SQL Database Firewall rules

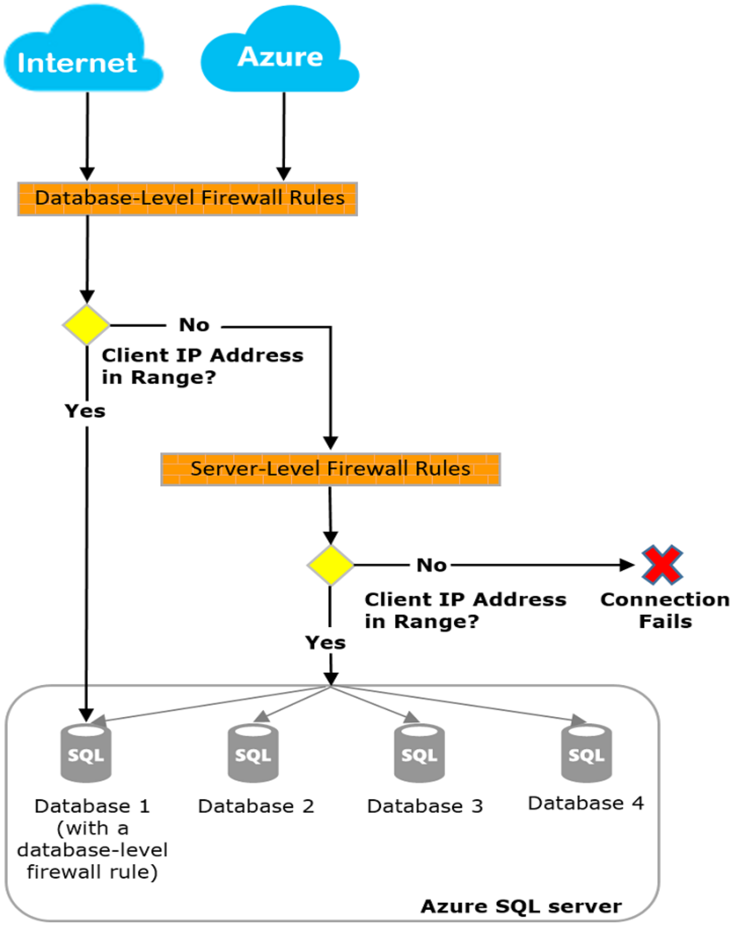
Last updated by | Rui Manuel Maia | Mar 13, 2023 at 8:52 AM PDT

Contents

SQL Database Firewall rules

How the firewall works

Connection attempts from the internet and Azure must pass through the firewall before they reach your SQL server or SQL database, as the following diagram shows.


Internet
Azure
Database-Level Firewall Rules
No
Client IP Address in Range?
Yes
Server-Level Firewall Rules
No
Client IP Address in Range?
Connection Fails
Yes
SQL
Database 1 (with a database-level firewall rule)
Database 2
Database 3
Database 4
Azure SQL server

Server-level IP firewall rules

These rules enable clients to access your entire Azure SQL server, that is, all the databases within the same SQL Database server. The rules are stored in the *master* database. You can have a maximum of 256 server-level IP firewall rules for an Azure SQL Server.

You can configure server-level IP firewall rules by using the Azure portal, PowerShell, or Transact-SQL statements.

- To use the portal or PowerShell, you must be the subscription owner or a subscription contributor.
- To use Transact-SQL, you must connect to the SQL Database instance as the server-level principal login or as the Azure Active Directory administrator. (A server-level IP firewall rule must first be created by a user who has Azure-level permissions.)

Database-level IP firewall rules

These rules enable clients to access certain (secure) databases within the same SQL Database server. You create the rules for each database (including the *master* database), and they're stored in the individual database.
You can only create and manage database-level IP firewall rules for master and user databases by using Transact-SQL statements and only after you configure the first server-level firewall.
If you specify an IP address range in the database-level IP firewall rule that's outside the range in the server-level IP firewall rule, only those clients that have IP addresses in the database-level range can access the database.
You can have a maximum of 256 database-level IP firewall rules for a database. For more information about

configuring database-level IP firewall rules, see the example later in this article and see sp_set_database_firewall_rule (Azure SQL Database) ↗.

Recommendations for how to set firewall rules

We recommend that you use database-level IP firewall rules whenever possible. This practice enhances security and makes your database more portable. Use server-level IP firewall rules for administrators. Also use them when you have many databases that have the same access requirements, and you don't want to configure each database individually.
Note: For information about portable databases in the context of business continuity, see Authentication requirements for disaster recovery ↗.

Server-level versus database-level IP firewall rules

Should users of one database be fully isolated from another database?
If *yes*, use database-level IP firewall rules to grant access. This method avoids using server-level IP firewall rules, which permit access through the firewall to all databases. That would reduce the depth of your defenses.

Do users at the IP addresses need access to all databases?
If *yes*, use server-level IP firewall rules to reduce the number of times that you have to configure IP firewall rules.

Does the person or team who configures the IP firewall rules only have access through the Azure portal, PowerShell, or the REST API?
If so, you must use server-level IP firewall rules. Database-level IP firewall rules can only be configured through Transact-SQL.

Is the person or team who configures the IP firewall rules prohibited from having high-level permission at the database level?
If so, use server-level IP firewall rules. You need at least *CONTROL DATABASE* permission at the database level to configure database-level IP firewall rules through Transact-SQL.

Does the person or team who configures or audits the IP firewall rules centrally manage IP firewall rules for many (perhaps hundreds) of databases?
In this scenario, best practices are determined by your needs and environment. Server-level IP firewall rules might be easier to configure, but scripting can configure rules at the database-level. And even if you use server-level IP firewall rules, you might need to audit database-level IP firewall rules to see if users with *CONTROL* permission on the database create database-level IP firewall rules.

Can I use a mix of server-level and database-level IP firewall rules?
Yes. Some users, such as administrators, might need server-level IP firewall rules. Other users, such as users of a database application, might need database-level IP firewall rules.

Connections from the internet

When a computer tries to connect to your database server from the internet, the firewall first checks the originating IP address of the request against the database-level IP firewall rules for the database that the connection requests.

- If the address is within a range that's specified in the database-level IP firewall rules, the connection is granted to the SQL database that contains the rule.

- If the address isn't within a range in the database-level IP firewall rules, the firewall checks the server-level IP firewall rules. If the address is within a range that's in the server-level IP firewall rules, the connection is granted. Server-level IP firewall rules apply to all SQL databases on the Azure SQL server.
- If the address isn't within a range that's in any of the database-level or server-level IP firewall rules, the connection request fails.

Note: To access SQL Database from your local computer, ensure that the firewall on your network and local computer allow outgoing communication on TCP port 1433.

Connections from inside Azure

To allow applications hosted inside Azure to connect to your SQL server, Azure connections must be enabled. When an application from Azure tries to connect to your database server, the firewall verifies that Azure connections are allowed. A firewall setting that has starting and ending IP addresses equal to *0.0.0.0* indicates that Azure connections are allowed. If the connection isn't allowed, the request doesn't reach the SQL Database server.

Classification

Root Cause: Azure SQL DB v2\Connectivity\Login Errors\Firewall errors and misconfigurations

How good have you found this content?

🙂 🙁