✓ 100 XP

Describe configuring Azure SQL Database, Azure Database for PostgreSQL, and Azure Database for MySQL

7 minutes

This unit explores the specific configuration options available to each type of data store within Azure relational data services.

Configure Azure SQL Database

The overarching principle for network security of the Azure SQL Database offering is to allow only the connection and communication that is necessary to allow the service to operate. All other ports, protocols, and connections are blocked by default. Virtual local area networks (VLANs) and access control lists (ACLs) are used to restrict network communications by source and destination networks, protocols, and port numbers.

> ⓘ Note
>
> An ACL contains a list of resources, and the objects (users, computers, and applications) that are allowed to access those resources. When an object attempts to use a resource that is protected by an ACL, if it's not in the list, it won't be given access.

Items that implement network-based ACLs include routers and load balancers. You control traffic flow through these items by defining firewall rules.

The following steps describe how a connection is established to an Azure SQL database:

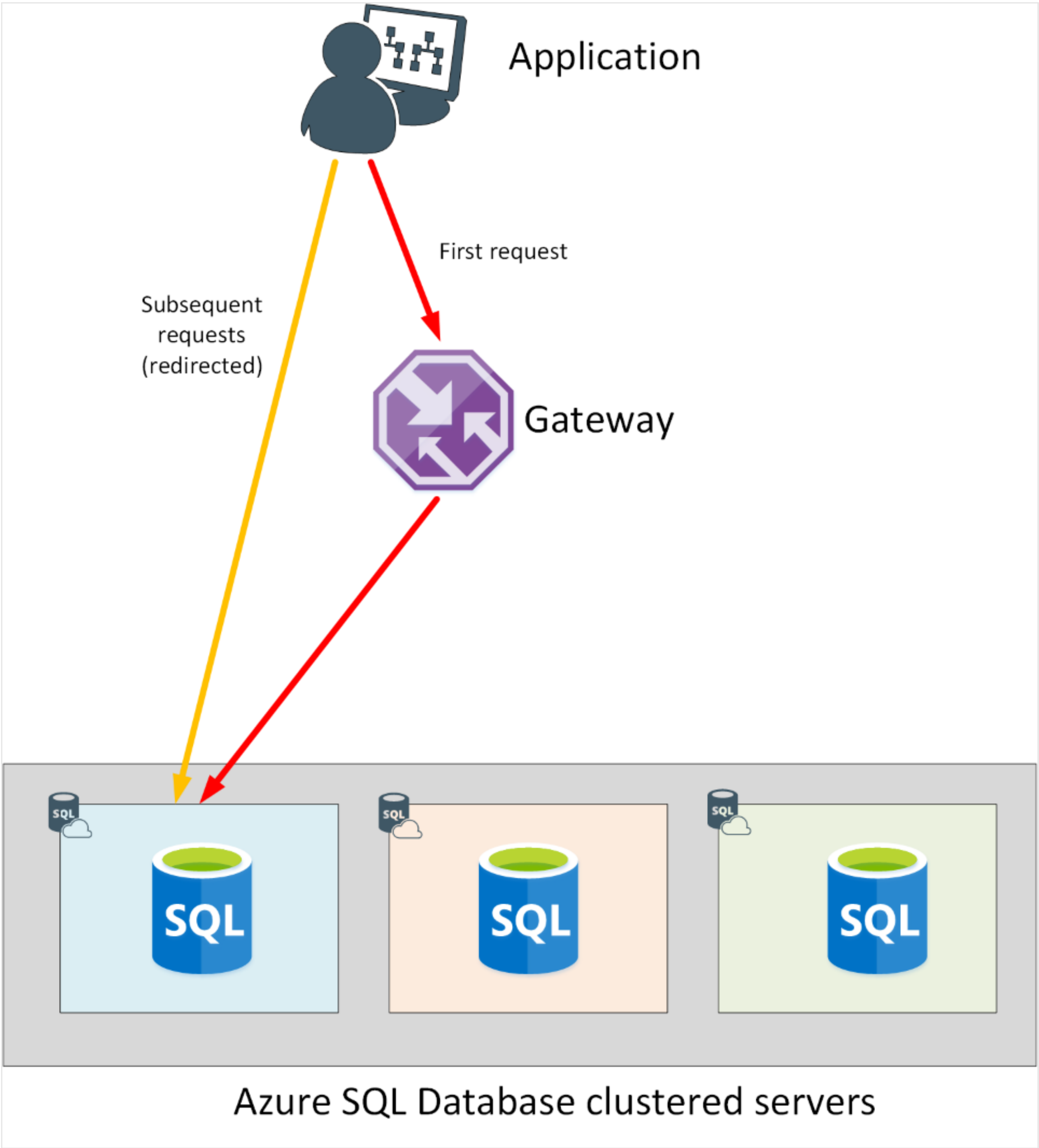
- Clients connect to a *gateway* that has a public IP address and listens on port 1433.
- Depending on the effective connection policy, the gateway either redirects traffic to the database cluster, or acts as a proxy for the database cluster.

  > ⓘ Note
  >
  > Azure SQL Database uses a clustered topology to provide high availability. Each server and database is transparently replicated to ensure that a server is always accessible, even in the event of a database or server failure.

- Inside the database cluster, traffic is forwarded to the appropriate Azure SQL database.

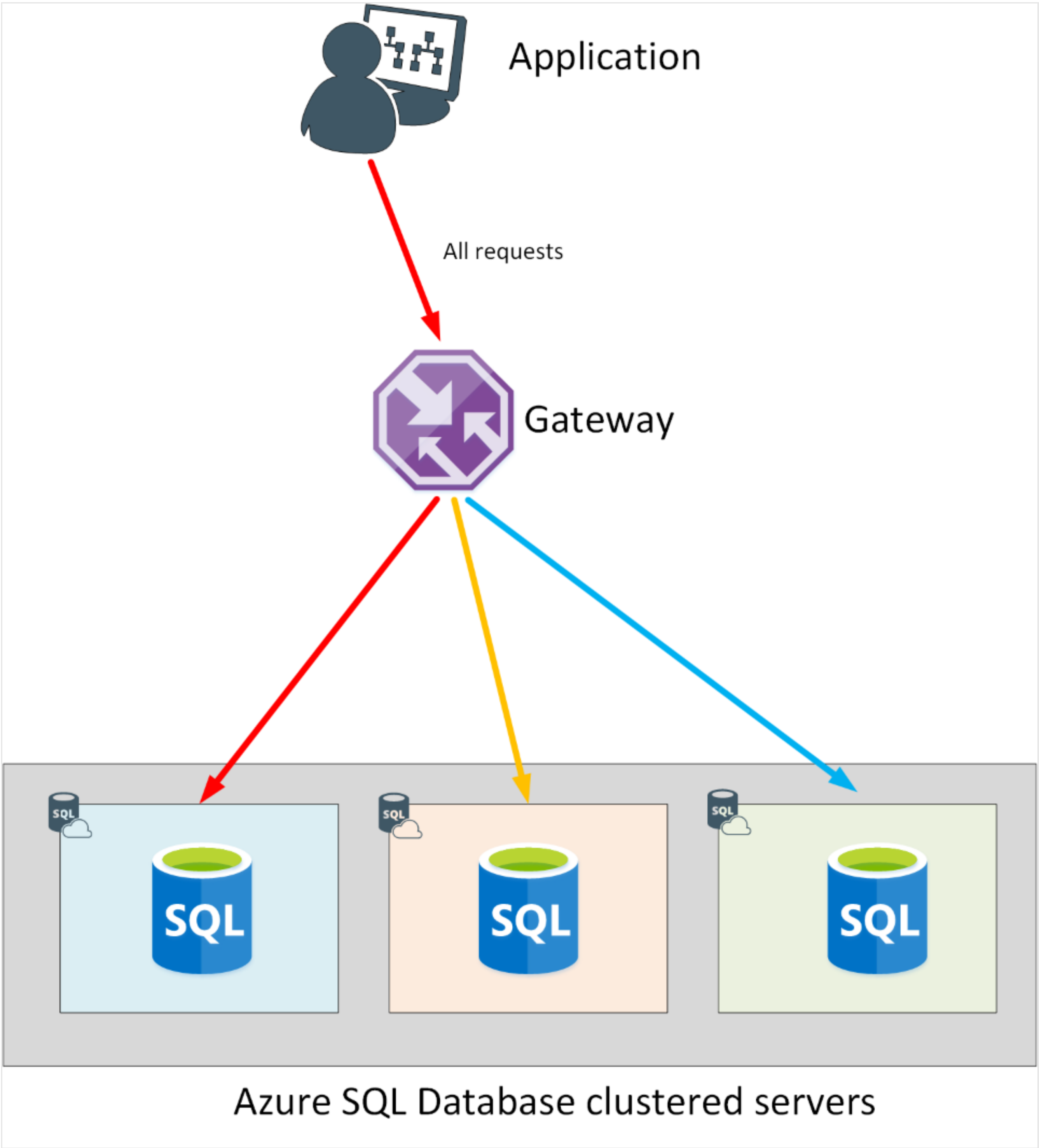
Connectivity from within Azure

If you're connecting from within another Azure service, such as a web application running under Azure App Service, your connections have a connection policy of Redirect by default. A policy of Redirect means that after your application establishes a connection to the Azure SQL database through the gateway, all following requests from your application will go directly to the database rather than through the gateway. If connectivity to the database subsequently fails, your application will have to reconnect through the gateway, when it might be directed to a different copy of the database running on another server in the cluster.


Application
First request
Subsequent requests (redirected)
Gateway
SQL
SQL
SQL
SQL
SQL
SQL
Azure SQL Database clustered servers

Connectivity from outside of Azure

If you're connecting from outside Azure, such as an on-premises application, your connections have a connection policy of Proxy by default. A policy of Proxy means the connection is established via the gateway, and all subsequent requests flow through the gateway. Each request could (potentially) be serviced by a different database in the cluster.



Configure DoSGuard

Denial of service (DoS) attacks are reduced by a SQL Database gateway service called DoSGuard. DoSGuard actively tracks failed logins from IP addresses. If there are multiple failed logins from a specific IP address within a period of time, the IP address is blocked from accessing any resources in the service for a short while.

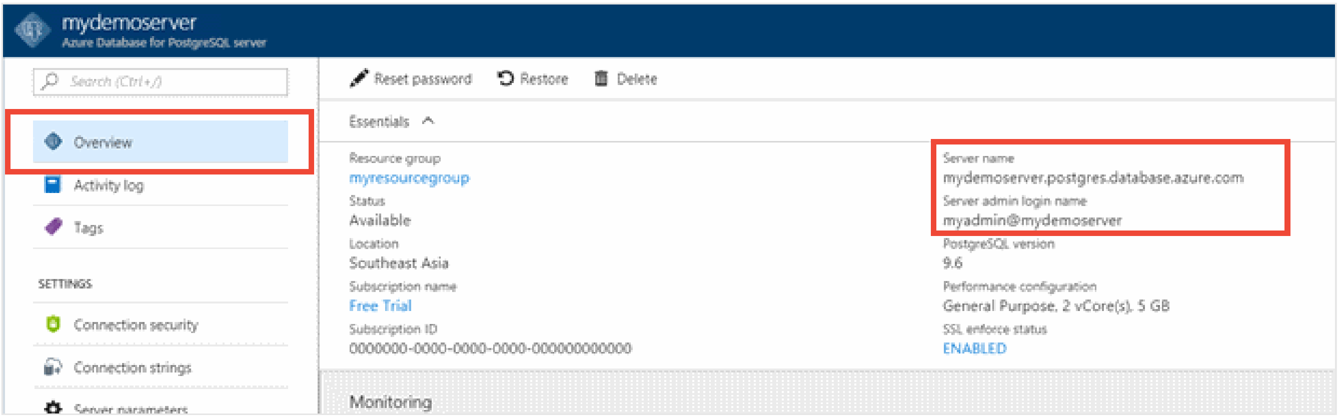
In addition, the Azure SQL Database gateway performs the following tasks:

- It validates all connections to the database servers, to ensure that they are from genuine clients.

- It encrypts all communications between a client and the database servers.
- It inspects each network packet sent over a client connection. The gateway validates the connection information in the packet, and forwards it to the appropriate physical server based on the database name that's specified in the connection string.

Configure Azure Database for PostgreSQL

When you create your Azure Database for PostgreSQL server, a default database named *postgres* is created. To connect to your database server, you need your full server name and admin sign-in credentials. You can easily find the server name and sign in information on the server **Overview** page in the portal. This page contains the Server name and the Server admin sign-in name.



Resource group
myresourcegroup

Status
Available

Location
Southeast Asia

Subscription name
Free Trial

Subscription ID
0000000-0000-0000-0000-000000000000

Server name
mydemoserver.postgres.database.azure.com

Server admin login name
myadmin@mydemoserver

PostgreSQL version
9.6

Performance configuration
General Purpose, 2 vCore(s), 5 GB

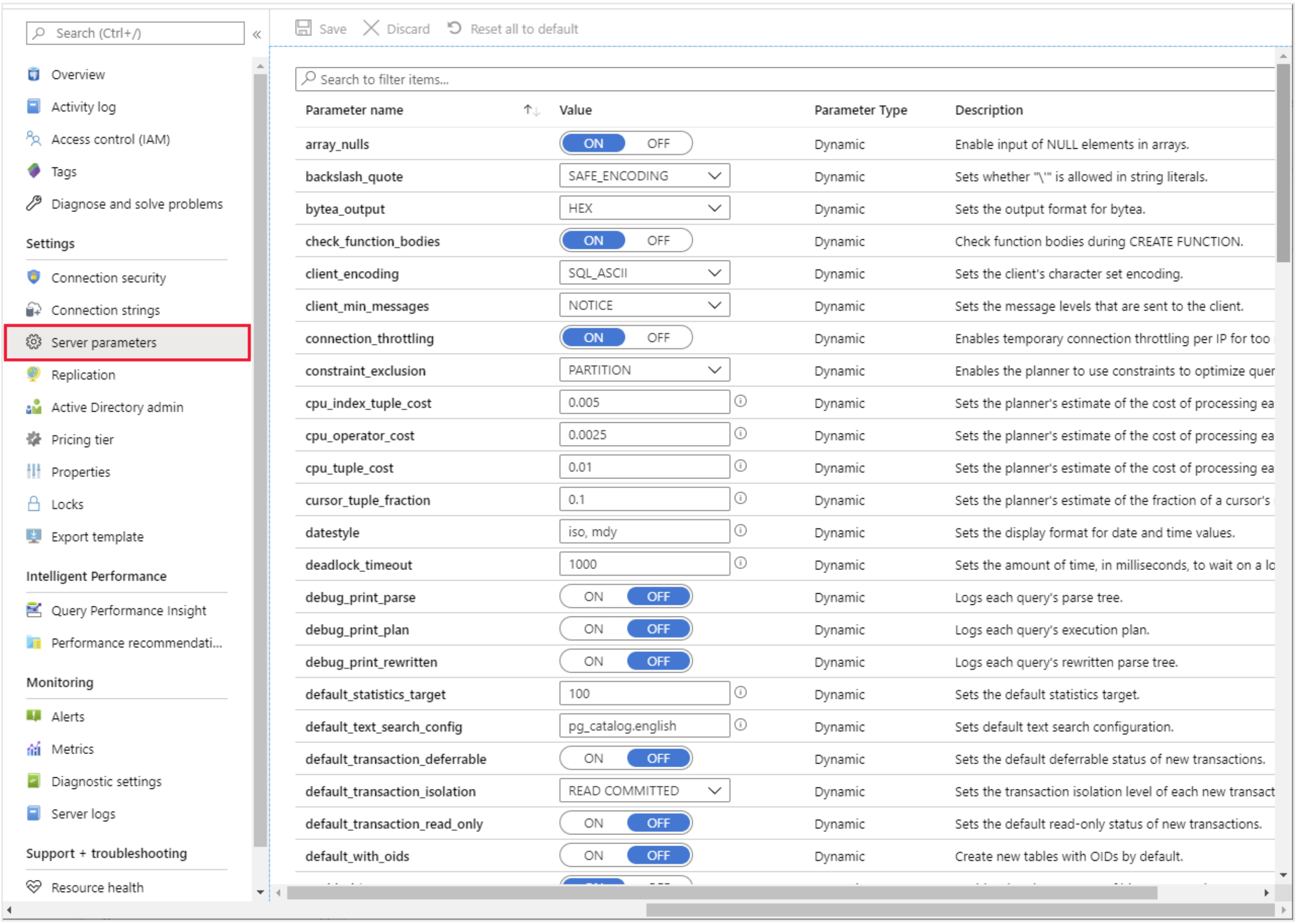
SSL enforce status
ENABLED

ⓘ Note

Connections to your Azure Database for PostgreSQL server communicate over port 5432. When you try to connect from within a corporate network, outbound traffic over port 5432 might not be allowed by your network's firewall. If so, you can't connect to your server unless your IT department opens port 5432.

Configure server parameters and extensions

A PostgreSQL database server has a number of configuration parameters that you can set. These parameters support fine-tuning of the database, and debugging of code in the database. You can modify these parameters using the **Server parameters** page in the Azure portal.



Parameter name	Value	Parameter Type	Description
array_nulls	[✓ ON] OFF	Dynamic	Enable input of NULL elements in arrays.
backslash_quote	SAFE_ENCODING	Dynamic	Sets whether "\'" is allowed in string literals.
bytea_output	HEX	Dynamic	Sets the output format for bytea.
check_function_bodies	[✓ ON] OFF	Dynamic	Check function bodies during CREATE FUNCTION.
client_encoding	SQL_ASCII	Dynamic	Sets the client's character set encoding.
client_min_messages	NOTICE	Dynamic	Sets the message levels that are sent to the client.
connection_throttling	[✓ ON] OFF	Dynamic	Enables temporary connection throttling per IP for too
constraint_exclusion	PARTITION	Dynamic	Enables the planner to use constraints to optimize quer
cpu_index_tuple_cost	0.005	Dynamic	Sets the planner's estimate of the cost of processing ea
cpu_operator_cost	0.0025	Dynamic	Sets the planner's estimate of the cost of processing ea
cpu_tuple_cost	0.01	Dynamic	Sets the planner's estimate of the cost of processing ea
cursor_tuple_fraction	0.1	Dynamic	Sets the planner's estimate of the fraction of a cursor's
datestyle	iso, mdy	Dynamic	Sets the display format for date and time values.
deadlock_timeout	1000	Dynamic	Sets the amount of time, in milliseconds, to wait on a lo
debug_print_parse	ON [✓ OFF]	Dynamic	Logs each query's parse tree.
debug_print_plan	ON [✓ OFF]	Dynamic	Logs each query's execution plan.
debug_print_rewritten	ON [✓ OFF]	Dynamic	Logs each query's rewritten parse tree.
default_statistics_target	100	Dynamic	Sets the default statistics target.
default_text_search_config	pg_catalog.english	Dynamic	Sets default text search configuration.
default_transaction_deferrable	ON [✓ OFF]	Dynamic	Sets the default deferrable status of new transactions.
default_transaction_isolation	READ COMMITTED	Dynamic	Sets the transaction isolation level of each new transact
default_transaction_read_only	ON [✓ OFF]	Dynamic	Sets the default read-only status of new transactions.
default_with_oids	ON [✓ OFF]	Dynamic	Create new tables with OIDs by default.

If you're familiar with PostgreSQL, you'll find that not all parameters are supported in Azure. The Server parameters page on the Microsoft website describes the PostgreSQL parameters that are available.

PostgreSQL also provides the ability to extend the functionality of your database using extensions. Extensions bundle multiple related SQL objects together in a single package that can be loaded or removed from your database with a single command. After being loaded in the database, extensions function like built-in features. You install an extension in your database before you can use it. To install a particular extension, run the *CREATE EXTENSION* command from psql tool to load the packaged objects into your database. Not all PostgreSQL extensions are supported in Azure. For a full list, read PostgreSQL extensions in Azure Database for PostgreSQL - Single Server.

Configure read replicas

You can replicate data from an Azure Database for PostgreSQL server to a read-only server. Azure Database for PostgreSQL supports replication from the master server to up to five replicas. Replicas are updated asynchronously with the PostgreSQL engine native replication technology.

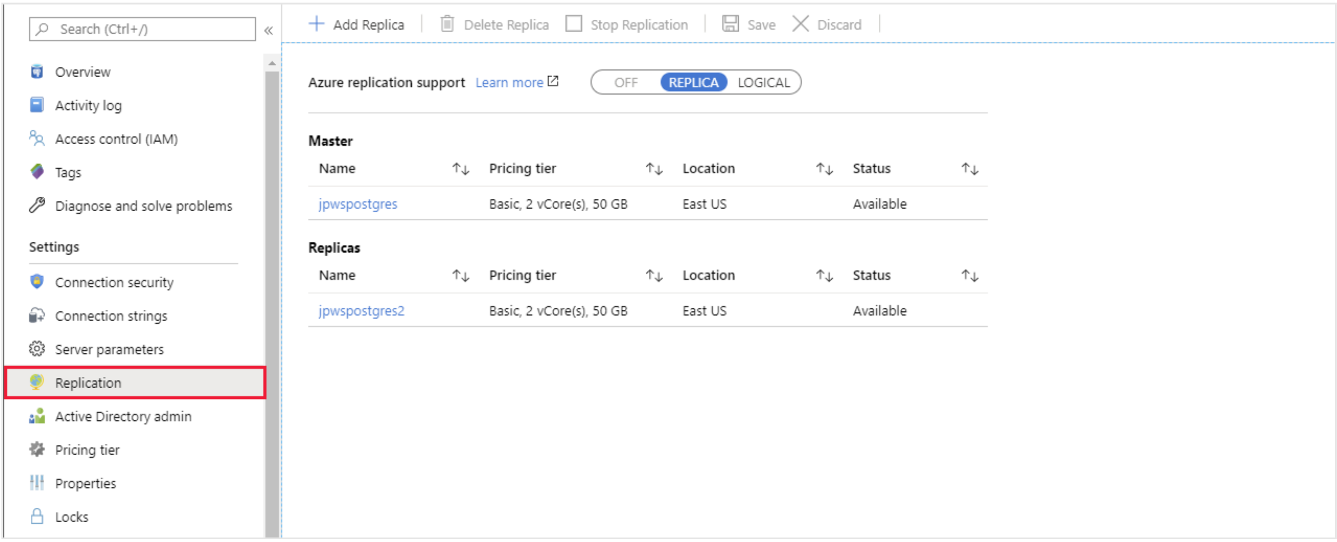
Read replicas help to improve the performance and scale of read-intensive workloads. Read workloads can be isolated to the replicas, while write workloads can be directed to the master.

A common scenario is to have BI and analytical workloads use read replicas as the data source for reporting.

Because replicas are read-only, they don't directly reduce the burden of write operations on the master. This feature isn't targeted at write-intensive workloads.

Replicas are new servers that you manage similar to regular Azure Database for PostgreSQL servers. For each read replica, you're billed for the provisioned compute in vCores and storage in GB/month.

Use the **Replication** page for a PostgreSQL server in the Azure portal to add read replicas to your database:



Azure replication support: REPLICA

Master

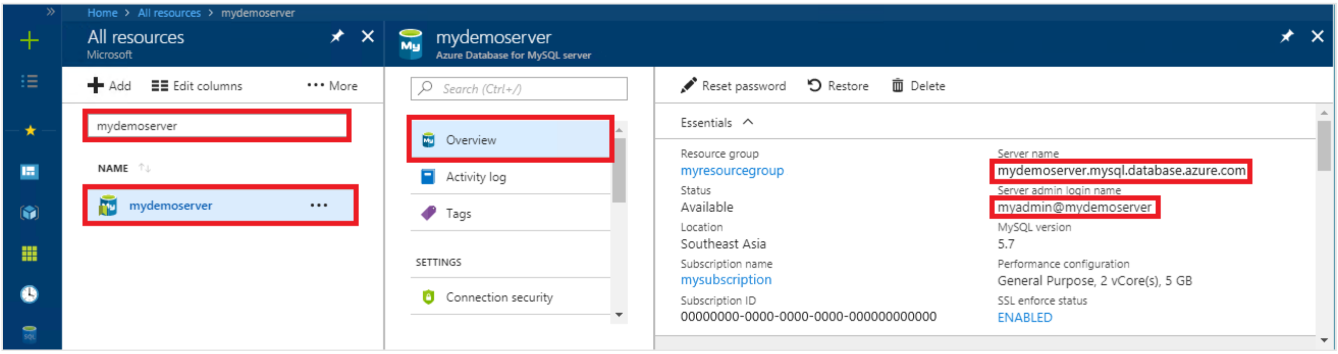
Name	Pricing tier	Location	Status
jpwspostgres	Basic, 2 vCore(s), 50 GB	East US	Available

Replicas

Name	Pricing tier	Location	Status
jpwspostgres2	Basic, 2 vCore(s), 50 GB	East US	Available

Configure Azure Database for MySQL

In order to connect to the MySQL database you've provisioned, you'll need to enter the connection information. This information includes fully qualified server name and sign-in credentials. You can find this information on the **Overview** page for your server:



Resource group
myresourcegroup
Status
Available
Location
Southeast Asia
Subscription name
mysubscription
Subscription ID
00000000-0000-0000-0000-000000000000

Server name
mydemoserver.mysql.database.azure.com
Server admin login name
myadmin@mydemoserver
MySQL version
5.7
Performance configuration
General Purpose, 2 vCore(s), 5 GB
SSL enforce status
ENABLED

ⓘ Note

Connections to your Azure Database for MySQL server communicate over port 3306. When you try to connect from within a corporate network, outbound traffic over port

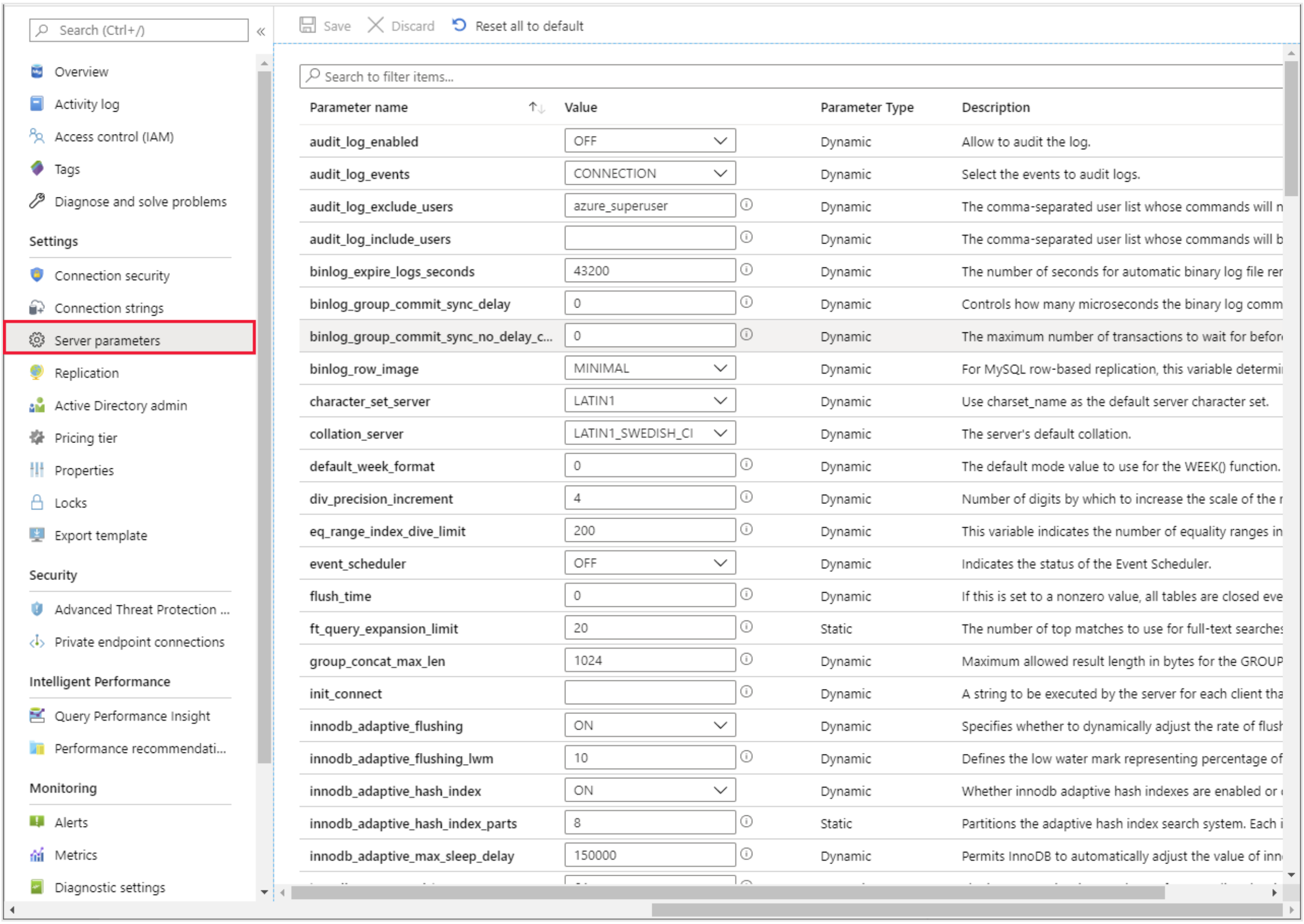
3306 might not be allowed by your network's firewall. If so, you can't connect to your server unless your IT department opens port 3306.

ⓘ **Important**

By default, SSL connection security is required and enforced on your Azure Database for MySQL server.

Configure server parameters

Like PostgreSQL, a MySQL database server has a number of configuration parameters that you can set. You can modify these parameters using the **Server parameters** page in the Azure portal.



Parameter name	Value	Parameter Type	Description
audit_log_enabled	OFF	Dynamic	Allow to audit the log.
audit_log_events	CONNECTION	Dynamic	Select the events to audit logs.
audit_log_exclude_users	azure_superuser	Dynamic	The comma-separated user list whose commands will n
audit_log_include_users		Dynamic	The comma-separated user list whose commands will b
binlog_expire_logs_seconds	43200	Dynamic	The number of seconds for automatic binary log file rer
binlog_group_commit_sync_delay	0	Dynamic	Controls how many microseconds the binary log comm
binlog_group_commit_sync_no_delay_c...	0	Dynamic	The maximum number of transactions to wait for befor
binlog_row_image	MINIMAL	Dynamic	For MySQL row-based replication, this variable determi
character_set_server	LATIN1	Dynamic	Use charset_name as the default server character set.
collation_server	LATIN1_SWEDISH_CI	Dynamic	The server's default collation.
default_week_format	0	Dynamic	The default mode value to use for the WEEK() function.
div_precision_increment	4	Dynamic	Number of digits by which to increase the scale of the r
eq_range_index_dive_limit	200	Dynamic	This variable indicates the number of equality ranges in
event_scheduler	OFF	Dynamic	Indicates the status of the Event Scheduler.
flush_time	0	Dynamic	If this is set to a nonzero value, all tables are closed eve
ft_query_expansion_limit	20	Static	The number of top matches to use for full-text searches
group_concat_max_len	1024	Dynamic	Maximum allowed result length in bytes for the GROUP
init_connect		Dynamic	A string to be executed by the server for each client tha
innodb_adaptive_flushing	ON	Dynamic	Specifies whether to dynamically adjust the rate of flush
innodb_adaptive_flushing_lwm	10	Dynamic	Defines the low water mark representing percentage of
innodb_adaptive_hash_index	ON	Dynamic	Whether innodb adaptive hash indexes are enabled or
innodb_adaptive_hash_index_parts	8	Static	Partitions the adaptive hash index search system. Each i
innodb_adaptive_max_sleep_delay	150000	Dynamic	Permits InnoDB to automatically adjust the value of inn

You can find more information about the parameters available for MySQL in Azure on the How to configure server parameters in Azure Database for MySQL by using the Azure portal page on the Microsoft website.

Configure read replicas

This feature is similar to that available for PostgreSQL. You can create up to five read replicas for a MySQL database. This feature enables you to geo-replicate data across regions and distribute the overhead associated with read-intensive workloads. Replication is asynchronous from the master server, so there may be some lag between records being written at the master and becoming available across all replicas.

Read replication isn't intended to support write-heavy workloads.

Use the **Replication** page for a MySQL server in the Azure portal to add read replicas to your database.

Next unit: Exercise: Provision Azure relational database services

Continue >