컨피그레이션을 수행하면 nova 목록과 같이 2개의 VM이 생성됩니다.

```
[root@POD]# openstack server list --all
+-----+-----+-----+-----+-----+
| ID | Name | Status | Networks | Image Name |
+-----+-----+-----+-----+-----+
.....
| fae2d54a-96c7-4199-a412-155e6c029082 | vpc-LAASmme-em-3 | ACTIVE | orch=192.168.12.53;
mgmt=192.168.11.53 | ultra-em |
| c89a3716-9028-4835-9237-759166b5b7fb | vpc-LAASmme-em-2 | ACTIVE | orch=192.168.12.52;
mgmt=192.168.11.52 | ultra-em |
```

그러나 3개의 IP 주소 요구 사항은 동일하게 유지됩니다. 즉, config(vnf.conf 파일)의 EM에서 3개의 IP 주소는 여전히 필수입니다.

```
vnfc em
health-check enabled
health-check probe-frequency 10
health-check probe-max-miss 6
health-check retry-count 3
health-check recovery-type restart
health-check boot-time 300
vdu vdu-id vdu-em
vdu image ultra-em
vdu flavor em-flavor
number-of-instances 2 ---> NOTE NUMBER OF INSTANCES is 2
connection-point eth0
virtual-link service-vl orch
virtual-link fixed-ip 172.x.y.51 --> IP #1
!
virtual-link fixed-ip 172.x.y.52 --> IP #2
!
```

virtual-link fixed-ip 172.x.y.53 --> IP #3

!

ZK가 ZK의 3개 인스턴스를 작동하려면 이 작업이 필요합니다. 모든 인스턴스에는 IP 주소가 필요합니다. 세 번째 인스턴스가 효과적으로 사용되지 않더라도 세 번째 IP는 세 번째 Arbiter ZK 인스턴스에 할당됩니다(자세한 내용은 Diff.2 참조).

오케스트레이션 네트워크에 미치는 영향은 무엇입니까?

오케스트레이션 네트워크에는 항상 3개의 포트가 생성되어(언급된 3개의 IP 주소를 바인딩합니다)

```
[root@POD# neutron port-list | grep -em_
```

```
| 02d6f499-b060-469a-b691-ef51ed047d8c | vpc-LAASmme-em_vpc-LA_0_70de6820-9a86-4569-b069-46f89b9e2856 | fa:16:3e:a4:9a:49 | {"subnet_id": "bf5dea3d-cd2f-4503-a32d-5345486d66dc", "ip_address": "192.168.12.52"} |
| 0edcb464-cd7a-44bb-b6d6-07688a6c130d | vpc-LAASmme-em_vpc-LA_0_2694b73a-412b-4103-aac2-4be2c284932c | fa:16:3e:80:eb:2f | {"subnet_id": "bf5dea3d-cd2f-4503-a32d-5345486d66dc", "ip_address": "192.168.12.51"} |
| 9123f1a8-b3ea-4198-9ea3-1f89f45dfe74 | vpc-LAASmme-em_vpc-LA_0_49ada683-a5ce-4166-aeb5-3316fe1427ea | fa:16:3e:5c:17:d6 | {"subnet_id": "bf5dea3d-cd2f-4503-a32d-5345486d66dc", "ip_address": "192.168.12.53"} |
```

동물원키퍼 분포

6.3ZK 이전의 ZK는 클러스터를 구성하는 데 사용되었으므로 이 요구 사항은 세 번째 VM에 대한 것입니다.

이 요구 사항은 변경되지 않았습니다. 그러나 2개의 UEM VM이 사용되는 설정의 경우 3번째 ZK 인스턴스가 동일한 VM 집합에 호스팅됩니다.

3 UEM VM을 사용하여 6.3 이전 및 6.3 이후 설정:

UEM VM1:호스팅 Zk 인스턴스 1

UEM VM2:호스팅 Zk 인스턴스 2

UEM VM3:호스팅 Zk 인스턴스 3

6.3 이상에서는 VM 2개만 사용:

UEM VM1:호스팅 Zk 인스턴스 1 및 Zk 인스턴스 3

UEM VM2:호스팅 Zk 인스턴스 2

UEM VM3:존재하지 않음

자세한 그래픽 표시는 이 문서 하단에 있는 그림 1을 참조하십시오.

Useful Zk commands:

To see Zk mode (leader/follower):

```
/opt/cisco/usp/packages/zookeeper/current/bin/zkServer.sh status
```

```
ZooKeeper JMX enabled by default
```

```
Using config: /opt/cisco/usp/packages/zookeeper/current/bin/./conf/zoo.cfg
```

```
Mode: leader
```

To check if Zk is running:

```
echo stat | nc IP_ADDRESS 2181
```

How to find the Ip address of Zk instance:

Run 'ip addr' from EM

In the /opt/cisco/em/config/ip.txt there are all the 3IP's

From vnf.conf file

From 'nova list' look for orchestration IP

For 2 EM's the arbiter IP can be found also in /opt/cisco/em/config/proxy-params.txt

How to check status of the Zk instance:

```
echo stat | nc 192.168.12.51 2181 | grep Mode
```

```
Mode: follower
```

You can run this command from one Zk for all other Zk instances (even they are on different VM)!

To connect to the Zk cli - now must use the IP (rather than localhost earlier):

```
/opt/cisco/usp/packages/zookeeper/current/bin/zkCli.sh -server
```

Some useful command you can run once you connect to ZkCli:

```
ls /config/vdus/control-function
ls /config/element-manager
ls /
ls /log
ls /stat
get /config/vdus/session-function/BOOTxx
```

HA용 Keepalived 소개

이전 릴리스에서는 마스터 EM을 결정하는 데 사용되는 ZK 리더 선택 프레임워크가 사용되었습니다. Cisco가 보존 프레임워크로 전환함에 따라 더 이상 그렇지 않습니다.

유지된 내용 및 작동 방식

Proven은 Linux 시스템 및 Linux 기반 인프라에 대한 로드 밸런싱 및 고가용성에 사용되는 [Linux 기반](#) 소프트웨어입니다.

HA의 경우 ESC에서 이미 사용되고 있습니다.

EM에서 Keepalived는 NCS를 Zk 클러스터 상태에서 분리하는 데 사용됩니다.

보존 프로세스는 EM의 처음 두 인스턴스에서만 실행되며 NCS 프로세스의 마스터 상태를 결정합니다.

To check if the keepalived process is running:

```
ps -aef | grep keepalived
```

(must return the process ID)

왜 바뀌죠?

이전 구현에서 (NCS/SCM) 마스터 노드 선택은 Zk 클러스터 상태와 긴밀하게 통합되었습니다(Zk 데이터베이스의 /em을 잠글 첫 번째 인스턴스는 마스터로 선택됨). 이렇게 하면 Zk가 클러스터와의 연결이 끊어질 때 문제가 발생합니다.

Keepalived는 VM을 기준으로 활성/대기 UEM 클러스터를 유지하는 데 사용됩니다.

NCS는 구성 데이터를 유지 관리합니다.
Zookeeper는 운영 데이터를 유지 관리합니다.

NCS 프로세스에서 SCM 분리

6.3 이전 릴리스에서는 SCM 구성 요소가 NCS와 번들로 제공되었습니다. 즉, NCS가 시작되면 SCM도 시작됩니다(그 결과). 이번 릴리즈에서는 이 기능이 분리되었으며 SCM은 그 자체로 별도의 프로세스입니다.

Commands to check the NCS and SCM services & processes.

To be executed from the ubuntu command line

```
ps -aef | grep ncs  
ps -aef | grep scm
```

```
sudo service show ncs  
sudo service scm status
```

EM 서비스는 마스터 노드에서만 실행됩니다.

6.3 릴리스 이전의 UEM 서비스는 마스터/슬레이브에서 모두 실행됩니다. 현재 6.3 서비스는 마스터 노드에서만 실행됩니다. 그러면 **show ems**에 표시되는 출력에 영향을 미칩니다. 6.3부터는 이 명령을 사용하여 UEM CLI에 로그인한 노드를 하나만 볼 수 있습니다.

```
root@vpc-em-2:/var/log# sudo -i  
root@vpc-em-2:~# ncs_cli -u admin -C
```

```
admin connected from 127.0.0.1 using console on vpc-LAASmme-em-2
```

```
admin@scm# show ems
```

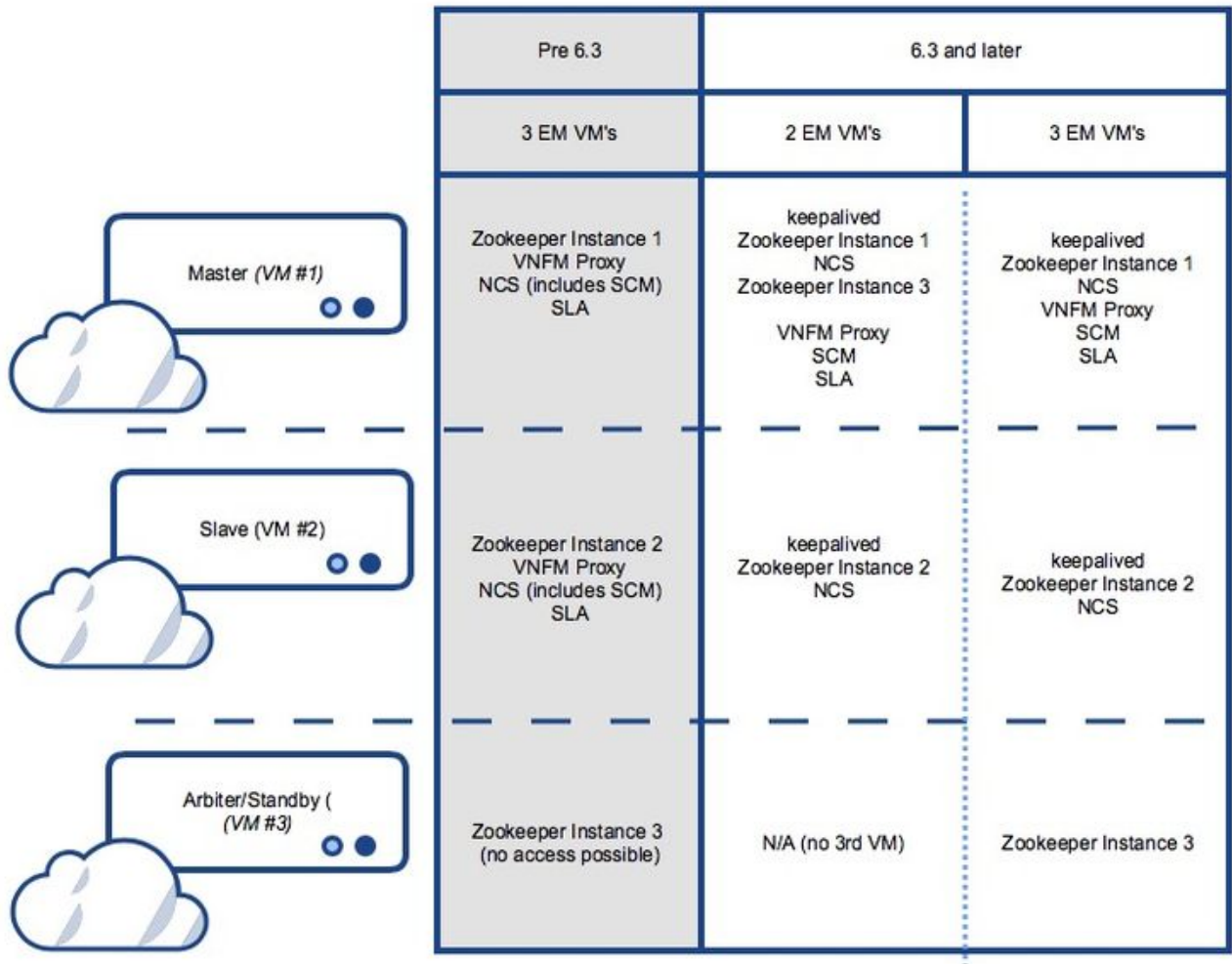
```
EM VNF
```

```
ID SLA SCM PROXY VERSION
```

```
-----  
52 UP UP UP 6.3.0 ==> HERE Only one EM instance is seen. In previous releases you were able to see 2 instances.
```

사실상 모든 서비스는 NCS를 제외하고 마스터 노드에서 실행되며 이는 NCS 요구 사항 때문입니다.

이 그림에서는 Ultra Element Manager에 대해 가능한 서비스 및 VM 배포의 요약은 보여 줍니다.



요소 관리자 관련 문제 해결 단계

부팅 중에 시작 시퀀스는 다음과 같습니다.

VM 2개가 포함된 UEM 설정 - 프로세스 시작 시퀀스 및 로그 위치

마스터 UEM:

- 유지 관리
- 주키퍼
- NCS
- Zookeeper(3번째) 인스턴스
- VNFM-프록시
- SCM
- SLA

슬레이브 UEM:

- 유지 관리
- 주키퍼
- NCS

3개의 VM을 사용하는 UEM 설정 - 프로세스 시작 시퀀스 및 로그 위치

마스터 UEM:

- 유지 관리
- 주키퍼
- NCS
- VNFM-프록시
- SCM
- SLA

슬레이브 UEM:

- 유지 관리
- 주키퍼
- NCS

세 번째 UEM:

- 주키퍼

UEM 프로세스 요약

실행해야 하는 UEM 프로세스의 요약입니다.

`ps -aef`를 사용하여 상태를 확인합니다. | `grep xx`

```
유지 관리  
중재자  
scm  
sla  
동물원.cfg  
ncs
```

서비스 `xx` 상태로 상태를 확인할 수 있습니다. 여기서 `xx`:

```
조키퍼  
프록시  
scm  
sla  
체크  
ncs
```