Improve Query Condition of FederationStateStore#getApplicationsHome SubCluster Test Report

Improve Query Condition of FederationStateStore#getApplicationsHomeSubCluster Test Report
1.SqlServer
- 1.1 SQL Server 2008 R2 Enterprise
  - 1.1.1 Table
  - 1.1.2 Stored Procedure
- 1.2 SQL Server 2012 Enterprise
  - 1.2.1 Table
  - 1.2.2 Stored Procedure
- 1.3 SQL Server 2016 Enterprise
  - 1.3.1 Table
  - 1.3.2 Stored Procedure
- 1.4 SQL Server 2017 Enterprise
  - 1.4.1 Table
  - 1.4.2 Stored Procedure
- 1.5 SQL Server 2019 Enterprise
  - 1.5.1 Table
  - 1.5.2 Stored Procedure

2.MySQL
- 2.1 Mysql5.7
  - 2.1.1 Table
  - 2.1.2 Stored Procedure
- 2.2 Mysql5.8
  - 2.2.1 Table
  - 2.2.2 Stored Procedure

1.SqlServer

1.1 SQL Server 2008 R2 Enterprise

1.1.1 Table

Verify that the table creation script (FederationStateStoreTables.sql) can be executed normally

```
SELECT @@Version
> OK
> Time: 0.015s

Microsoft SQL Server 2008 R2 (SP3) - 10.50.6000.34 (X64)
```

```
Aug 19 2014 12:21:34
Copyright (c) Microsoft Corporation
Enterprise Edition (64-bit) on Windows NT 6.3 <X64> (Build 14393: )

USE [FederationStateStore]
> OK
> Time: 0.010s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'applicationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table applicationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[applicationsHomeSubCluster](
            applicationId   VARCHAR(64) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createAppTime DEFAULT
GETUTCDATE(),

            CONSTRAINT [pk_applicationId] PRIMARY KEY
            (
                [applicationId]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table applicationsHomeSubCluster created.'
    END
ELSE
    PRINT 'Table applicationsHomeSubCluster exists, no operation required...'
> OK
> Time: 0.015s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'membership'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table membership does not exist, create it...'
```

```
        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[membership](
            [subClusterId]            VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT
NULL,
            [amRMServiceAddress]      VARCHAR(256) NOT NULL,
            [clientRMServiceAddress]  VARCHAR(256) NOT NULL,
            [rmAdminServiceAddress]   VARCHAR(256) NOT NULL,
            [rmWebServiceAddress]     VARCHAR(256) NOT NULL,
            [lastHeartBeat]           DATETIME2 NOT NULL,
            [state]                   VARCHAR(32) NOT NULL,
            [lastStartTime]           BIGINT NOT NULL,
            [capability]              VARCHAR(6000) NOT NULL,

            CONSTRAINT [pk_subClusterId] PRIMARY KEY
            (
                [subClusterId]
            ),
            CONSTRAINT [uc_lastStartTime] UNIQUE
            (
                [lastStartTime]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table membership created.'
    END
ELSE
    PRINT 'Table membership exists, no operation required...'
> OK
> Time: 0.017s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'policies'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table policies does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON
```

```
        CREATE TABLE [dbo].[policies](
            queue       VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT NULL,
            policyType  VARCHAR(256) NOT NULL,
            params      VARBINARY(6000) NOT NULL,

            CONSTRAINT [pk_queue] PRIMARY KEY
            (
                [queue]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table policies created.'
    END
ELSE
    PRINT 'Table policies exists, no operation required...'
> OK
> Time: 0.021s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'reservationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table reservationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[reservationsHomeSubCluster](
            reservationId   VARCHAR(128) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createResTime DEFAULT
GETUTCDATE(),

            CONSTRAINT [pk_reservationId] PRIMARY KEY
            (
                [reservationId]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table reservationsHomeSubCluster created.'
```

```
    END
ELSE
    PRINT 'Table reservationsHomeSubCluster exists, no operation required...'
> OK
> Time: 0.016s
```

1.1.2 Stored Procedure

Verify that the stored procedure script (FederationStateStoreStoredProcs.sql) can be executed normally.

```
USE [FederationStateStore]
> OK
> Time: 0.016s


IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
> OK
> Time: 0.009s


CREATE PROCEDURE [dbo].[sp_addApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                       FROM [dbo].[applicationsHomeSubCluster]
                       WHERE [applicationId] = @applicationId)

                INSERT INTO [dbo].[applicationsHomeSubCluster] (
                    [applicationId],
                    [homeSubCluster])
                VALUES (
                    @applicationId,
                    @homeSubCluster);
            -- End of the IF block

            SELECT @rowCount = @@ROWCOUNT;
```

```
            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.011s


IF OBJECT_ID ( '[sp_updateApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateApplicationHomeSubCluster];
> OK
> Time: 0.008s


CREATE PROCEDURE [dbo].[sp_updateApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[applicationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
            WHERE [applicationId] = @applicationid;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN
```

```
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.009s


IF OBJECT_ID ( '[sp_getApplicationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationsHomeSubCluster];
> OK
> Time: 0.009s


CREATE PROCEDURE [dbo].[sp_getApplicationsHomeSubCluster]
    @limit int,
    @homeSubCluster VARCHAR(256)
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT
            [applicationId],
            [homeSubCluster],
            [createTime]
        FROM
        (SELECT
            [applicationId],
            [homeSubCluster],
            [createTime],
            row_number() over(partition by [homeSubCluster] order by [createTime] desc)
AS row_num
        FROM [dbo].[applicationsHomeSubCluster]) AS t
        WHERE row_num <= @limit
          AND (CASE WHEN @homeSubCluster IS NULL THEN 1
                    WHEN @homeSubCluster IS NOT NULL AND [homeSubCluster] =
@homeSubCluster THEN 1
                ELSE 0 END) = 1
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())
```

```
        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.010s


IF OBJECT_ID ( '[sp_getApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationHomeSubCluster];
> OK
> Time: 0.009s


CREATE PROCEDURE [dbo].[sp_getApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[applicationsHomeSubCluster]
        WHERE [applicationId] = @applicationid;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.008s


IF OBJECT_ID ( '[sp_deleteApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteApplicationHomeSubCluster];
> OK
```

```
> Time: 0.008s


CREATE PROCEDURE [dbo].[sp_deleteApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.009s


IF OBJECT_ID ( '[sp_registerSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_registerSubCluster];
> OK
> Time: 0.012s


CREATE PROCEDURE [dbo].[sp_registerSubCluster]
    @subClusterId VARCHAR(256),
    @amRMServiceAddress VARCHAR(256),
    @clientRMServiceAddress VARCHAR(256),
    @rmAdminServiceAddress VARCHAR(256),
    @rmWebServiceAddress VARCHAR(256),
    @state VARCHAR(32),
    @lastStartTime BIGINT,
```

```
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId;
            INSERT INTO [dbo].[membership] (
                [subClusterId],
                [amRMServiceAddress],
                [clientRMServiceAddress],
                [rmAdminServiceAddress],
                [rmWebServiceAddress],
                [lastHeartBeat],
                [state],
                [lastStartTime],
                [capability] )
            VALUES (
                @subClusterId,
                @amRMServiceAddress,
                @clientRMServiceAddress,
                @rmAdminServiceAddress,
                @rmWebServiceAddress,
                GETUTCDATE(),
                @state,
                @lastStartTime,
                @capability);
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.009s
```

```
IF OBJECT_ID ( '[sp_getSubClusters]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubClusters];
> OK
> Time: 0.011s


CREATE PROCEDURE [dbo].[sp_getSubClusters]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [subClusterId], [amRMServiceAddress], [clientRMServiceAddress],
               [rmAdminServiceAddress], [rmWebServiceAddress], [lastHeartBeat],
               [state], [lastStartTime], [capability]
        FROM [dbo].[membership]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.009s


IF OBJECT_ID ( '[sp_getSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubCluster];
> OK
> Time: 0.011s


CREATE PROCEDURE [dbo].[sp_getSubCluster]
    @subClusterId VARCHAR(256),
    @amRMServiceAddress VARCHAR(256) OUTPUT,
    @clientRMServiceAddress VARCHAR(256) OUTPUT,
    @rmAdminServiceAddress VARCHAR(256) OUTPUT,
    @rmWebServiceAddress VARCHAR(256) OUTPUT,
    @lastHeartbeat DATETIME2 OUTPUT,
    @state VARCHAR(256) OUTPUT,
    @lastStartTime BIGINT OUTPUT,
    @capability VARCHAR(6000) OUTPUT
```

```
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            SELECT @subClusterId = [subClusterId],
                   @amRMServiceAddress = [amRMServiceAddress],
                   @clientRMServiceAddress = [clientRMServiceAddress],
                   @rmAdminServiceAddress = [rmAdminServiceAddress],
                   @rmWebServiceAddress = [rmWebServiceAddress],
                   @lastHeartBeat = [lastHeartBeat],
                   @state = [state],
                   @lastStartTime = [lastStartTime],
                   @capability = [capability]
            FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.010s


IF OBJECT_ID ( '[sp_subClusterHeartbeat]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_subClusterHeartbeat];
> OK
> Time: 0.012s


CREATE PROCEDURE [dbo].[sp_subClusterHeartbeat]
    @subClusterId VARCHAR(256),
    @state VARCHAR(256),
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
```

```
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[membership]
            SET [state] = @state,
                [lastHeartbeat] = GETUTCDATE(),
                [capability] = @capability
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.009s


IF OBJECT_ID ( '[sp_deregisterSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deregisterSubCluster];
> OK
> Time: 0.009s


CREATE PROCEDURE [dbo].[sp_deregisterSubCluster]
    @subClusterId VARCHAR(256),
    @state VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[membership]
            SET [state] = @state
```

```
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.013s


IF OBJECT_ID ( '[sp_setPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_setPolicyConfiguration];
> OK
> Time: 0.009s


CREATE PROCEDURE [dbo].[sp_setPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256),
    @params VARBINARY(512),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[policies]
            WHERE [queue] = @queue;
            INSERT INTO [dbo].[policies] (
                [queue],
                [policyType],
                [params])
            VALUES (
                @queue,
                @policyType,
                @params);
```

```
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.010s


IF OBJECT_ID ( '[sp_getPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPolicyConfiguration];
> OK
> Time: 0.009s


CREATE PROCEDURE [dbo].[sp_getPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256) OUTPUT,
    @params VARBINARY(6000) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @policyType = [policyType],
               @params = [params]
        FROM [dbo].[policies]
        WHERE [queue] = @queue

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
```

```
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.008s


IF OBJECT_ID ( '[sp_getPoliciesConfigurations]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPoliciesConfigurations];
> OK
> Time: 0.009s


CREATE PROCEDURE [dbo].[sp_getPoliciesConfigurations]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT  [queue], [policyType], [params] FROM [dbo].[policies]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.009s


IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
> OK
> Time: 0.009s


CREATE PROCEDURE [dbo].[sp_addReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
```

```
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                      FROM [dbo].[reservationsHomeSubCluster]
                      WHERE [reservationId] = @reservationId)

                INSERT INTO [dbo].[reservationsHomeSubCluster] (
                    [reservationId],
                    [homeSubCluster])
                VALUES (
                    @reservationId,
                    @homeSubCluster);
            -- End of the IF block

            SELECT @rowCount = @@ROWCOUNT;

            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.011s


IF OBJECT_ID ( '[sp_updateReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateReservationHomeSubCluster];
> OK
> Time: 0.008s
```

```
CREATE PROCEDURE [dbo].[sp_updateReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[reservationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.009s


IF OBJECT_ID ( '[sp_getReservationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationsHomeSubCluster];
> OK
> Time: 0.013s


CREATE PROCEDURE [dbo].[sp_getReservationsHomeSubCluster]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [reservationId], [homeSubCluster], [createTime]
        FROM [dbo].[reservationsHomeSubCluster]
    END TRY
```

```
    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.010s


IF OBJECT_ID ( '[sp_getReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationHomeSubCluster];
> OK
> Time: 0.008s


CREATE PROCEDURE [dbo].[sp_getReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[reservationsHomeSubCluster]
        WHERE [reservationId] = @reservationId;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.011s
```

```
IF OBJECT_ID ( '[sp_deleteReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteReservationHomeSubCluster];
> OK
> Time: 0.009s


CREATE PROCEDURE [dbo].[sp_deleteReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.009s
```

1.2 SQL Server 2012 Enterprise

1.2.1 Table

Verify that the table creation script (FederationStateStoreTables.sql) can be executed normally

```
SELECT @@Version
> OK
> Time: 0.015s
```

```
Microsoft SQL Server 2012 (SP3) (KB3072779) - 11.0.6020.0 (X64)
Oct 20 2015 15:36:27
Copyright (c) Microsoft Corporation
Enterprise Edition (64-bit) on Windows NT 6.3 <X64> (Build 9600: )

USE [FederationStateStore]
> OK
> Time: 0.041s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'applicationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table applicationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[applicationsHomeSubCluster](
            applicationId   VARCHAR(64) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createAppTime DEFAULT
GETUTCDATE(),

            CONSTRAINT [pk_applicationId] PRIMARY KEY
            (
                [applicationId]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table applicationsHomeSubCluster created.'
    END
ELSE
    PRINT 'Table applicationsHomeSubCluster exists, no operation required...'
> OK
> Time: 0.050s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'membership'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table membership does not exist, create it...'
```

```
        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[membership](
            [subClusterId]            VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT
NULL,
            [amRMServiceAddress]      VARCHAR(256) NOT NULL,
            [clientRMServiceAddress]  VARCHAR(256) NOT NULL,
            [rmAdminServiceAddress]   VARCHAR(256) NOT NULL,
            [rmWebServiceAddress]     VARCHAR(256) NOT NULL,
            [lastHeartBeat]           DATETIME2 NOT NULL,
            [state]                   VARCHAR(32) NOT NULL,
            [lastStartTime]           BIGINT NOT NULL,
            [capability]              VARCHAR(6000) NOT NULL,

            CONSTRAINT [pk_subClusterId] PRIMARY KEY
            (
                [subClusterId]
            ),
            CONSTRAINT [uc_lastStartTime] UNIQUE
            (
                [lastStartTime]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table membership created.'
    END
ELSE
    PRINT 'Table membership exists, no operation required...'
> OK
> Time: 0.051s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'policies'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table policies does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON
```

```
        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[policies](
            queue       VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT NULL,
            policyType  VARCHAR(256) NOT NULL,
            params      VARBINARY(6000) NOT NULL,

            CONSTRAINT [pk_queue] PRIMARY KEY
            (
                [queue]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table policies created.'
    END
ELSE
    PRINT 'Table policies exists, no operation required...'
> OK
> Time: 0.046s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'reservationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table reservationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[reservationsHomeSubCluster](
            reservationId   VARCHAR(128) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createResTime DEFAULT
GETUTCDATE(),

            CONSTRAINT [pk_reservationId] PRIMARY KEY
            (
                [reservationId]
            )
        )

        SET ANSI_PADDING OFF
```

```
        PRINT 'Table reservationsHomeSubCluster created.'
    END
ELSE
    PRINT 'Table reservationsHomeSubCluster exists, no operation required...'
> OK
> Time: 0.048s
```

1.2.2 Stored Procedure

Verify that the stored procedure script (FederationStateStoreStoredProcs.sql) can be executed normally.

```
USE [FederationStateStore]
> OK
> Time: 0.045s


IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_addApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                       FROM [dbo].[applicationsHomeSubCluster]
                       WHERE [applicationId] = @applicationId)

                INSERT INTO [dbo].[applicationsHomeSubCluster] (
                    [applicationId],
                    [homeSubCluster])
                VALUES (
                    @applicationId,
                    @homeSubCluster);
            -- End of the IF block
```

```
            SELECT @rowCount = @@ROWCOUNT;

            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.040s


IF OBJECT_ID ( '[sp_updateApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateApplicationHomeSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_updateApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[applicationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
            WHERE [applicationId] = @applicationid;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY
```

```
    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.040s


IF OBJECT_ID ( '[sp_getApplicationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationsHomeSubCluster];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_getApplicationsHomeSubCluster]
    @limit int,
    @homeSubCluster VARCHAR(256)
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT
            [applicationId],
            [homeSubCluster],
            [createTime]
        FROM
        (SELECT
            [applicationId],
            [homeSubCluster],
            [createTime],
            row_number() over(partition by [homeSubCluster] order by [createTime] desc)
AS row_num
        FROM [dbo].[applicationsHomeSubCluster]) AS t
        WHERE row_num <= @limit
          AND (CASE WHEN @homeSubCluster IS NULL THEN 1
                    WHEN @homeSubCluster IS NOT NULL AND [homeSubCluster] =
@homeSubCluster THEN 1
               ELSE 0 END) = 1
    END TRY
```

```
    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.055s


IF OBJECT_ID ( '[sp_getApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationHomeSubCluster];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_getApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[applicationsHomeSubCluster]
        WHERE [applicationId] = @applicationid;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.040s
```

```
IF OBJECT_ID ( '[sp_deleteApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteApplicationHomeSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_deleteApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.045s


IF OBJECT_ID ( '[sp_registerSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_registerSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_registerSubCluster]
    @subClusterId VARCHAR(256),
    @amRMServiceAddress VARCHAR(256),
    @clientRMServiceAddress VARCHAR(256),
    @rmAdminServiceAddress VARCHAR(256),
```

```
    @rmWebServiceAddress VARCHAR(256),
    @state VARCHAR(32),
    @lastStartTime BIGINT,
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId;
            INSERT INTO [dbo].[membership] (
                [subClusterId],
                [amRMServiceAddress],
                [clientRMServiceAddress],
                [rmAdminServiceAddress],
                [rmWebServiceAddress],
                [lastHeartBeat],
                [state],
                [lastStartTime],
                [capability] )
            VALUES (
                @subClusterId,
                @amRMServiceAddress,
                @clientRMServiceAddress,
                @rmAdminServiceAddress,
                @rmWebServiceAddress,
                GETUTCDATE(),
                @state,
                @lastStartTime,
                @capability);
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
```

```
END;
> Affected rows: 0
> Time: 0.040s


IF OBJECT_ID ( '[sp_getSubClusters]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubClusters];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_getSubClusters]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [subClusterId], [amRMServiceAddress], [clientRMServiceAddress],
               [rmAdminServiceAddress], [rmWebServiceAddress], [lastHeartBeat],
               [state], [lastStartTime], [capability]
        FROM [dbo].[membership]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.047s


IF OBJECT_ID ( '[sp_getSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubCluster];
> OK
> Time: 0.040s


CREATE PROCEDURE [dbo].[sp_getSubCluster]
    @subClusterId VARCHAR(256),
    @amRMServiceAddress VARCHAR(256) OUTPUT,
    @clientRMServiceAddress VARCHAR(256) OUTPUT,
    @rmAdminServiceAddress VARCHAR(256) OUTPUT,
    @rmWebServiceAddress VARCHAR(256) OUTPUT,
    @lastHeartbeat DATETIME2 OUTPUT,
```

```
    @state VARCHAR(256) OUTPUT,
    @lastStartTime BIGINT OUTPUT,
    @capability VARCHAR(6000) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            SELECT @subClusterId = [subClusterId],
                   @amRMServiceAddress = [amRMServiceAddress],
                   @clientRMServiceAddress = [clientRMServiceAddress],
                   @rmAdminServiceAddress = [rmAdminServiceAddress],
                   @rmWebServiceAddress = [rmWebServiceAddress],
                   @lastHeartBeat = [lastHeartBeat],
                   @state = [state],
                   @lastStartTime = [lastStartTime],
                   @capability = [capability]
            FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.041s


IF OBJECT_ID ( '[sp_subClusterHeartbeat]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_subClusterHeartbeat];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_subClusterHeartbeat]
    @subClusterId VARCHAR(256),
    @state VARCHAR(256),
```

```
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[membership]
            SET [state] = @state,
                [lastHeartbeat] = GETUTCDATE(),
                [capability] = @capability
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s


IF OBJECT_ID ( '[sp_deregisterSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deregisterSubCluster];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_deregisterSubCluster]
    @subClusterId VARCHAR(256),
    @state VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
```

```
            UPDATE [dbo].[membership]
            SET [state] = @state
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.045s


IF OBJECT_ID ( '[sp_setPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_setPolicyConfiguration];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_setPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256),
    @params VARBINARY(512),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[policies]
            WHERE [queue] = @queue;
            INSERT INTO [dbo].[policies] (
                [queue],
                [policyType],
                [params])
            VALUES (
```

```
                @queue,
                @policyType,
                @params);
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.051s


IF OBJECT_ID ( '[sp_getPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPolicyConfiguration];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_getPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256) OUTPUT,
    @params VARBINARY(6000) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @policyType = [policyType],
               @params = [params]
        FROM [dbo].[policies]
        WHERE [queue] = @queue

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())
```

```
        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.041s


IF OBJECT_ID ( '[sp_getPoliciesConfigurations]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPoliciesConfigurations];
> OK
> Time: 0.043s


CREATE PROCEDURE [dbo].[sp_getPoliciesConfigurations]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT  [queue], [policyType], [params] FROM [dbo].[policies]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.054s


IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
> OK
> Time: 0.041s


CREATE PROCEDURE [dbo].[sp_addReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
```

```
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                       FROM [dbo].[reservationsHomeSubCluster]
                       WHERE [reservationId] = @reservationId)

                INSERT INTO [dbo].[reservationsHomeSubCluster] (
                    [reservationId],
                    [homeSubCluster])
                VALUES (
                    @reservationId,
                    @homeSubCluster);
            -- End of the IF block

            SELECT @rowCount = @@ROWCOUNT;

            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.041s


IF OBJECT_ID ( '[sp_updateReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateReservationHomeSubCluster];
> OK
```

```
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_updateReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[reservationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s


IF OBJECT_ID ( '[sp_getReservationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationsHomeSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_getReservationsHomeSubCluster]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [reservationId], [homeSubCluster], [createTime]
```

```
        FROM [dbo].[reservationsHomeSubCluster]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.041s


IF OBJECT_ID ( '[sp_getReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationHomeSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_getReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[reservationsHomeSubCluster]
        WHERE [reservationId] = @reservationId;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
```

```
> Time: 0.039s


IF OBJECT_ID ( '[sp_deleteReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteReservationHomeSubCluster];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_deleteReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.040s
```

1.3 SQL Server 2016 Enterprise

1.3.1 Table

Verify that the table creation script (FederationStateStoreTables.sql) can be executed normally

```
SELECT @@Version
> OK
> Time: 0.015s

Microsoft SQL Server 2016 (SP2) (KB4052908) - 13.0.5026.0 (X64)
Mar 18 2018 09:11:49
Copyright (c) Microsoft Corporation
Enterprise Edition: Core-based Licensing (64-bit) on Windows Server 2016 Datacenter
10.0 <X64> (Build 14393: )

USE [FederationStateStore]
> OK
> Time: 0.047s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'applicationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table applicationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[applicationsHomeSubCluster](
            applicationId   VARCHAR(64) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createAppTime DEFAULT
GETUTCDATE(),

            CONSTRAINT [pk_applicationId] PRIMARY KEY
            (
                [applicationId]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table applicationsHomeSubCluster created.'
    END
ELSE
    PRINT 'Table applicationsHomeSubCluster exists, no operation required...'
```

```
> OK
> Time: 0.050s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'membership'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table membership does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[membership](
            [subClusterId]            VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT
NULL,
            [amRMServiceAddress]      VARCHAR(256) NOT NULL,
            [clientRMServiceAddress]  VARCHAR(256) NOT NULL,
            [rmAdminServiceAddress]   VARCHAR(256) NOT NULL,
            [rmWebServiceAddress]     VARCHAR(256) NOT NULL,
            [lastHeartBeat]           DATETIME2 NOT NULL,
            [state]                   VARCHAR(32) NOT NULL,
            [lastStartTime]           BIGINT NOT NULL,
            [capability]              VARCHAR(6000) NOT NULL,

            CONSTRAINT [pk_subClusterId] PRIMARY KEY
            (
                [subClusterId]
            ),
            CONSTRAINT [uc_lastStartTime] UNIQUE
            (
                [lastStartTime]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table membership created.'
    END
ELSE
    PRINT 'Table membership exists, no operation required...'
> OK
> Time: 0.062s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
```

```
    WHERE name = 'policies'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table policies does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[policies](
            queue       VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT NULL,
            policyType  VARCHAR(256) NOT NULL,
            params      VARBINARY(6000) NOT NULL,

            CONSTRAINT [pk_queue] PRIMARY KEY
            (
                [queue]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table policies created.'
    END
ELSE
    PRINT 'Table policies exists, no operation required...'
> OK
> Time: 0.047s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'reservationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table reservationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[reservationsHomeSubCluster](
            reservationId   VARCHAR(128) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createResTime DEFAULT
GETUTCDATE(),
```

```
            CONSTRAINT [pk_reservationId] PRIMARY KEY
            (
                [reservationId]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table reservationsHomeSubCluster created.'
    END
ELSE
    PRINT 'Table reservationsHomeSubCluster exists, no operation required...'
> OK
> Time: 0.047s
```

1.3.2 Stored Procedure

Verify that the stored procedure script (FederationStateStoreStoredProcs.sql) can be executed normally.

```
USE [FederationStateStore]
> OK
> Time: 0.045s


IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_addApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                      FROM [dbo].[applicationsHomeSubCluster]
                      WHERE [applicationId] = @applicationId)

                INSERT INTO [dbo].[applicationsHomeSubCluster] (
                    [applicationId],
```

```
                    [homeSubCluster])
                VALUES (
                    @applicationId,
                    @homeSubCluster);
            -- End of the IF block

            SELECT @rowCount = @@ROWCOUNT;

            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.065s


IF OBJECT_ID ( '[sp_updateApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateApplicationHomeSubCluster];
> OK
> Time: 0.052s


CREATE PROCEDURE [dbo].[sp_updateApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[applicationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
```

```
            WHERE [applicationId] = @applicationid;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.043s


IF OBJECT_ID ( '[sp_getApplicationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationsHomeSubCluster];
> OK
> Time: 0.042s


CREATE PROCEDURE [dbo].[sp_getApplicationsHomeSubCluster]
    @limit int,
    @homeSubCluster VARCHAR(256)
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT
            [applicationId],
            [homeSubCluster],
            [createTime]
        FROM
        (SELECT
            [applicationId],
            [homeSubCluster],
            [createTime],
            row_number() over(partition by [homeSubCluster] order by [createTime] desc)
AS row_num
        FROM [dbo].[applicationsHomeSubCluster]) AS t
        WHERE row_num <= @limit
          AND (CASE WHEN @homeSubCluster IS NULL THEN 1
```

```
                    WHEN @homeSubCluster IS NOT NULL AND [homeSubCluster] =
@homeSubCluster THEN 1
                ELSE 0 END) = 1
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.042s


IF OBJECT_ID ( '[sp_getApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationHomeSubCluster];
> OK
> Time: 0.041s


CREATE PROCEDURE [dbo].[sp_getApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[applicationsHomeSubCluster]
        WHERE [applicationId] = @applicationid;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
```

```
END;
> OK
> Time: 0.040s


IF OBJECT_ID ( '[sp_deleteApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteApplicationHomeSubCluster];
> OK
> Time: 0.041s


CREATE PROCEDURE [dbo].[sp_deleteApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.043s


IF OBJECT_ID ( '[sp_registerSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_registerSubCluster];
> OK
> Time: 0.045s
```

```
CREATE PROCEDURE [dbo].[sp_registerSubCluster]
    @subClusterId VARCHAR(256),
    @amRMServiceAddress VARCHAR(256),
    @clientRMServiceAddress VARCHAR(256),
    @rmAdminServiceAddress VARCHAR(256),
    @rmWebServiceAddress VARCHAR(256),
    @state VARCHAR(32),
    @lastStartTime BIGINT,
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId;
            INSERT INTO [dbo].[membership] (
                [subClusterId],
                [amRMServiceAddress],
                [clientRMServiceAddress],
                [rmAdminServiceAddress],
                [rmWebServiceAddress],
                [lastHeartBeat],
                [state],
                [lastStartTime],
                [capability] )
            VALUES (
                @subClusterId,
                @amRMServiceAddress,
                @clientRMServiceAddress,
                @rmAdminServiceAddress,
                @rmWebServiceAddress,
                GETUTCDATE(),
                @state,
                @lastStartTime,
                @capability);
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
```

```
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.041s


IF OBJECT_ID ( '[sp_getSubClusters]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubClusters];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_getSubClusters]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [subClusterId], [amRMServiceAddress], [clientRMServiceAddress],
               [rmAdminServiceAddress], [rmWebServiceAddress], [lastHeartBeat],
               [state], [lastStartTime], [capability]
        FROM [dbo].[membership]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.040s


IF OBJECT_ID ( '[sp_getSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubCluster];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_getSubCluster]
    @subClusterId VARCHAR(256),
```

```
    @amRMServiceAddress VARCHAR(256) OUTPUT,
    @clientRMServiceAddress VARCHAR(256) OUTPUT,
    @rmAdminServiceAddress VARCHAR(256) OUTPUT,
    @rmWebServiceAddress VARCHAR(256) OUTPUT,
    @lastHeartbeat DATETIME2 OUTPUT,
    @state VARCHAR(256) OUTPUT,
    @lastStartTime BIGINT OUTPUT,
    @capability VARCHAR(6000) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
        BEGIN TRAN

            SELECT @subClusterId = [subClusterId],
                   @amRMServiceAddress = [amRMServiceAddress],
                   @clientRMServiceAddress = [clientRMServiceAddress],
                   @rmAdminServiceAddress = [rmAdminServiceAddress],
                   @rmWebServiceAddress = [rmWebServiceAddress],
                   @lastHeartbeat = [lastHeartBeat],
                   @state = [state],
                   @lastStartTime = [lastStartTime],
                   @capability = [capability]
            FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.047s


IF OBJECT_ID ( '[sp_subClusterHeartbeat]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_subClusterHeartbeat];
> OK
```

```
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_subClusterHeartbeat]
    @subClusterId VARCHAR(256),
    @state VARCHAR(256),
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[membership]
            SET [state] = @state,
                [lastHeartBeat] = GETUTCDATE(),
                [capability] = @capability
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.046s


IF OBJECT_ID ( '[sp_deregisterSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deregisterSubCluster];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_deregisterSubCluster]
    @subClusterId VARCHAR(256),
    @state VARCHAR(256),
```

```
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[membership]
            SET [state] = @state
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.041s


IF OBJECT_ID ( '[sp_setPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_setPolicyConfiguration];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_setPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256),
    @params VARBINARY(512),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[policies]
```

```
            WHERE [queue] = @queue;
            INSERT INTO [dbo].[policies] (
                [queue],
                [policyType],
                [params])
            VALUES (
                @queue,
                @policyType,
                @params);
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.040s


IF OBJECT_ID ( '[sp_getPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPolicyConfiguration];
> OK
> Time: 0.042s


CREATE PROCEDURE [dbo].[sp_getPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256) OUTPUT,
    @params VARBINARY(6000) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @policyType = [policyType],
               @params = [params]
        FROM [dbo].[policies]
        WHERE [queue] = @queue
```

```
    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.041s


IF OBJECT_ID ( '[sp_getPoliciesConfigurations]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPoliciesConfigurations];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_getPoliciesConfigurations]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT  [queue], [policyType], [params] FROM [dbo].[policies]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.040s


IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
> OK
```

```
> Time: 0.043s


CREATE PROCEDURE [dbo].[sp_addReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                       FROM [dbo].[reservationsHomeSubCluster]
                       WHERE [reservationId] = @reservationId)

                INSERT INTO [dbo].[reservationsHomeSubCluster] (
                    [reservationId],
                    [homeSubCluster])
                VALUES (
                    @reservationId,
                    @homeSubCluster);
            -- End of the IF block

            SELECT @rowCount = @@ROWCOUNT;

            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
```

```
> Time: 0.043s


IF OBJECT_ID ( '[sp_updateReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateReservationHomeSubCluster];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_updateReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[reservationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.040s


IF OBJECT_ID ( '[sp_getReservationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationsHomeSubCluster];
> OK
> Time: 0.040s
```

```
CREATE PROCEDURE [dbo].[sp_getReservationsHomeSubCluster]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [reservationId], [homeSubCluster], [createTime]
        FROM [dbo].[reservationsHomeSubCluster]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.040s


IF OBJECT_ID ( '[sp_getReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationHomeSubCluster];
> OK
> Time: 0.041s


CREATE PROCEDURE [dbo].[sp_getReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[reservationsHomeSubCluster]
        WHERE [reservationId] = @reservationId;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
```

```
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.040s


IF OBJECT_ID ( '[sp_deleteReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteReservationHomeSubCluster];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_deleteReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.041s
```

1.4 SQL Server 2017 Enterprise

1.4.1 Table

Verify that the table creation script (FederationStateStoreTables.sql) can be executed normally

```
SELECT @@VERSION
> OK
> Time: 0.043s


Microsoft SQL Server 2017 (RTM-CU14) (KB4484710) - 14.0.3076.1 (X64)
Mar 12 2019 19:29:19
Copyright (C) 2017 Microsoft Corporation
Enterprise Edition: Core-based Licensing (64-bit) on Windows Server 2012 R2 Standard
6.3 <X64> (Build 9600: )

USE [FederationStateStore]
> OK
> Time: 0.036s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'applicationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table applicationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[applicationsHomeSubCluster](
            applicationId   VARCHAR(64) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createAppTime DEFAULT
GETUTCDATE(),

            CONSTRAINT [pk_applicationId] PRIMARY KEY
            (
                [applicationId]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table applicationsHomeSubCluster created.'
    END
ELSE
```

```
    PRINT 'Table applicationsHomeSubCluster exists, no operation required...'
> OK
> Time: 0.053s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'membership'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table membership does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[membership](
            [subClusterId]            VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT
NULL,
            [amRMServiceAddress]      VARCHAR(256) NOT NULL,
            [clientRMServiceAddress]  VARCHAR(256) NOT NULL,
            [rmAdminServiceAddress]   VARCHAR(256) NOT NULL,
            [rmWebServiceAddress]     VARCHAR(256) NOT NULL,
            [lastHeartBeat]           DATETIME2 NOT NULL,
            [state]                   VARCHAR(32) NOT NULL,
            [lastStartTime]           BIGINT NOT NULL,
            [capability]              VARCHAR(6000) NOT NULL,

            CONSTRAINT [pk_subClusterId] PRIMARY KEY
            (
                [subClusterId]
            ),
            CONSTRAINT [uc_lastStartTime] UNIQUE
            (
                [lastStartTime]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table membership created.'
    END
ELSE
    PRINT 'Table membership exists, no operation required...'
> OK
> Time: 0.049s
```

```
IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'policies'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table policies does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[policies](
            queue       VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT NULL,
            policyType  VARCHAR(256) NOT NULL,
            params      VARBINARY(6000) NOT NULL,

            CONSTRAINT [pk_queue] PRIMARY KEY
            (
                [queue]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table policies created.'
    END
ELSE
    PRINT 'Table policies exists, no operation required...'
> OK
> Time: 0.049s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'reservationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table reservationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[reservationsHomeSubCluster](
            reservationId   VARCHAR(128) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
```

```
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createResTime DEFAULT
GETUTCDATE(),

            CONSTRAINT [pk_reservationId] PRIMARY KEY
            (
                [reservationId]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table reservationsHomeSubCluster created.'
    END
ELSE
    PRINT 'Table reservationsHomeSubCluster exists, no operation required...'
> OK
> Time: 0.045s
```

1.4.2 Stored Procedure

Verify that the stored procedure script (FederationStateStoreStoredProcs.sql) can be executed normally.

```
USE [FederationStateStore]
> OK
> Time: 0.037s


IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_addApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                       FROM [dbo].[applicationsHomeSubCluster]
                       WHERE [applicationId] = @applicationId)
```

```
                INSERT INTO [dbo].[applicationsHomeSubCluster] (
                    [applicationId],
                    [homeSubCluster])
                VALUES (
                    @applicationId,
                    @homeSubCluster);
            -- End of the IF block

            SELECT @rowCount = @@ROWCOUNT;

            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.038s


IF OBJECT_ID ( '[sp_updateApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateApplicationHomeSubCluster];
> OK
> Time: 0.036s


CREATE PROCEDURE [dbo].[sp_updateApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
```

```
            UPDATE [dbo].[applicationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
            WHERE [applicationId] = @applicationid;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.038s


IF OBJECT_ID ( '[sp_getApplicationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationsHomeSubCluster];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_getApplicationsHomeSubCluster]
    @limit int,
    @homeSubCluster VARCHAR(256)
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT
            [applicationId],
            [homeSubCluster],
            [createTime]
        FROM
        (SELECT
            [applicationId],
            [homeSubCluster],
            [createTime],
            row_number() over(partition by [homeSubCluster] order by [createTime] desc)
AS row_num
        FROM [dbo].[applicationsHomeSubCluster]) AS t
```

```
        WHERE row_num <= @limit
          AND (CASE WHEN @homeSubCluster IS NULL THEN 1
                    WHEN @homeSubCluster IS NOT NULL AND [homeSubCluster] =
@homeSubCluster THEN 1
                ELSE 0 END) = 1
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.040s


IF OBJECT_ID ( '[sp_getApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationHomeSubCluster];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_getApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[applicationsHomeSubCluster]
        WHERE [applicationId] = @applicationid;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
```

```
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.041s


IF OBJECT_ID ( '[sp_deleteApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteApplicationHomeSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_deleteApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.040s


IF OBJECT_ID ( '[sp_registerSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_registerSubCluster];
> OK
> Time: 0.037s
```

```
CREATE PROCEDURE [dbo].[sp_registerSubCluster]
    @subClusterId VARCHAR(256),
    @amRMServiceAddress VARCHAR(256),
    @clientRMServiceAddress VARCHAR(256),
    @rmAdminServiceAddress VARCHAR(256),
    @rmWebServiceAddress VARCHAR(256),
    @state VARCHAR(32),
    @lastStartTime BIGINT,
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId;
            INSERT INTO [dbo].[membership] (
                [subClusterId],
                [amRMServiceAddress],
                [clientRMServiceAddress],
                [rmAdminServiceAddress],
                [rmWebServiceAddress],
                [lastHeartBeat],
                [state],
                [lastStartTime],
                [capability] )
            VALUES (
                @subClusterId,
                @amRMServiceAddress,
                @clientRMServiceAddress,
                @rmAdminServiceAddress,
                @rmWebServiceAddress,
                GETUTCDATE(),
                @state,
                @lastStartTime,
                @capability);
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())
```

```
        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s


IF OBJECT_ID ( '[sp_getSubClusters]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubClusters];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_getSubClusters]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [subClusterId], [amRMServiceAddress], [clientRMServiceAddress],
               [rmAdminServiceAddress], [rmWebServiceAddress], [lastHeartBeat],
               [state], [lastStartTime], [capability]
        FROM [dbo].[membership]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.038s


IF OBJECT_ID ( '[sp_getSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubCluster];
> OK
> Time: 0.037s
```

```
CREATE PROCEDURE [dbo].[sp_getSubCluster]
    @subClusterId VARCHAR(256),
    @amRMServiceAddress VARCHAR(256) OUTPUT,
    @clientRMServiceAddress VARCHAR(256) OUTPUT,
    @rmAdminServiceAddress VARCHAR(256) OUTPUT,
    @rmWebServiceAddress VARCHAR(256) OUTPUT,
    @lastHeartbeat DATETIME2 OUTPUT,
    @state VARCHAR(256) OUTPUT,
    @lastStartTime BIGINT OUTPUT,
    @capability VARCHAR(6000) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            SELECT @subClusterId = [subClusterId],
                   @amRMServiceAddress = [amRMServiceAddress],
                   @clientRMServiceAddress = [clientRMServiceAddress],
                   @rmAdminServiceAddress = [rmAdminServiceAddress],
                   @rmWebServiceAddress = [rmWebServiceAddress],
                   @lastHeartBeat = [lastHeartBeat],
                   @state = [state],
                   @lastStartTime = [lastStartTime],
                   @capability = [capability]
            FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.039s


IF OBJECT_ID ( '[sp_subClusterHeartbeat]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_subClusterHeartbeat];
```

```
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_subClusterHeartbeat]
    @subClusterId VARCHAR(256),
    @state VARCHAR(256),
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[membership]
            SET [state] = @state,
                [lastHeartbeat] = GETUTCDATE(),
                [capability] = @capability
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.042s


IF OBJECT_ID ( '[sp_deregisterSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deregisterSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_deregisterSubCluster]
    @subClusterId VARCHAR(256),
```

```
    @state VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[membership]
            SET [state] = @state
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.040s


IF OBJECT_ID ( '[sp_setPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_setPolicyConfiguration];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_setPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256),
    @params VARBINARY(512),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
```

```
            DELETE FROM [dbo].[policies]
            WHERE [queue] = @queue;
            INSERT INTO [dbo].[policies] (
                [queue],
                [policyType],
                [params])
            VALUES (
                @queue,
                @policyType,
                @params);
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.038s


IF OBJECT_ID ( '[sp_getPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPolicyConfiguration];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_getPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256) OUTPUT,
    @params VARBINARY(6000) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @policyType = [policyType],
               @params = [params]
        FROM [dbo].[policies]
```

```
        WHERE [queue] = @queue

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.041s


IF OBJECT_ID ( '[sp_getPoliciesConfigurations]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPoliciesConfigurations];
> OK
> Time: 0.036s


CREATE PROCEDURE [dbo].[sp_getPoliciesConfigurations]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT  [queue], [policyType], [params] FROM [dbo].[policies]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.038s


IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
```

```
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_addReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                      FROM [dbo].[reservationsHomeSubCluster]
                      WHERE [reservationId] = @reservationId)

                INSERT INTO [dbo].[reservationsHomeSubCluster] (
                    [reservationId],
                    [homeSubCluster])
                VALUES (
                    @reservationId,
                    @homeSubCluster);
            -- End of the IF block

            SELECT @rowCount = @@ROWCOUNT;

            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
```

```
> Affected rows: 0
> Time: 0.038s


IF OBJECT_ID ( '[sp_updateReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateReservationHomeSubCluster];
> OK
> Time: 0.036s


CREATE PROCEDURE [dbo].[sp_updateReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[reservationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s


IF OBJECT_ID ( '[sp_getReservationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationsHomeSubCluster];
> OK
> Time: 0.038s
```

```
CREATE PROCEDURE [dbo].[sp_getReservationsHomeSubCluster]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [reservationId], [homeSubCluster], [createTime]
        FROM [dbo].[reservationsHomeSubCluster]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.038s


IF OBJECT_ID ( '[sp_getReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationHomeSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_getReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[reservationsHomeSubCluster]
        WHERE [reservationId] = @reservationId;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
```

```
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.037s


IF OBJECT_ID ( '[sp_deleteReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteReservationHomeSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_deleteReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.041s
```

1.5 SQL Server 2019 Enterprise

1.5.1 Table

Verify that the table creation script (FederationStateStoreTables.sql) can be executed normally

```
SELECT @@VERSION
> OK
> Time: 0.038s

Microsoft SQL Server 2019 (RTM-CU10) (KB5001090) - 15.0.4123.1 (X64)
Mar 22 2021 18:10:24
Copyright (C) 2019 Microsoft Corporation
Enterprise Edition: Core-based Licensing (64-bit) on Windows Server 2016 Datacenter
10.0 <X64> (Build 14393: )

USE [FederationStateStore]
> OK
> Time: 0.037s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'applicationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table applicationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[applicationsHomeSubCluster](
            applicationId   VARCHAR(64) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createAppTime DEFAULT
GETUTCDATE(),

            CONSTRAINT [pk_applicationId] PRIMARY KEY
            (
                [applicationId]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table applicationsHomeSubCluster created.'
    END
ELSE
    PRINT 'Table applicationsHomeSubCluster exists, no operation required...'
```

```
> OK
> Time: 0.054s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'membership'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table membership does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[membership](
            [subClusterId]            VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT
NULL,
            [amRMServiceAddress]      VARCHAR(256) NOT NULL,
            [clientRMServiceAddress]  VARCHAR(256) NOT NULL,
            [rmAdminServiceAddress]   VARCHAR(256) NOT NULL,
            [rmWebServiceAddress]     VARCHAR(256) NOT NULL,
            [lastHeartBeat]           DATETIME2 NOT NULL,
            [state]                   VARCHAR(32) NOT NULL,
            [lastStartTime]           BIGINT NOT NULL,
            [capability]              VARCHAR(6000) NOT NULL,

            CONSTRAINT [pk_subClusterId] PRIMARY KEY
            (
                [subClusterId]
            ),
            CONSTRAINT [uc_lastStartTime] UNIQUE
            (
                [lastStartTime]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table membership created.'
    END
ELSE
    PRINT 'Table membership exists, no operation required...'
> OK
> Time: 0.050s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
```

```
    WHERE name = 'policies'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table policies does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[policies](
            queue       VARCHAR(256) COLLATE Latin1_General_100_BIN2 NOT NULL,
            policyType  VARCHAR(256) NOT NULL,
            params      VARBINARY(6000) NOT NULL,

            CONSTRAINT [pk_queue] PRIMARY KEY
            (
                [queue]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table policies created.'
    END
ELSE
    PRINT 'Table policies exists, no operation required...'
> OK
> Time: 0.051s


IF NOT EXISTS ( SELECT * FROM [FederationStateStore].sys.tables
    WHERE name = 'reservationsHomeSubCluster'
    AND schema_id = SCHEMA_ID('dbo'))
    BEGIN
        PRINT 'Table reservationsHomeSubCluster does not exist, create it...'

        SET ANSI_NULLS ON

        SET QUOTED_IDENTIFIER ON

        SET ANSI_PADDING ON

        CREATE TABLE [dbo].[reservationsHomeSubCluster](
            reservationId   VARCHAR(128) COLLATE Latin1_General_100_BIN2 NOT NULL,
            homeSubCluster  VARCHAR(256) NOT NULL,
            createTime      DATETIME2 NOT NULL CONSTRAINT ts_createResTime DEFAULT
GETUTCDATE(),
```

```
            CONSTRAINT [pk_reservationId] PRIMARY KEY
            (
                [reservationId]
            )
        )

        SET ANSI_PADDING OFF

        PRINT 'Table reservationsHomeSubCluster created.'
    END
ELSE
    PRINT 'Table reservationsHomeSubCluster exists, no operation required...'
> OK
> Time: 0.046s
```

1.5.2 Stored Procedure

Verify that the stored procedure script (FederationStateStoreStoredProcs.sql) can be executed normally.

```
USE [FederationStateStore]
> OK
> Time: 0.050s

IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
> OK
> Time: 0.037s

CREATE PROCEDURE [dbo].[sp_addApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                      FROM [dbo].[applicationsHomeSubCluster]
                      WHERE [applicationId] = @applicationId)

                INSERT INTO [dbo].[applicationsHomeSubCluster] (
                    [applicationId],
```

```
                    [homeSubCluster])
                VALUES (
                    @applicationId,
                    @homeSubCluster);
            -- End of the IF block

            SELECT @rowCount = @@ROWCOUNT;

            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s


IF OBJECT_ID ( '[sp_updateApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateApplicationHomeSubCluster];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_updateApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[applicationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
```

```
            WHERE [applicationId] = @applicationid;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.041s


IF OBJECT_ID ( '[sp_getApplicationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationsHomeSubCluster];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_getApplicationsHomeSubCluster]
    @limit int,
    @homeSubCluster VARCHAR(256)
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT
            [applicationId],
            [homeSubCluster],
            [createTime]
        FROM
        (SELECT
            [applicationId],
            [homeSubCluster],
            [createTime],
            row_number() over(partition by [homeSubCluster] order by [createTime] desc)
AS row_num
        FROM [dbo].[applicationsHomeSubCluster]) AS t
        WHERE row_num <= @limit
          AND (CASE WHEN @homeSubCluster IS NULL THEN 1
```

```
                    WHEN @homeSubCluster IS NOT NULL AND [homeSubCluster] =
@homeSubCluster THEN 1
                ELSE 0 END) = 1
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.040s


IF OBJECT_ID ( '[sp_getApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getApplicationHomeSubCluster];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_getApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[applicationsHomeSubCluster]
        WHERE [applicationId] = @applicationid;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
```

```
END;
> OK
> Time: 0.039s


IF OBJECT_ID ( '[sp_deleteApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteApplicationHomeSubCluster];
> OK
> Time: 0.041s


CREATE PROCEDURE [dbo].[sp_deleteApplicationHomeSubCluster]
    @applicationId VARCHAR(64),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[applicationsHomeSubCluster]
            WHERE [applicationId] = @applicationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s


IF OBJECT_ID ( '[sp_registerSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_registerSubCluster];
> OK
> Time: 0.037s
```

```
CREATE PROCEDURE [dbo].[sp_registerSubCluster]
    @subClusterId VARCHAR(256),
    @amRMServiceAddress VARCHAR(256),
    @clientRMServiceAddress VARCHAR(256),
    @rmAdminServiceAddress VARCHAR(256),
    @rmWebServiceAddress VARCHAR(256),
    @state VARCHAR(32),
    @lastStartTime BIGINT,
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId;
            INSERT INTO [dbo].[membership] (
                [subClusterId],
                [amRMServiceAddress],
                [clientRMServiceAddress],
                [rmAdminServiceAddress],
                [rmWebServiceAddress],
                [lastHeartBeat],
                [state],
                [lastStartTime],
                [capability] )
            VALUES (
                @subClusterId,
                @amRMServiceAddress,
                @clientRMServiceAddress,
                @rmAdminServiceAddress,
                @rmWebServiceAddress,
                GETUTCDATE(),
                @state,
                @lastStartTime,
                @capability);
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
```

```
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s


IF OBJECT_ID ( '[sp_getSubClusters]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubClusters];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_getSubClusters]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [subClusterId], [amRMServiceAddress], [clientRMServiceAddress],
               [rmAdminServiceAddress], [rmWebServiceAddress], [lastHeartBeat],
               [state], [lastStartTime], [capability]
        FROM [dbo].[membership]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.039s


IF OBJECT_ID ( '[sp_getSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getSubCluster];
> OK
> Time: 0.036s


CREATE PROCEDURE [dbo].[sp_getSubCluster]
    @subClusterId VARCHAR(256),
```

```
    @amRMServiceAddress VARCHAR(256) OUTPUT,
    @clientRMServiceAddress VARCHAR(256) OUTPUT,
    @rmAdminServiceAddress VARCHAR(256) OUTPUT,
    @rmWebServiceAddress VARCHAR(256) OUTPUT,
    @lastHeartbeat DATETIME2 OUTPUT,
    @state VARCHAR(256) OUTPUT,
    @lastStartTime BIGINT OUTPUT,
    @capability VARCHAR(6000) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            SELECT @subClusterId = [subClusterId],
                   @amRMServiceAddress = [amRMServiceAddress],
                   @clientRMServiceAddress = [clientRMServiceAddress],
                   @rmAdminServiceAddress = [rmAdminServiceAddress],
                   @rmWebServiceAddress = [rmWebServiceAddress],
                   @lastHeartBeat = [lastHeartBeat],
                   @state = [state],
                   @lastStartTime = [lastStartTime],
                   @capability = [capability]
            FROM [dbo].[membership]
            WHERE [subClusterId] = @subClusterId

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.039s


IF OBJECT_ID ( '[sp_subClusterHeartbeat]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_subClusterHeartbeat];
> OK
> Time: 0.037s
```

```
CREATE PROCEDURE [dbo].[sp_subClusterHeartbeat]
    @subClusterId VARCHAR(256),
    @state VARCHAR(256),
    @capability VARCHAR(6000),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[membership]
            SET [state] = @state,
                [lastHeartbeat] = GETUTCDATE(),
                [capability] = @capability
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s


IF OBJECT_ID ( '[sp_deregisterSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deregisterSubCluster];
> OK
> Time: 0.039s


CREATE PROCEDURE [dbo].[sp_deregisterSubCluster]
    @subClusterId VARCHAR(256),
    @state VARCHAR(256),
    @rowCount int OUTPUT
```

```
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[membership]
            SET [state] = @state
            WHERE [subClusterId] = @subClusterId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.038s


IF OBJECT_ID ( '[sp_setPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_setPolicyConfiguration];
> OK
> Time: 0.036s


CREATE PROCEDURE [dbo].[sp_setPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256),
    @params VARBINARY(512),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[policies]
            WHERE [queue] = @queue;
```

```
            INSERT INTO [dbo].[policies] (
                [queue],
                [policyType],
                [params])
            VALUES (
                @queue,
                @policyType,
                @params);
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s


IF OBJECT_ID ( '[sp_getPolicyConfiguration]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPolicyConfiguration];
> OK
> Time: 0.036s


CREATE PROCEDURE [dbo].[sp_getPolicyConfiguration]
    @queue VARCHAR(256),
    @policyType VARCHAR(256) OUTPUT,
    @params VARBINARY(6000) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @policyType = [policyType],
               @params = [params]
        FROM [dbo].[policies]
        WHERE [queue] = @queue
```

```
    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.038s


IF OBJECT_ID ( '[sp_getPoliciesConfigurations]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getPoliciesConfigurations];
> OK
> Time: 0.037s


CREATE PROCEDURE [dbo].[sp_getPoliciesConfigurations]
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT  [queue], [policyType], [params] FROM [dbo].[policies]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.038s


IF OBJECT_ID ( '[sp_addApplicationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_addApplicationHomeSubCluster];
> OK
> Time: 0.041s
```

```
CREATE PROCEDURE [dbo].[sp_addReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
    @storedHomeSubCluster VARCHAR(256) OUTPUT,
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN
            -- If application to sub-cluster map doesn't exist, insert it.
            -- Otherwise don't change the current mapping.
            IF NOT EXISTS (SELECT TOP 1 *
                       FROM [dbo].[reservationsHomeSubCluster]
                       WHERE [reservationId] = @reservationId)

                INSERT INTO [dbo].[reservationsHomeSubCluster] (
                    [reservationId],
                    [homeSubCluster])
                VALUES (
                    @reservationId,
                    @homeSubCluster);
            -- End of the IF block

            SELECT @rowCount = @@ROWCOUNT;

            SELECT @storedHomeSubCluster = [homeSubCluster]
            FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s
```

```
IF OBJECT_ID ( '[sp_updateReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_updateReservationHomeSubCluster];
> OK
> Time: 0.042s


CREATE PROCEDURE [dbo].[sp_updateReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            UPDATE [dbo].[reservationsHomeSubCluster]
            SET [homeSubCluster] = @homeSubCluster
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.038s


IF OBJECT_ID ( '[sp_getReservationsHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationsHomeSubCluster];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_getReservationsHomeSubCluster]
```

```
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        SELECT [reservationId], [homeSubCluster], [createTime]
        FROM [dbo].[reservationsHomeSubCluster]
    END TRY

    BEGIN CATCH
        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.038s


IF OBJECT_ID ( '[sp_getReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_getReservationHomeSubCluster];
> OK
> Time: 0.036s


CREATE PROCEDURE [dbo].[sp_getReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @homeSubCluster VARCHAR(256) OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY

        SELECT @homeSubCluster = [homeSubCluster]
        FROM [dbo].[reservationsHomeSubCluster]
        WHERE [reservationId] = @reservationId;

    END TRY

    BEGIN CATCH

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
```

```
            -1 -- State
        ) WITH log
    END CATCH
END;
> OK
> Time: 0.039s


IF OBJECT_ID ( '[sp_deleteReservationHomeSubCluster]', 'P' ) IS NOT NULL
    DROP PROCEDURE [sp_deleteReservationHomeSubCluster];
> OK
> Time: 0.038s


CREATE PROCEDURE [dbo].[sp_deleteReservationHomeSubCluster]
    @reservationId VARCHAR(128),
    @rowCount int OUTPUT
AS BEGIN
    DECLARE @errorMessage nvarchar(4000)

    BEGIN TRY
        BEGIN TRAN

            DELETE FROM [dbo].[reservationsHomeSubCluster]
            WHERE [reservationId] = @reservationId;
            SELECT @rowCount = @@ROWCOUNT;

        COMMIT TRAN
    END TRY

    BEGIN CATCH
        ROLLBACK TRAN

        SET @errorMessage = dbo.func_FormatErrorMessage(ERROR_MESSAGE(), ERROR_LINE())

        /*  raise error and terminate the execution */
        RAISERROR(@errorMessage, --- Error Message
            1, -- Severity
            -1 -- State
        ) WITH log
    END CATCH
END;
> Affected rows: 0
> Time: 0.039s
```

2.MySQL

MySQL 5.7 and above must be required, MySQL 5.5 and MySQL 5.6 cannot be supported.

MySQL-5.5 and MySQL-5.6 Can't Create Table **membership** and **policies**, for the following reasons:

1.The primary key of the **membership** table is **subClusterId varchar(256)** , Mysql will prompt the following error:
Specified key was too long; max key length is 767 bytes.

2.The primary key of the **policies** table is **subClusterId varchar(256)** , Mysql will prompt the following error:
Specified key was too long; max key length is 767 bytes.

2.1 Mysql5.7

2.1.1 Table

Verify that the table creation script (FederationStateStoreTables.sql) can be executed normally

```
mysql> select version();
+------------------+
| version()        |
+------------------+
| 5.7.36 |
+------------------+
1 row in set (18.10 sec)

CREATE TABLE applicationsHomeSubCluster(
   applicationId varchar(64) NOT NULL,
   homeSubCluster varchar(256) NOT NULL,
   createTime datetime NOT NULL,
   CONSTRAINT pk_applicationId PRIMARY KEY (applicationId)
)
> OK
> Time: 0.018s

CREATE TABLE membership(
   subClusterId varchar(256) NOT NULL,
   amRMServiceAddress varchar(256) NOT NULL,
   clientRMServiceAddress varchar(256) NOT NULL,
   rmAdminServiceAddress varchar(256) NOT NULL,
   rmWebServiceAddress varchar(256) NOT NULL,
   lastHeartBeat datetime NOT NULL,
   state varchar(32) NOT NULL,
   lastStartTime bigint NULL,
   capability varchar(6000),
   CONSTRAINT pk_subClusterId PRIMARY KEY (subClusterId),
   UNIQUE(lastStartTime)
)
> OK
> Time: 0.018s
```

```
CREATE TABLE policies(
   queue varchar(256) NOT NULL,
   policyType varchar(256) NOT NULL,
   params varbinary(32768),
   CONSTRAINT pk_queue PRIMARY KEY (queue)
)
> OK
> Time: 0.014s


CREATE TABLE reservationsHomeSubCluster (
   reservationId varchar(128) NOT NULL,
   homeSubCluster varchar(256) NOT NULL,
   CONSTRAINT pk_reservationId PRIMARY KEY (reservationId)
)
> OK
> Time: 0.014s
```

2.1.2 Stored Procedure

Verify that the stored procedure script (FederationStateStoreStoredProcs.sql) can be executed normally.

```
mysql> USE FederationStateStore;
Database changed

mysql> DELIMITER //

CREATE PROCEDURE sp_registerSubCluster(
   IN subClusterId_IN varchar(256),
   IN amRMServiceAddress_IN varchar(256),
   IN clientRMServiceAddress_IN varchar(256),
   IN rmAdminServiceAddress_IN varchar(256),
   IN rmWebServiceAddress_IN varchar(256),
   IN state_IN varchar(256),
   IN lastStartTime_IN bigint, IN capability_IN varchar(6000),
   OUT rowCount_OUT int)
BEGIN
   DELETE FROM membership WHERE (subClusterId = subClusterId_IN);
   INSERT INTO membership (subClusterId, amRMServiceAddress, clientRMServiceAddress,
          rmAdminServiceAddress, rmWebServiceAddress, lastHeartBeat, state,
lastStartTime, capability)
      VALUES (subClusterId_IN, amRMServiceAddress_IN, clientRMServiceAddress_IN,
      rmAdminServiceAddress_IN, rmWebServiceAddress_IN, NOW(), state_IN,
lastStartTime_IN, capability_IN);
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //
```

```
CREATE PROCEDURE sp_deregisterSubCluster(
    IN subClusterId_IN varchar(256),
    IN state_IN varchar(64),
    OUT rowCount_OUT int)
BEGIN
    UPDATE membership SET state = state_IN
    WHERE (subClusterId = subClusterId_IN AND state != state_IN);
    SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_subClusterHeartbeat(
    IN subClusterId_IN varchar(256), IN state_IN varchar(64),
    IN capability_IN varchar(6000), OUT rowCount_OUT int)
BEGIN
   UPDATE membership
   SET capability = capability_IN,
       state = state_IN,
       lastHeartBeat = NOW()
   WHERE subClusterId = subClusterId_IN;
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_getSubCluster(
   IN subClusterId_IN varchar(256),
   OUT amRMServiceAddress_OUT varchar(256),
   OUT clientRMServiceAddress_OUT varchar(256),
   OUT rmAdminServiceAddress_OUT varchar(256),
   OUT rmWebServiceAddress_OUT varchar(256),
   OUT lastHeartBeat_OUT datetime, OUT state_OUT varchar(64),
   OUT lastStartTime_OUT bigint,
   OUT capability_OUT varchar(6000))
BEGIN
   SELECT amRMServiceAddress, clientRMServiceAddress, rmAdminServiceAddress,
rmWebServiceAddress,
          lastHeartBeat, state, lastStartTime, capability
   INTO amRMServiceAddress_OUT, clientRMServiceAddress_OUT, rmAdminServiceAddress_OUT,
        rmWebServiceAddress_OUT, lastHeartBeat_OUT, state_OUT, lastStartTime_OUT,
capability_OUT
   FROM membership WHERE subClusterId = subClusterId_IN;
END //

CREATE PROCEDURE sp_getSubClusters()
BEGIN
   SELECT subClusterId, amRMServiceAddress, clientRMServiceAddress,
          rmAdminServiceAddress, rmWebServiceAddress, lastHeartBeat,
          state, lastStartTime, capability
   FROM membership;
END //
```

```
CREATE PROCEDURE sp_addApplicationHomeSubCluster(
   IN applicationId_IN varchar(64), IN homeSubCluster_IN varchar(256),
   OUT storedHomeSubCluster_OUT varchar(256), OUT rowCount_OUT int)
BEGIN
   INSERT INTO applicationsHomeSubCluster
      (applicationId,homeSubCluster)
      (SELECT applicationId_IN, homeSubCluster_IN
       FROM applicationsHomeSubCluster
       WHERE applicationId = applicationId_IN
       HAVING COUNT(*) = 0 );
   SELECT ROW_COUNT() INTO rowCount_OUT;
   SELECT homeSubCluster INTO storedHomeSubCluster_OUT
   FROM applicationsHomeSubCluster
   WHERE applicationId = applicationID_IN;
END //

CREATE PROCEDURE sp_updateApplicationHomeSubCluster(
   IN applicationId_IN varchar(64),
   IN homeSubCluster_IN varchar(256), OUT rowCount_OUT int)
BEGIN
   UPDATE applicationsHomeSubCluster
     SET homeSubCluster = homeSubCluster_IN
   WHERE applicationId = applicationId_IN;
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_getApplicationHomeSubCluster(
   IN applicationId_IN varchar(64),
   OUT homeSubCluster_OUT varchar(256))
BEGIN
   SELECT homeSubCluster INTO homeSubCluster_OUT
   FROM applicationsHomeSubCluster
   WHERE applicationId = applicationID_IN;
END //

CREATE PROCEDURE sp_getApplicationsHomeSubCluster(IN limit_IN int, IN homeSubCluster_IN
varchar(256))
BEGIN
   SELECT
       applicationId,
       homeSubCluster,
       createTime
   FROM
    (SELECT
       applicationId,
       homeSubCluster,
       createTime,
       @app_rank := IF(@home_sc = homeSubCluster, @app_rank + 1, 1) AS app_rank,
```

```
       @home_sc := homeSubCluster
     FROM applicationshomesubcluster
     ORDER BY createTime DESC
    ) ranked
   WHERE app_rank <= limit_IN
     AND (CASE WHEN t.homeSubCluster_IN IS NULL THEN 1 = 1
               WHEN t.homeSubCluster_IN IS NOT NULL THEN homeSubCluster =
homeSubCluster_IN
        END);
END //

CREATE PROCEDURE sp_deleteApplicationHomeSubCluster(
   IN applicationId_IN varchar(64), OUT rowCount_OUT int)
BEGIN
   DELETE FROM applicationsHomeSubCluster
   WHERE applicationId = applicationId_IN;
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_setPolicyConfiguration(
   IN queue_IN varchar(256), IN policyType_IN varchar(256),
   IN params_IN varbinary(32768), OUT rowCount_OUT int)
BEGIN
   DELETE FROM policies WHERE queue = queue_IN;
   INSERT INTO policies (queue, policyType, params)
   VALUES (queue_IN, policyType_IN, params_IN);
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_getPoliciesConfigurations()
BEGIN
   SELECT queue, policyType, params FROM policies;
END //

CREATE PROCEDURE sp_getPolicyConfiguration(
   IN queue_IN varchar(256), OUT policyType_OUT varchar(256),
   OUT params_OUT varbinary(32768))
BEGIN
   SELECT policyType, params INTO policyType_OUT, params_OUT
   FROM policies WHERE queue = queue_IN;
END //

CREATE PROCEDURE sp_addReservationHomeSubCluster(
   IN reservationId_IN varchar(128), IN homeSubCluster_IN varchar(256),
   OUT storedHomeSubCluster_OUT varchar(256), OUT rowCount_OUT int)
BEGIN
   INSERT INTO reservationsHomeSubCluster
      (reservationId,homeSubCluster)
      (SELECT reservationId_IN, homeSubCluster_IN
```

```
       FROM applicationsHomeSubCluster
       WHERE reservationId = reservationId_IN
       HAVING COUNT(*) = 0 );
   SELECT ROW_COUNT() INTO rowCount_OUT;
   SELECT homeSubCluster INTO storedHomeSubCluster_OUT
   FROM reservationsHomeSubCluster
   WHERE applicationId = reservationId_IN;
END //

CREATE PROCEDURE sp_getReservationHomeSubCluster(
   IN reservationId_IN varchar(128),
   OUT homeSubCluster_OUT varchar(256))
BEGIN
   SELECT homeSubCluster INTO homeSubCluster_OUT
   FROM reservationsHomeSubCluster
   WHERE reservationId = reservationId_IN;
END //

CREATE PROCEDURE sp_getReservationsHomeSubCluster()
BEGIN
   SELECT reservationId, homeSubCluster
   FROM reservationsHomeSubCluster;
END //

CREATE PROCEDURE sp_updateReservationHomeSubCluster(
   IN reservationId_IN varchar(128),
   IN homeSubCluster_IN varchar(256), OUT rowCount_OUT int)
BEGIN
   UPDATE reservationsHomeSubCluster
     SET homeSubCluster = homeSubCluster_IN
   WHERE reservationId = reservationId_IN;
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_deleteReservationHomeSubCluster(
   IN reservationId_IN varchar(128), OUT rowCount_OUT int)
BEGIN
   DELETE FROM reservationsHomeSubCluster
   WHERE reservationId = reservationId_IN;
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

DELIMITER ;
Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)
```

```
Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)
```

2.2 Mysql5.8

2.2.1 Table

Verify that the table creation script (FederationStateStoreTables.sql) can be executed normally

```
mysql> select version();
+--------------+
| version()    |
+--------------+
| 8.0.22       |
+--------------+
1 row in set (0.02 sec)

mysql> USE FederationStateStore;
Database changed
```

```
mysql> CREATE TABLE applicationsHomeSubCluster(
   applicationId varchar(64) NOT NULL,
   homeSubCluster varchar(256) NOT NULL,
   createTime datetime NOT NULL,
   CONSTRAINT pk_applicationId PRIMARY KEY (applicationId)
);

CREATE TABLE membership(
   subClusterId varchar(256) NOT NULL,
   amRMServiceAddress varchar(256) NOT NULL,
   clientRMServiceAddress varchar(256) NOT NULL,
   rmAdminServiceAddress varchar(256) NOT NULL,
   rmWebServiceAddress varchar(256) NOT NULL,
   lastHeartBeat datetime NOT NULL,
   state varchar(32) NOT NULL,
   lastStartTime bigint NULL,
   capability varchar(6000),
   CONSTRAINT pk_subClusterId PRIMARY KEY (subClusterId),
   UNIQUE(lastStartTime)
);

CREATE TABLE policies(
   queue varchar(256) NOT NULL,
   policyType varchar(256) NOT NULL,
   params varbinary(32768),
   CONSTRAINT pk_queue PRIMARY KEY (queue)
);

CREATE TABLE reservationsHomeSubCluster (
   reservationId varchar(128) NOT NULL,
   homeSubCluster varchar(256) NOT NULL,
   CONSTRAINT pk_reservationId PRIMARY KEY (reservationId)
);
Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.03 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

mysql>
```

2.2.2 Stored Procedure

Verify that the stored procedure script (FederationStateStoreStoredProcs.sql) can be executed normally.

```
mysql> DELIMITER //

CREATE PROCEDURE sp_registerSubCluster(
   IN subClusterId_IN varchar(256),
   IN amRMServiceAddress_IN varchar(256),
   IN clientRMServiceAddress_IN varchar(256),
   IN rmAdminServiceAddress_IN varchar(256),
   IN rmWebServiceAddress_IN varchar(256),
   IN state_IN varchar(256),
   IN lastStartTime_IN bigint, IN capability_IN varchar(6000),
   OUT rowCount_OUT int)
BEGIN
   DELETE FROM membership WHERE (subClusterId = subClusterId_IN);
   INSERT INTO membership (subClusterId, amRMServiceAddress, clientRMServiceAddress,
          rmAdminServiceAddress, rmWebServiceAddress, lastHeartBeat, state,
lastStartTime, capability)
      VALUES (subClusterId_IN, amRMServiceAddress_IN, clientRMServiceAddress_IN,
      rmAdminServiceAddress_IN, rmWebServiceAddress_IN, NOW(), state_IN,
lastStartTime_IN, capability_IN);
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_deregisterSubCluster(
    IN subClusterId_IN varchar(256),
    IN state_IN varchar(64),
    OUT rowCount_OUT int)
BEGIN
    UPDATE membership SET state = state_IN
    WHERE (subClusterId = subClusterId_IN AND state != state_IN);
    SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_subClusterHeartbeat(
    IN subClusterId_IN varchar(256), IN state_IN varchar(64),
    IN capability_IN varchar(6000), OUT rowCount_OUT int)
BEGIN
   UPDATE membership
   SET capability = capability_IN,
       state = state_IN,
       lastHeartBeat = NOW()
   WHERE subClusterId = subClusterId_IN;
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_getSubCluster(
```

```
   IN subClusterId_IN varchar(256),
   OUT amRMServiceAddress_OUT varchar(256),
   OUT clientRMServiceAddress_OUT varchar(256),
   OUT rmAdminServiceAddress_OUT varchar(256),
   OUT rmWebServiceAddress_OUT varchar(256),
   OUT lastHeartBeat_OUT datetime, OUT state_OUT varchar(64),
   OUT lastStartTime_OUT bigint,
   OUT capability_OUT varchar(6000))
BEGIN
   SELECT amRMServiceAddress, clientRMServiceAddress, rmAdminServiceAddress,
rmWebServiceAddress,
          lastHeartBeat, state, lastStartTime, capability
   INTO amRMServiceAddress_OUT, clientRMServiceAddress_OUT, rmAdminServiceAddress_OUT,
        rmWebServiceAddress_OUT, lastHeartBeat_OUT, state_OUT, lastStartTime_OUT,
capability_OUT
   FROM membership WHERE subClusterId = subClusterId_IN;
END //

CREATE PROCEDURE sp_getSubClusters()
BEGIN
   SELECT subClusterId, amRMServiceAddress, clientRMServiceAddress,
          rmAdminServiceAddress, rmWebServiceAddress, lastHeartBeat,
          state, lastStartTime, capability
   FROM membership;
END //

CREATE PROCEDURE sp_addApplicationHomeSubCluster(
   IN applicationId_IN varchar(64), IN homeSubCluster_IN varchar(256),
   OUT storedHomeSubCluster_OUT varchar(256), OUT rowCount_OUT int)
BEGIN
   INSERT INTO applicationsHomeSubCluster
      (applicationId,homeSubCluster)
      (SELECT applicationId_IN, homeSubCluster_IN
       FROM applicationsHomeSubCluster
       WHERE applicationId = applicationId_IN
       HAVING COUNT(*) = 0 );
   SELECT ROW_COUNT() INTO rowCount_OUT;
   SELECT homeSubCluster INTO storedHomeSubCluster_OUT
   FROM applicationsHomeSubCluster
   WHERE applicationId = applicationID_IN;
END //

CREATE PROCEDURE sp_updateApplicationHomeSubCluster(
   IN applicationId_IN varchar(64),
   IN homeSubCluster_IN varchar(256), OUT rowCount_OUT int)
BEGIN
   UPDATE applicationsHomeSubCluster
     SET homeSubCluster = homeSubCluster_IN
   WHERE applicationId = applicationId_IN;
```

```
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_getApplicationHomeSubCluster(
   IN applicationId_IN varchar(64),
   OUT homeSubCluster_OUT varchar(256))
BEGIN
   SELECT homeSubCluster INTO homeSubCluster_OUT
   FROM applicationsHomeSubCluster
   WHERE applicationId = applicationID_IN;
END //

CREATE PROCEDURE sp_getApplicationsHomeSubCluster(IN limit_IN int, IN homeSubCluster_IN
varchar(256))
BEGIN
   SELECT
       applicationId,
       homeSubCluster,
       createTime
   FROM
    (SELECT
       applicationId,
       homeSubCluster,
       createTime,
       @app_rank := IF(@home_sc = homeSubCluster, @app_rank + 1, 1) AS app_rank,
       @home_sc := homeSubCluster
     FROM applicationshomesubcluster
     ORDER BY createTime DESC
    ) ranked
   WHERE app_rank <= limit_IN
     AND (CASE WHEN t.homeSubCluster_IN IS NULL THEN 1 = 1
               WHEN t.homeSubCluster_IN IS NOT NULL THEN homeSubCluster =
homeSubCluster_IN
        END);
END //

CREATE PROCEDURE sp_deleteApplicationHomeSubCluster(
   IN applicationId_IN varchar(64), OUT rowCount_OUT int)
BEGIN
   DELETE FROM applicationsHomeSubCluster
   WHERE applicationId = applicationId_IN;
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_setPolicyConfiguration(
   IN queue_IN varchar(256), IN policyType_IN varchar(256),
   IN params_IN varbinary(32768), OUT rowCount_OUT int)
BEGIN
   DELETE FROM policies WHERE queue = queue_IN;
```

```
   INSERT INTO policies (queue, policyType, params)
   VALUES (queue_IN, policyType_IN, params_IN);
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_getPoliciesConfigurations()
BEGIN
   SELECT queue, policyType, params FROM policies;
END //

CREATE PROCEDURE sp_getPolicyConfiguration(
   IN queue_IN varchar(256), OUT policyType_OUT varchar(256),
   OUT params_OUT varbinary(32768))
BEGIN
   SELECT policyType, params INTO policyType_OUT, params_OUT
   FROM policies WHERE queue = queue_IN;
END //

CREATE PROCEDURE sp_addReservationHomeSubCluster(
   IN reservationId_IN varchar(128), IN homeSubCluster_IN varchar(256),
   OUT storedHomeSubCluster_OUT varchar(256), OUT rowCount_OUT int)
BEGIN
   INSERT INTO reservationsHomeSubCluster
      (reservationId,homeSubCluster)
      (SELECT reservationId_IN, homeSubCluster_IN
       FROM applicationsHomeSubCluster
       WHERE reservationId = reservationId_IN
       HAVING COUNT(*) = 0 );
   SELECT ROW_COUNT() INTO rowCount_OUT;
   SELECT homeSubCluster INTO storedHomeSubCluster_OUT
   FROM reservationsHomeSubCluster
   WHERE applicationId = reservationId_IN;
END //

CREATE PROCEDURE sp_getReservationHomeSubCluster(
   IN reservationId_IN varchar(128),
   OUT homeSubCluster_OUT varchar(256))
BEGIN
   SELECT homeSubCluster INTO homeSubCluster_OUT
   FROM reservationsHomeSubCluster
   WHERE reservationId = reservationId_IN;
END //

CREATE PROCEDURE sp_getReservationsHomeSubCluster()
BEGIN
   SELECT reservationId, homeSubCluster
   FROM reservationsHomeSubCluster;
END //
```

```
CREATE PROCEDURE sp_updateReservationHomeSubCluster(
   IN reservationId_IN varchar(128),
   IN homeSubCluster_IN varchar(256), OUT rowCount_OUT int)
BEGIN
   UPDATE reservationsHomeSubCluster
     SET homeSubCluster = homeSubCluster_IN
   WHERE reservationId = reservationId_IN;
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

CREATE PROCEDURE sp_deleteReservationHomeSubCluster(
   IN reservationId_IN varchar(128), OUT rowCount_OUT int)
BEGIN
   DELETE FROM reservationsHomeSubCluster
   WHERE reservationId = reservationId_IN;
   SELECT ROW_COUNT() INTO rowCount_OUT;
END //

DELIMITER ;
Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.04 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)
```

```
Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

Query OK, 0 rows affected (0.02 sec)

mysql>
```