

Element Manager após a versão Ultra 6.3 - Alterações de arquitetura e como solucionar problemas do EM

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Componentes Utilizados](#)

[Informações de Apoio](#)

[Abreviaturas](#)

[Ultra Element Manager após a versão Ultra 6.3 - Alterações na arquitetura](#)

[O número de instâncias de VM do UEM é configurável a partir da versão 6.3](#)

[Distribuição do ZooKeeper](#)

[Introdução à HA](#)

[Separe o SCM do processo do NCS](#)

[O serviço EM é executado somente no nó mestre](#)

[Etapas para Solucionar Problemas Relacionados ao Element Manager](#)

[Configuração do UEM com 2 VMs - Sequência de inicialização do processo e local do registro](#)

[Configuração do UEM com 3 VMs - Sequência de inicialização do processo e local do registro](#)

[Resumo dos processos do UEM](#)

Introduction

Este documento descreve as alterações na arquitetura do Element Manager (EM) introduzidas como parte da versão 6.3 UltraM.

Prerequisites

Requirements

A Cisco recomenda que você tenha conhecimento destes tópicos:

- STAROs
- Arquitetura básica Ultra-M

Componentes Utilizados

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. Se a rede estiver ativa, certifique-se de que você entenda o impacto potencial de qualquer comando.

Informações de Apoio

Antes da versão Ultra 6.3, para que o Ultra Element Manager funcionasse, era necessário criar 3 VMs UEM. O terceiro não estava em uso e estava lá para ajudar a formar o cluster do ZooKeeper. A partir da versão 6.3, este design mudou.

Abreviaturas

Abreviaturas usadas neste artigo:

VNF	Função de rede virtual
CF	Função de controle
SF	Função de serviço
ESC	Controlador de serviço elástico
VIM	Virtual Infrastructure Manager
VM	Máquina virtual
EM	Gestor de Elementos
UAS	Ultra Automation Services
UUID	Identificador de ID universal exclusivo
ZK	Zoo Keeper

Ultra Element Manager após a versão Ultra 6.3 - Alterações na arquitetura

Este documento descreve estas 5 alterações que são introduzidas como parte da versão 6.3 UltraM:

O número de instâncias de VM do UEM é configurável a partir da versão 6.3

Antes da versão 6.3, 3 VM UEM eram obrigatórias. Você pode ver que com a **nova lista** após a origem do arquivo de espaço principal:

```
[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 |
```

Este instantâneo de configuração (do arquivo vnf.conf) foi usado:

```
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
...
```

Independentemente do número de instâncias especificadas neste comando, o número de VMs spun sempre foi 3. Em outras palavras, o valor de número de instâncias foi ignorado.

A partir da versão 6.3, isso é alterado - o valor configurado pode ser 2 ou 3.

Quando você configura 2, as 2 VMs UEM são criadas.

Quando você configura 3, as 3 VMs UEM são criadas.

```
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
....
```

Essa configuração resultaria em 2 VMs, conforme visto na [lista nova](#).

```
[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 |
```

Observe, no entanto, que 3 requisitos de endereço IP permaneceram os mesmos. Ou seja, na parte **EM** da configuração (arquivo vnf.conf) o endereço IP 3 ainda é **obrigatório**:

```
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
!

```

Isso é necessário para que ZK trabalhe 3 instâncias de ZK. Cada instância requer um endereço IP. Embora a terceira instância não seja usada com eficiência, o 3º IP é alocado para a terceira, a chamada instância ZK **Arbiter** (*consulte Diff.2 para obter mais explicações*).

Que efeito isso tem na rede de orquestração?

Sempre haverá 3 portas criadas na rede de orquestração (para ligar os 3 endereços IP mencionados).

```

[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"} |

```

Distribuição do ZooKeeper

Antes da 6,3 ZK foi usada para formar o cluster, portanto este requisito é para a 3ª VM. Esse requisito não mudou. No entanto, para as configurações em que 2 VMs UEM são usadas, uma terceira instância ZK é hospedada no mesmo conjunto de VMs:

Antes da 6.3 e depois da 6.3 em uma configuração com 3 VMs UEM:

VM1 UEM: hospedando a instância Zk 1

VM UEM2: hospedando a instância Zk 2

UEM VM3: hospedando a instância Zk 3

Na versão 6.3 e posterior, onde somente 2 VMs:

VM1 UEM: hospedando a instância Zk 1 e a instância Zk 3

VM UEM2: hospedando a instância Zk 2

UEM VM3: não existe

Veja a figura 1. na parte inferior deste artigo para uma representação gráfica detalhada.

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
```

Introdução à HA

Com as versões anteriores, a estrutura de eleição do líder da ZK foi usada para determinar o mestre EM. Esse não é mais o caso, pois a Cisco passou para a estrutura mantida.

O que é mantido e como funciona?

Keepalived é o [software baseado em Linux](#) usado para balanceamento de carga e alta disponibilidade para sistemas Linux e infraestruturas baseadas em Linux.

Ele já é usado em ESC para HA.

Em EM, Keepalived é usado para separar o NCS do estado de cluster Zk.

O processo mantido em funcionamento é executado somente nas duas primeiras instâncias do

EM e determinaria o estado mestre para o processo NCS.

To check if the keepalived process is running:

```
ps -aef | grep keepalived  
(must return the process ID)
```

Por que mudar?

Em uma implementação anterior, a seleção do nó mestre (NCS/SCM) foi intimamente integrada com o status do cluster Zk (a primeira instância a ser bloqueada no banco de dados Zk foi eleita master). Isso cria problemas quando Zk perde a conectividade com o cluster.

O Keepalived é usado para manter o cluster UEM Ativo/Standby com base em VM.

O NCS mantém os dados de configuração.

O Zookeeper mantém os dados operacionais.

Separe o SCM do processo do NCS

Nas versões anteriores à 6.3, o componente SCM foi incluído no pacote com o NCS. Isso significa que, quando o NCS começou, o SCM também começou (como consequência). Nesta versão, isso agora é dissociado e o SCM é um processo separado para si mesmo.

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
```

O serviço EM é executado somente no nó mestre

Antes da versão 6.3, os serviços UEM são executados no Master/Slave. A partir de 6.3, os serviços são executados somente no nó mestre. Isso afetaria a saída exibida em **show ems**. A partir da versão 6.3, espera-se ver apenas um nó (mestre) com esse comando, uma vez conectado à CLI do UEM:

```
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
```

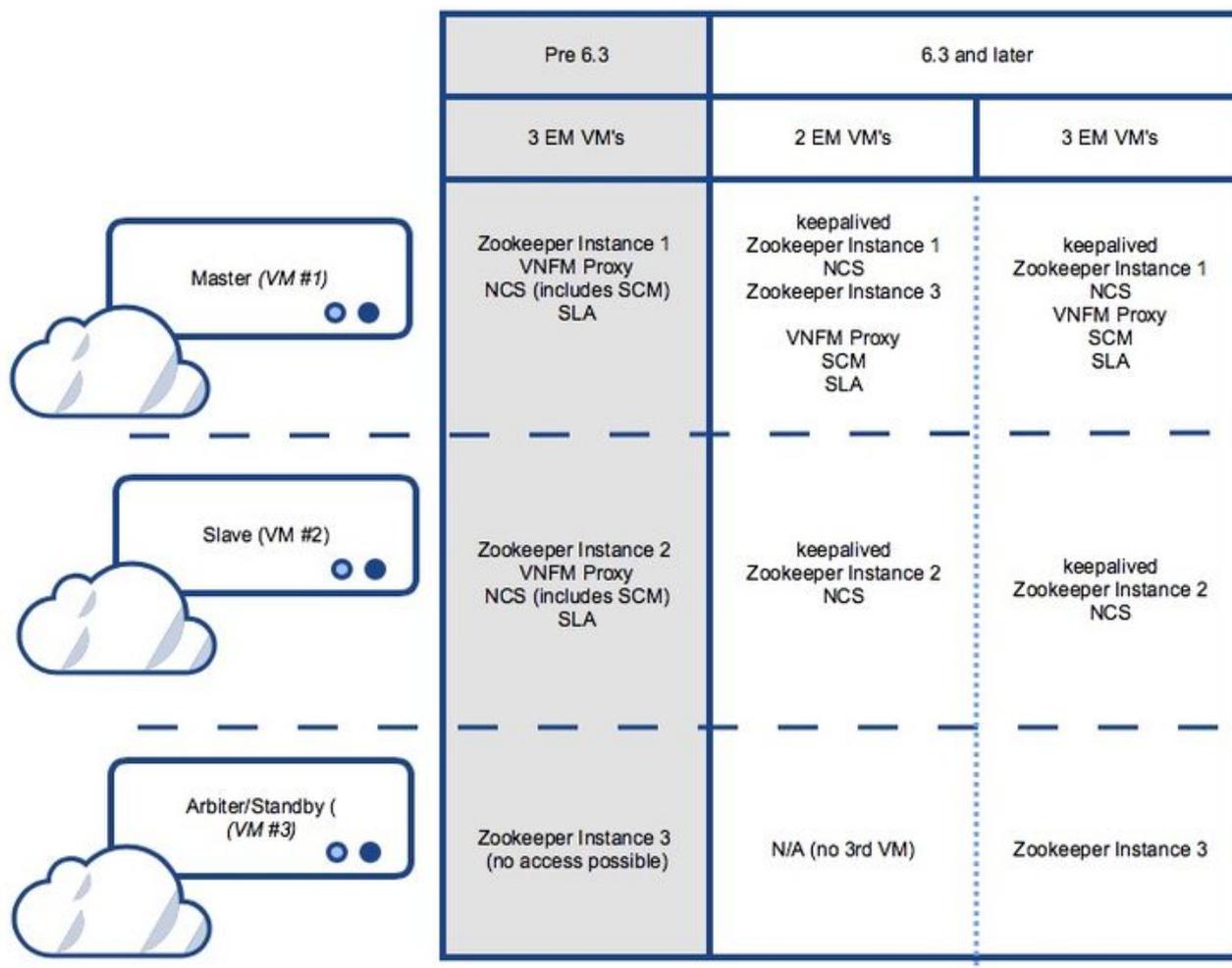
```
ID SLA SCM PROXY VERSION
```

```
-----
```

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

Efetivamente, todos os serviços seriam executados no nó mestre, com exceção do NCS, e isso se deve aos requisitos do NCS.

Esta imagem mostra o resumo dos possíveis serviços e distribuição de VM para o Ultra Element Manager



Etapas para Solucionar Problemas Relacionados ao Element Manager

Durante a inicialização, esta é a sequência de inicialização:

Configuração do UEM com 2 VMs - Sequência de inicialização do processo e local do registro

UEM mestre:

- manutenção
- Zookeeper
- NCS
- Instância de árbitro (3^o) do Zookeeper
- VNFM-Proxy
- SCM
- SLA

Slave UEM:

- manutenção
- Zookeeper
- NCS

Configuração do UEM com 3 VMs - Sequência de inicialização do processo e local do registro

UEM mestre:

- manutenção
- Zookeeper
- NCS
- VNFM-Proxy
- SCM
- SLA

Slave UEM:

- manutenção
- Zookeeper
- NCS

3º UEM:

- Zookeeper

Resumo dos processos do UEM

Este é o resumo dos processos UEM que você precisa executar.

Você verifica o status com **ps -aef | grep xx**

manutenção
árbitro
scm
sla
zoo.cfg
ncs

Você pode verificar o status com o **status xx do serviço**, onde xx:

zooarbitro
proxy
scm
sla
zk
ncs