Connectivity - Using CNAME for the Server FQDN

Last updated by | Subbu Kandhaswamy | Dec 7, 2020 at 2:19 PM PST

Contents

Scenario

Customer is trying to connect to their server using a CNAME for the server name but it requires that they add the server name after the user to be able to connect.

Example:

Azure SQL Database server: Dbserver01.database.windows.net

CNAME for server: AzureDB.contoso.com -> dbserver01.database.windows.net

Login details1:

Server: AzureDB.contoso.com

Username: User@dbserver01

This login attempt will work

Login details2:

Server: AzureDB.contoso.com

Username: User

This fails with this error

Cannot open server "AzureDB.contoso.com" requested by the login. The login failed. (.Net SqlClient Data Provider)

This comes down to two things. When connecting to the server, you are specifying a FQDN. This is used to resolve the Azure DB gateway to connect. So a CNAME that points to server.database.windows.net or even an A record that points to the IP of your server.database.windows.net will work (IP can change). This is simply from resolution need, so a CNAME can resolve when setup correctly, that is just standard behavior of DNS.

The second part being that the connection needs to also identify the specific server it is trying to connect to. In the case of the CNAME, at no point are you indicating the server name when using the standalone user (no @server). So after resolving the CNAME to get to Azure, our gateway has no idea which server you are trying to connect to. When it hits our gateway, we don't necessarily know how you got there so relying on resolution to determine the server is not plausible. So it comes down to being as simple as when trying to connect, we take the highest level portion of the FQDN, in a normal situation, servername (e.g. dbserver01) and can use that to determine the server to connect to. So in this case for the customer, they are using a CNAME that the highest level portion is not the same as the servername (AzureDB vs dbserver01) so it cannot use this and requires it be explicit after the user.

- I tested on my end with my own domain
- First I just did an nslookup on my test-sc.database.windows.net

C:\Users>nslookup test-sc.database.windows.net

Server: UnKnown

Address: 192.168.5.2

Non-authoritative answer:

Name: southcentralus1-a.control.database.windows.net

Address: 23.98.162.75

Aliases: test-sc.database.windows.net

- Tried to connect to the server with just user and using the IP, it failed.

Login details:

Server: 23.98.162.75

Username: User

Cannot open server "23.98.162.75" requested by the login. The login failed. (.Net SqlClient Data Provider)

- If I add @test-sc to the end of my user, test@test-sc, it connects without issue.
- This completely removes a CNAME involvement and demonstrates both DNS resolution but also server name being explicitly needed.
- I then took personal domain, nickdomain.com and added a CNAME, azure.nickdomain.com pointing to test-sc.database.windows.net
- Trying to connect with the user only: test

Cannot open server "azure.nickdomain.com" requested by the login. The login failed. (.Net SqlClient Data Provider)

- Using the @test-sc it worked.

- Created a CNAME with test-sc as the FQDN, test-sc.nickdomain.com
- Trying to connect with the user only, test, it connected without issue.
- This demonstrates that the servername is parsed before being sent to the server once resolved. So for this to work without adding the @server, the CNAME for the customer would need to be qualifactsdbserv01.ssbintel.com for it to work without the @server on the username. I do anticipate that this should cause no issues at this point but for safety sake would still recommend connecting to the normal FQDN as there could be other unforeseen behaviors such as validating certificates. This is not necessarily limited to CNAMES but overall DNS that is being used to point to the FQDN of a server or IP.

How good have you found this content?