

# Procedura per la gestione del nodo arbitro nel set di repliche CPS

## Sommario

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Premesse](#)

[Problema](#)

[Procedura per la gestione dell'arbitro in un set di repliche](#)

## Introduzione

In questo documento viene descritta la procedura per gestire il nodo Arbitro nel set di repliche di Cisco Policy Suite (CPS).

## Prerequisiti

### Requisiti

Cisco raccomanda la conoscenza dei seguenti argomenti:

- Linux
- CPS
- MongoDB

**Nota:** Cisco consiglia di disporre dell'accesso privilegiato alla directory principale di CPS CLI.

### Componenti usati

Le informazioni fornite in questo documento si basano sulle seguenti versioni software e hardware:

- CPS 20.2
- Unified Computing System (UCS)-B
- MongoDB v3.6.17 e v3.4.16

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti. Se la rete è operativa, valutare attentamente eventuali conseguenze derivanti dall'uso dei comandi.

## Premesse

CPS utilizza MongoDB per costituire la struttura di base del database (DB). Possiede più set di repliche per vari scopi: ADMIN, Subscriber Profile Repository (SPR), BALANCE, SESSION, REPORTING e AUDIT.

Un set di repliche in MongoDB è un gruppo di processi mondani che gestiscono lo stesso set di dati. I set di repliche forniscono ridondanza e alta disponibilità (HA, High Availability). Con più copie di dati su server di database diversi, consente operazioni di lettura con condivisione del carico.

In alcune circostanze (ad esempio se si dispone di un database primario e di un database secondario ma i vincoli di costo impediscono l'aggiunta di un altro database secondario), è possibile scegliere di aggiungere un'istanza mondana a un set di repliche come arbitro per votare alle elezioni. Un arbitro ha esattamente 1 voto elettorale. Per impostazione predefinita, un arbitro ha priorità 0.

Gli arbitri sono istanze monotone che fanno parte di un set di repliche ma non contengono dati (il che significa che non forniscono ridondanza dei dati). Possono tuttavia partecipare alle elezioni. Un arbitro partecipa alle elezioni per il primario ma non dispone di una copia del set di dati e non può diventare un primario.

Gli arbitri hanno requisiti minimi in termini di risorse e non richiedono hardware dedicato. È possibile distribuire un arbitro in un server applicazioni o in un host che esegue solo il monitoraggio della rete.

Un arbitro non memorizza i dati, ma fino a quando il processo arbitro mondio non viene aggiunto al set di repliche, l'arbitro agisce come qualsiasi altro processo mondio e l'avvio con un set di file di dati e un giornale di registrazione di dimensioni intere.

Di seguito è riportato un esempio di set di repliche, ovvero **set07**.

```
| SET NAME - PORT : IP ADDRESS - REPLICA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY
|-----|
| SESSION:set07 |
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |
| Member-2 - 27727 : 192.168.10.146 - ARBITER - arbitervip - ON-LINE - ----- - 0 |
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |
|-----|
```

## Problema

Si supponga che esista un problema con un arbitro o un requisito per modificare l'arbitro in un set di repliche, è necessario rimuovere l'arbitro corrente e aggiungere un nuovo arbitro al set di repliche.

## Procedura per la gestione dell'arbitro in un set di repliche

Passaggio 1. Verificare la versione della shell mongo in CPS e il nuovo arbitro. Eseguire questo comando dal gestore sessioni primario nel set di repliche e nel nuovo nodo arbitro.

## Output di esempio di sessionmgr:

```
[root@sessionmgr02 ~]# mongo --version
MongoDB shell version v3.6.17
```

Se la versione della shell mongo è la stessa in sessionmgr primario e in new arbiter o se la nuova versione della shell mongo arbitro è superiore, passare al punto 6.

Altrimenti, se la nuova versione di arbiter mongo shell è inferiore, è necessario impostare **featureCompatibilityVersion** come il valore più basso nel database di amministrazione del set di repliche con i passaggi successivi.

Caso di esempio in cui la nuova versione della shell di arbitro mongo è inferiore a quella di CPS sessionmgr:

```
[root@pcrfclient02 ~]# mongo --version
MongoDB shell version v3.4.16
```

Passaggio 2. Accedere all'istanza mongo primaria del set di repliche.

Command template:

```
#mongo --host <sessionmgrXX> --port <Replica Set port>
```

Sample command:

```
#mongo --host sessionmgr02 --port 27727
```

Passaggio 3. Eseguire questo comando per visualizzare il **featureCompatibilityVersion** nel database admin del set di repliche.

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{
  "featureCompatibilityVersion" : {
    "version" : "3.6"
  },
  "ok" : 1,
  "operationTime" : Timestamp(1663914140, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1663914140, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
set07:PRIMARY>
```

Passaggio 4. Eseguire questo comando per **setfeatureCompatibilityVersion** come 3.4 nel database admin del set di repliche.

```
set07:PRIMARY> db.adminCommand( { setFeatureCompatibilityVersion: "3.4" } )
{ "ok" : 1 }
set07:PRIMARY>
```

Passaggio 5. Eseguire questo comando per verificare che **featureCompatibilityVersion** è stato modificato nella versione 3.4 nel database admin del set di repliche.

```
set07:PRIMARY> db.adminCommand( { getParameter: 1, featureCompatibilityVersion: 1 } )
{ "featureCompatibilityVersion" : { "version" : "3.4" }, "ok" : 1 }
set07:PRIMARY>
```

**Passaggio 6. Accedere a Gestione cluster e modificare `/var/qps/config/deploy/csv/AdditionalHosts.csv` con i nuovi dettagli dell'arbitro.**

```
#vi /var/qps/config/deploy/csv/AdditionalHosts.csv
```

Provide new arbiter details in this format:

Host Alias IP Address

```
new-arbiter new-arbiter xx.xx.xx.xx
```

**Passaggio 7. Importare la configurazione CSV.**

```
#!/var/qps/install/current/scripts/import/import_deploy.sh
```

**Passaggio 8. Verifica `/etc/hosts` che sono stati aggiornati con le informazioni dei nuovi arbitri.**

```
#cat /etc/hosts | grep arbiter
```

**Passaggio 9. Eseguire questo comando per la sincronizzazione `/etc/hosts`.**

```
#!/var/qps/bin/update/synchosts.sh
```

Syncing to following QNS Servers:

```
lb01 lb02 sessionmgr01 sessionmgr02 qns01 qns02 pcrfclient01 pcrfclient02
```

Do you want to Proceed? (y/n):y

```
lb01
```

```
lb02
```

```
sessionmgr01
```

```
sessionmgr02
```

```
qns01
```

```
qns02
```

```
pcrfclient01
```

```
pcrfclient02
```

**Passaggio 10. Verificare che gli script `mon_db` siano stati arrestati nelle VM client `pcrf`.**

```
#monsum | grep mon_db_for
```

**Se arrestato, questo è l'output:**

```
mon_db_for_lb_failover Not monitored Program
```

```
mon_db_for_callmodel Not monitored Program
```

**Se non viene interrotto, viene visualizzato l'output:**

```
mon_db_for_lb_failover OK Program
```

```
mon_db_for_callmodel OK Program
```

**Nota:** se gli script `mon_db` non vengono arrestati, eseguire questi comandi sulle rispettive VM client `pcffper` arrestarli manualmente.

```
#monit stop mon_db_for_lb_failover
```

```
#monit stop mon_db_for_callmodel
```

Passaggio 11. Eseguire questo comando da pcrfclient01 per rimuovere l'arbitro corrente dal set di repliche (set07 è un esempio in questo passaggio).

```
#build_set.sh --session --remove-members --setname set07
```

Please enter the member details which you going to remove from the replica-set

Member:Port -----> arbitervip:27727

arbitervip:27727

Do you really want to remove [yes(y)/no(n)]: y

Passaggio 12. Eseguire questo comando da Gestione cluster per verificare se l'arbitro è stato rimosso da set07, l'output di set07 non può contenere l'arbitro corrente.

```
#diagnostics.sh --get_replica_status
```

Expected output:

```
-----|
|-----|
| SESSION:set07 |
| Status via sessionmgr01:27727 sessionmgr02:27727 |
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec -|
| Member-2 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- -|
|-----|
-----|
```

Passaggio 13. Aggiornare il mongoConfig.cfg per disporre dell'arbitro appropriato nel set di repliche modificato. Sostituire l'arbitro corrente (ARBITER=arbitro) con il nuovo arbitro (ARBITER=nuovo-arbitro). Eseguire questo comando da Gestione cluster.

```
#vi /etc/broadhop/mongoConfig.cfg
```

Configurazione corrente:

```
[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=5120
ARBITER=arbitervip:27727
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:27727
MEMBER2=sessionmgr01:27727
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]
```

Configurazione richiesta:

```
[SESSION-SET2]
SETNAME=set07
OPLOG_SIZE=5120
ARBITER=new-arbitervip:27727
ARBITER_DATA_PATH=/var/data/sessions.7
MEMBER1=sessionmgr02:27727
MEMBER2=sessionmgr01:27727
DATA_PATH=/var/data/sessions.1/2
[SESSION-SET2-END]
```

Passaggio 14. Copia l'aggiornamento mongoConfig.cfg in tutte le VM. Eseguire questo comando da Gestione cluster.

```
#copytoall.sh /etc/broadhop/mongoConfig.cfg /etc/broadhop/mongoConfig.cfg
```

Passaggio 15. Aggiungere un nuovo membro arbitro a set07. Da Gestione cluster, eseguire `/var/qps/install/current/scripts/build/build_etc.sh` per compilare il file `/etc/directory`.

Passaggio 16. Verificare che il nuovo membro dell'arbitro venga aggiunto al set di repliche dopo l'esecuzione del `build_etc.sh`, è necessario attendere che il server AIDO crei/aggiorni i set di repliche con il nuovo arbitro.

```
#diagnostics.sh --get_replica_status
```

Expected Output:

```
| SET NAME - PORT : IP ADDRESS - REPLICA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY  
|-----|  
|  
| SESSION:set07 |  
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |  
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |  
| Member-2 - 27727 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - ----- - 0 |  
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |  
|-----|
```

**Nota:** se il nuovo arbitro non viene aggiunto, procedere con i passi successivi. In caso contrario, passare al passo 18.

Passaggio 17. Eseguire questo comando da Gestione cluster per aggiungere forzatamente un nuovo membro arbitro.

```
#build_set.sh --DB_NAME --add-members --setname Setxxx --force
```

Passaggio 18. Se la porta dell'arbitro non è ancora attiva, eseguire questo comando dal nuovo nodo dell'arbitro per avviarlo.

Command syntax:

```
#/etc/init.d/sessionmgr-XXXXX start
```

Sample command:

```
#/etc/init.d/sessionmgr-27727 start
```

Passaggio 19. Verificare che il nuovo arbitro sia stato aggiunto correttamente.

```
#diagnostics.sh --get_replica_status
```

Passaggio 20. Eseguire questo comando da Gestione cluster per aggiornare di conseguenza la priorità del database.

```
# cd /var/qps/bin/support/mongo/  
# ./set_priority.sh --db session  
# ./set_priority.sh --db spr  
# ./set_priority.sh --db admin  
# ./set_priority.sh --db balance  
# ./set_priority.sh --db audit  
# ./set_priority.sh --db report
```

Passaggio 21. Eseguire questo comando da Gestione cluster per verificare le modifiche nel set di

repliche.

```
#diagnostics.sh --get_replica_status
```

Expected Output:

```
| SET NAME - PORT : IP ADDRESS - REPLICA STATE - HOST NAME - HEALTH - LAST SYNC -PRIORITY  
|-----  
|  
| SESSION:set07 |  
| Status via arbitervip:27727 sessionmgr01:27727 sessionmgr02:27727 |  
| Member-1 - 27727 : - SECONDARY - sessionmgr01 - ON-LINE - 0 sec - 2 |  
| Member-2 - 27727 : xx.xx.xx.xx - ARBITER - new-arbiter - ON-LINE - ----- - 0 |  
| Member-3 - 27727 : - PRIMARY - sessionmgr02 - ON-LINE - ----- - 3 |  
|-----|
```

Passaggio 22. Verificare che gli script mon\_db vengano ripristinati nelle VM client pcrf. In caso contrario, è necessario avviarli manualmente.

```
#monsum | grep mon_db_for
```

Per abilitare lo script mon\_db, accedere a tutte le VM pcrfclient ed eseguire i seguenti comandi:

```
# monit start mon_db_for_lb_failover  
# monit start mon_db_for_callmodel
```

## Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).