CAS commands for MySQL

Last updated by | Mohammad Abu Hamdieh | Jan 23, 2023 at 12:46 AM PST

Contents

For MySQL, we can run the CAS commands in DS Console against each different data center to get information from the server as well as update the setting of the server.

In this article, we will cover how to setup the environment and give examples on the common commands that CSS can use.

Environment Setup

1. Launch the XTS.

2. Find out the data center of customer's server, right click on it and choose [Launch DS Consolidated Console]

3. Generally speaking, you will need to wait 1-2 minutes for the console to startup successfully with add-in or update installed successfully. Sometimes it may require you to choose **Yes** for some updates.
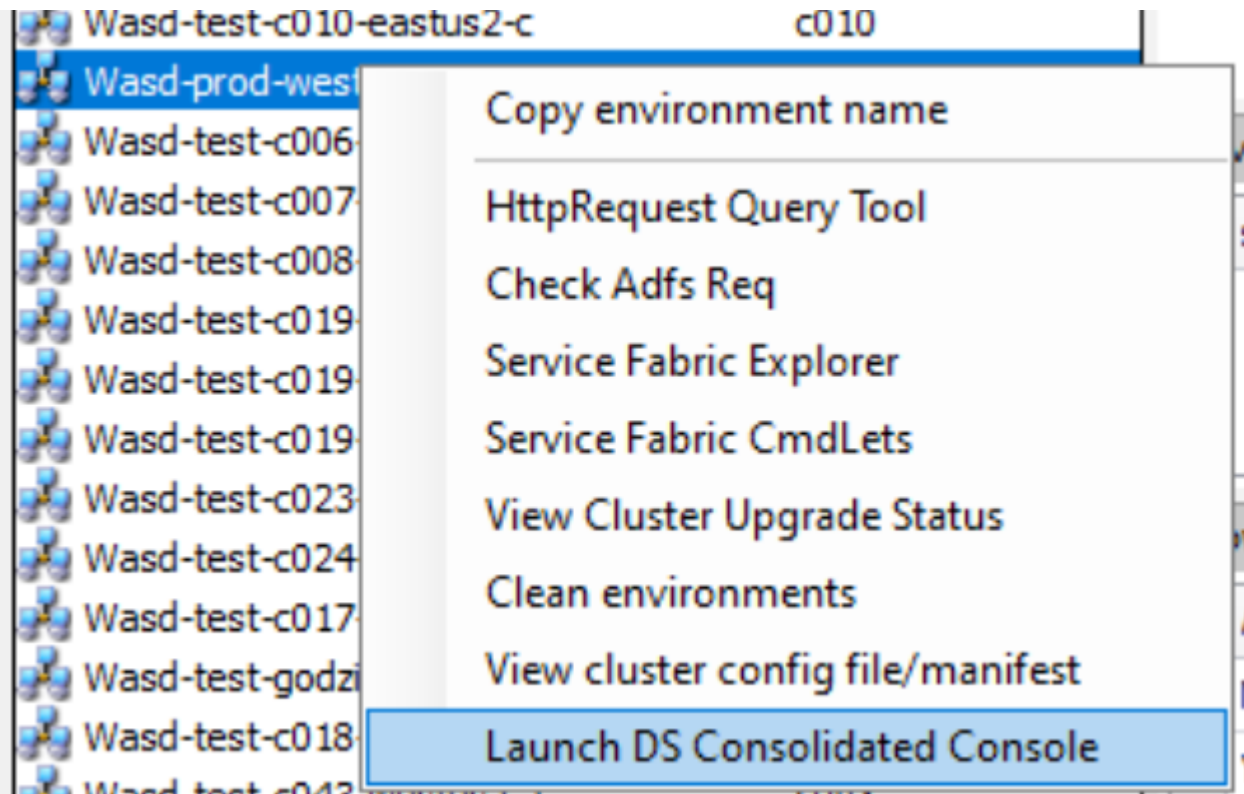
4. After the console launched successfully, you would need to choose the environment and cluster as below. The cluster should be different for different data center. If you have no idea to find the correct cluster name for your server, please go to the sample commands part which also convers it.

```
Select-SqlAzureEnvironment Prod
```

```
Select-SqlAzureCluster Wasd-prod-*westus2-a-CR1*
```

Permission

Please connect to My Access and apply for permission below:

- Azure SQL CSS Std CAS [Read for vendors, ReadWrite for FTE]

Flexible server

Connect to the cluster

Launch the view **mysql servers.xts** and input the server name in the step1 section. After that, you will find these sample commands in the [CAS Actions] part. And you can find the 1st and 2nd command to connect to the correct cluster.

CAS Actions

	action	command
▶	Select Environment	Select-SqlAzureEnvironment Prod
	Select Cluster	Select-SqlAzureCluster Wasd-prod-westus2-a-CR1
	JIT access	ServerName: marlonmysqlf1 , SubscriptionId: 0B321D47-5

CAS commands scenarios

Server Parameters

1. Get the Replication Set ID

   You can use Kusto using below query

   ```
   MonOrcasMySQLBreadthCMSSnapshot
   | where TIMESTAMP > ago(12h)
   | where ha_original_primary_server_name == "servername"
   | where server_state != "Tombstoned"
   | summarize by server_name, replication_set_id
   ```

server_name	replication_set_id
____01	4E1500BB-1A8B-49BD-9CA8-C5B1B286032D

   You can use XTS view **meru\mysql servers.xts**

2. To get all server parameters:

   ```
   Get-MySqlServerparameters -ReplicationsetID 4E1500BB-1A8B-49BD-9CA8-C5B1B286032D
   ```

3. To get any particular Parameter:

   ```
   Get-MySqlServerparameters -ReplicationsetID 4E1500BB-1A8B-49BD-9CA8-C5B1B286032D -ConfigurationName binlo
   ```

```
Name              : binlog_expire_logs_seconds
Value             : 0
DefaultValue      : 0
DataType          : Integer
IsPendingRestart  : False
IsReserved        : False
IsDynamic         : True
AllowedValues     : 0-4294967295
Description       : Number of seconds to service waits before the binary log file gets purged. See:
                    https://docs.microsoft.com/en-us/azure/mysql/flexible-server/concepts-server-parameters#b
Source            : system-default
Tag               : Top
Link              : https://dev.mysql.com/doc/refman/8.0/en/replication-options-binary-
log.html#sysvar_binlog_expire_logs_seconds
```

Review Memory usage and CPU usages on the OS level.

```
Invoke-MySqlServerScriptWithRunCommand  -OrcasInstanceId Orcas_Instance_Id -ServerName MySQL_Server_Name  -Sub


Invoke-MySqlServerScriptWithRunCommand  -OrcasInstanceId Orcas_Instance_Id -ServerName MySQL_Server_Name  -Sub


Invoke-MySqlServerScriptWithRunCommand  -OrcasInstanceId Orcas_Instance_Id -ServerName MySQL_Server_Name  -Sub
```

Check processlist information

```
Invoke-MySqlServerDbCommand -ServerName "MySQL Server Name" -CustomerSubscriptionId "subscription_id" -
CustomerResourceGroup "MySQL" -command show_processlist
```