Triggering Multi-Site Replication Failover and Switchover

Contents

The topic describes how to trigger a failover and switchover when using a Multi-Site Replication plan.

Overview

You can trigger a failover or switchover to redirect traffic to a secondary foundation. You can only trigger a failover or switchover in MySQL for v2.7.3 and later.

For information about when to trigger a failover or switchover, see About Failover and Switchover.

Before you trigger a failover or switchover, you must verify that the follower service instance is healthy. See Verify Follower Health below.

> **Note**
>
> The procedures in this topic assume that you created the leader service instance in the primary foundation and the follower service instance in the secondary foundation.

Verify Follower Health

Before you trigger a failover or switchover, you must verify that the follower service instance is healthy. If your follower service instance is unhealthy, contact Support.

To verify the service instance:

1. Log in to the deployment for your secondary foundation by running:

   ```
   cf login SECONDARY-API-URL
   ```

   Where `SECONDARY-API-URL` is the API endpoint for your secondary foundation.
2. Record the GUID of the follower service instance by running:

   ```
   cf service SERVICE-INSTANCE-NAME --guid
   ```

   Where `SERVICE-INSTANCE-NAME` is the name of the follower service instance. For example:

   ```
   $ cf service my-follower --guid
     12345678-90ab-cdef-1234-567890abcdef
   ```

3. Obtain the credentials and IP address needed to SSH into the Ops Manager VM by following the procedure in Gather Credential and IP Address Information.
4. SSH into the Ops Manager VM by following the procedure in Log in to the Ops Manager VM with SSH.
5. From the Ops Manager VM, log in to your BOSH Director by following the procedure in Authenticate with the BOSH Director VM.
6. View the health of the service instance by running:

   ```
   bosh -d service-instance_GUID instance
   ```

   For example:

   ```
   bosh -d service-instance_12345678-90ab-cdef-1234-567890abcdef instance
       Using environment 'https<span>:</span>//10.0.0.6:25555' as client
   'admin'

       Task 21409. Done

       Deployment 'service-instance_12345678-90ab-cdef-1234-567890abcdef'

       Instance                                     Process State  AZ  IPs
       mysql/1373022d-4eab-46d3-8fd1-a12067edf597  running        z2
   10.0.17.14

       1 instances

       Succeeded
   ```

7. Ensure that the service instance is `running`. If the service instance is `failing`, contact Support.

Trigger a Failover

> **Warning**
>
> You should only trigger failover if you do not need to recover the leader service instance. You cannot recover the downed leader service instance or re-establish multi-site replication with the new leader service instance.

To trigger a failover:

1. Promote the Follower
2. Delete the Former Leader

Promote the Follower

> **Note**
>
> If you try to promote a leader-follower, highly available cluster, or single node service instance to leader or make it read-only you get an error message similar to the following:
>
> ```
> Updating service instance haDB as admin...
>   FAILED
>   Server error, status code: 502, error code: 10001, message: Service
> broker error: 1 error occurred:
>       * the configuration parameter 'initiate-failover' is not a valid
> option
> ```

To promote the follower service instance to leader:

1. Log in to the deployment for your secondary foundation by running:

```
cf login SECONDARY-API-URL
```

Where `SECONDARY-API-URL` is the API endpoint for your secondary foundation.

2. Promote the follower service instance to leader by running:

```
cf update-service SECONDARY-INSTANCE \
-c '{"initiate-failover":"promote-follower-to-leader"}'
```

For example:

```
$ cf update-service secondary-node \
    -c '{"initiate-failover":"promote-follower-to-leader"}'
Updating service instance secondary-node as admin...
```

3. If the above command fails, do one of the following:
   - If you have local transactions that are not applied on the follower service instance, wait for the transactions to be applied and then run the above command again. The error message looks like the following:

```
Updating service instance secondary-node as admin...
FAILED
Server error, status code: 502, error code: 10001, message: Service
broker error: Promotion of follower failed - has 1 transactions
still unapplied
```

   - If the leader service instance is still reachable and in read-write mode, follow the procedure in Trigger a Switchover below instead. The error message looks like the following:

```
Updating service instance secondary-node as admin...
FAILED
Server error, status code: 502, error code: 10001, message: Service
broker error: Promotion of follower failed - the leader is still
writable
```

4. Watch the progress of the service instance update by running:

```
watch cf services
```

Wait for the `last operation` for your instance to show as `update succeeded`. For example:

```
$ watch cf services

Getting services in org my-org / space my-space as admin...
OK
name             service       plan                      bound apps
last operation
secondary-node   p.mysql       db-pxc-single-node-small
update succeeded
```

5. Reconfigure your global DNS load balancer to direct all traffic to apps in your secondary foundation. See Configure Your GLB.

Delete the Former Leader

When you do a failover, the leader service instance cannot be manually recovered. After you promote the follower service instance to leader, you should delete the former leader service instance. Otherwise, the service instance could recover in read-write mode.

To delete the former leader service instance:

1. Log in to the deployment for your primary foundation by running:

```
cf login PRIMARY-API-URL
```

Where `PRIMARY-API-URL` is the API endpoint for the primary foundation.

2. Remove all bindings and service keys from the former leader service instance by doing the procedure in [Unbind an App from a Service Instance](#).
3. Delete the former leader service instance by doing the procedure in [Delete a Service Instance](#).

Trigger a Switchover

To trigger a switchover:

1. [Promote the Follower](#)
2. [Reconfigure Multi-Site Replication](#)

Promote the Follower

Before you promote the follower service instance, you must make the leader service instance, which is in the primary foundation, read-only.

To make the leader read-only and promote the follower to leader in the secondary foundation:

1. Log in to the deployment for your primary foundation by running:

   ```
   cf login PRIMARY-API-URL
   ```

   Where `PRIMARY-API-URL` is the API endpoint for the primary foundation.
2. Set the service instance that is currently the leader to read-only by running:

   ```
   cf update-service PRIMARY-INSTANCE \
     -c '{"initiate-failover":"make-leader-read-only"}
   ```

   For example:

   ```
   $ cf update-service primary-node \
       -c '{"initiate-failover":"make-leader-read-only"}'

   Updating service instance primary-node as admin...
   OK
   ```

3. Watch the progress of the service instance update by running:

   ```
   watch cf services
   ```

   Wait for the `last operation` for your instance to show as `update succeeded`.
4. Log in to the deployment for your secondary foundation by running:

   ```
   cf login SECONDARY-API-URL
   ```

   Where `SECONDARY-API-URL` is the API endpoint for your secondary foundation.
5. Promote the service instance in the secondary foundation to leader by running:

   ```
   cf update-service SECONDARY-INSTANCE \
     -c '{"initiate-failover":"promote-follower-to-leader"}'
   ```

6. Watch the progress of the service instance update by running:

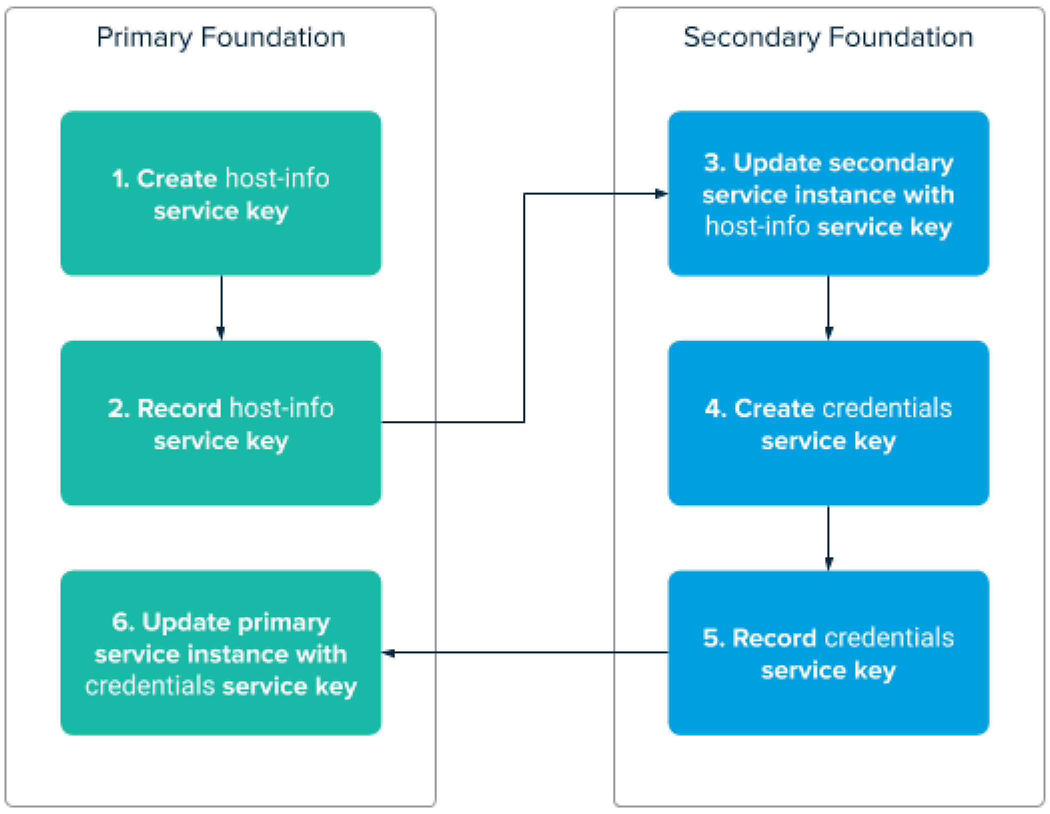
   ```
   watch cf services
   ```

   Wait for the `last operation` for your instance to show as `update succeeded`.

Reconfigure Multi-Site Replication

To establish a connection between the service instances in the primary and secondary foundations, you must reconfigure replication. Re-configuring replication is similar to the procedure in Configure Multi-Site Replication except that the service instance in the primary foundation is the follower and the service instance in the secondary foundation is the leader.

The following diagram describes the workflow for re-configuring multi-site replication:


Primary Foundation
Secondary Foundation
1. Create host-info service key
2. Record host-info service key
3. Update secondary service instance with host-info service key
4. Create credentials service key
5. Record credentials service key
6. Update primary service instance with credentials service key

To reconfigure replication for the service instances:

1. Log in to the deployment for your primary foundation by running:

```
cf login PRIMARY-API-URL
```

2. Create a host-info service key for the service instance in your primary foundation:

```
cf create-service-key PRIMARY-INSTANCE SERVICE-KEY \
  -c '{"replication-request": "host-info"}
```

   Where:
   - `PRIMARY-INSTANCE` is the name of the follower service instance in the primary foundation.
   - `SERVICE-KEY` is a name you choose for the host-info service key.

   For example:

```
$ cf create-service-key primary-node host-info \
      -c '{"replication-request": "host-info" }'
Creating service key host-info for service instance primary-node as
admin...
OK
```

3. View the `replication-credentials` for your host-info service key by running:

```
cf service-key PRIMARY-INSTANCE SERVICE-KEY
```

   Where:
   - `PRIMARY-INSTANCE` is the name of the follower service instance in the primary foundation.
   - `SERVICE-KEY` is the name of the host-info service key you created in the step above.

   For example:

```
$ cf service-key primary-node host-info-key

Getting key host-info-key for service instance primary-node as admin...

  {
    "replication": {
        "peer-info": {
          "hostname": "primary.bosh",
          "ip": "10.0.19.12",
          "system_domain": "sys.primary-domain.com",
          "uuid": "ab12cd34-5678-91e2-345f-67891h234567"
      },
      "role": "leader"
    }
  }
```

4. Record the output of the above command.
5. Log in to the deployment for your secondary foundation by running:

   ```
   cf login SECONDARY-API-URL
   ```

6. Update your leader service instance in the secondary foundation with the host-info service key by running:

   ```
   cf update-service SECONDARY-INSTANCE -c HOST-INFO
   ```

   Where:
   - `SECONDARY-INSTANCE` is the name of the leader service instance in the secondary foundation.
   - `HOST-INFO` is the output you recorded in the step above.

   For example:

   ```
   $ cf update-service secondary-node -c {"replication":{
     "peer-info":{
         "hostname": "primary.bosh",
         "ip": "10.0.18.12",
         "system_domain": "sys.primary-domain.com",
         "uuid": "ab12cd34-5678-91e2-345f-67891h234567"
       },
     "role": "leader"
     }
   }

   Updating service instance secondary-node as admin...

   OK
   ```

7. Watch the progress of the service instance update by running:

   ```
   watch cf services
   ```

   Wait for the `last operation` for your instance to show as `update succeeded`.
8. Create a credentials service key for the service instance in your secondary foundation by running:

   ```
   cf create-service-key SECONDARY-INSTANCE SERVICE-KEY-NAME \
     -c '{"replication-request": "credentials"}
   ```

   Where:
   - `SECONDARY-INSTANCE` is the name of the service instance in the secondary foundation.
   - `SERVICE-KEY-NAME` is a name you choose for the credentials service key.

   For example:

```
$ cf create-service-key secondary-node cred-key \
    -c '{"replication-request": "credentials" }'

Creating service key cred-key for service instance secondary-node as
user<span>@</span>example.com...
OK
```

> **Note**
>
> The `-c` flag is different than the one in the step above.

9. View the `replication-credentials` for your credentials service key by running:

   ```
   cf service-key SECONDARY-INSTANCE SERVICE-KEY-NAME
   ```

   Where:
   - `SECONDARY-INSTANCE` is the name of the leader service instance in the secondary foundation.
   - `SERVICE-KEY-NAME` is the name of the credentials service key you created in the step above

   For example:

   ```
   $ cf service-key secondary-node cred-key

   Getting key cred-key for service instance secondary as admin...

     {
       "replication": {
         "credentials": {
           "password": "a22aaa2a2a2aaaaa",
           "username": "6bf07ae455a14064a9073cec8696366c"
         },
         "peer-info": {
           "hostname": "secondary.bosh",
           "ip": "10.0.17.12",
           "system_domain": "sys.secondary-domain.com",
           "uuid": "zy98xw76-5432-19v8-765u-43219t876543"
         },
         "role": "follower"
       }
     }
   ```

10. Record the output of the above command.
11. Log in to the deployment for your primary foundation by running:

    ```
    cf login PRIMARY-API-URL
    ```

12. Update the follower service instance in the primary foundation with the credentials service key by running:

    ```
    cf update-service PRIMARY-INSTANCE -c CREDENTIALS
    ```

    Where:
    - `PRIMARY-INSTANCE` is name of the follower service instance in the primary foundation.
    - `CREDENTIALS` is the output you recorded in the step above.

    For example:

```
$ cf update-service primary-node -c {"replication": {
    "credentials": {
      "password": "a22aaa2a2a2aaaaa",
      "username": "6bf07ae455a14064a9073cec8696366c"
    },
    "peer-info": {
      "hostname": "secondary.bosh",
      "ip": "10.0.17.12",
      "system_domain": "sys.secondary-domain.com",
      "uuid": "zy98xw76-5432-19v8-765u-43219t876543"
    },
    "role": "follower"
  }
}

Updating service instance primary-node as admin...
OK
```

13. Watch the progress of the service instance update by running:

    ```
    watch cf services
    ```

    Wait for the `last operation` for your instance to show as `update succeeded`.

    You should now have a leader-follower service instance successfully configured, where the leader is in your secondary foundation and your follower is in the primary foundation.

    If the `cf update-service` command fails, you must create a new multi-site replication service instance and reconfigure replication using this new, empty instance as the follower.
14. Reconfigure your global DNS load balancer to direct traffic to the correct foundations of your choice. See Configure Your GLB.