Microsoft

PostgreSQL Questions

Sridhar Kothalanka (CSA-Data&AI)

Jan 2020

Introduction

Overview

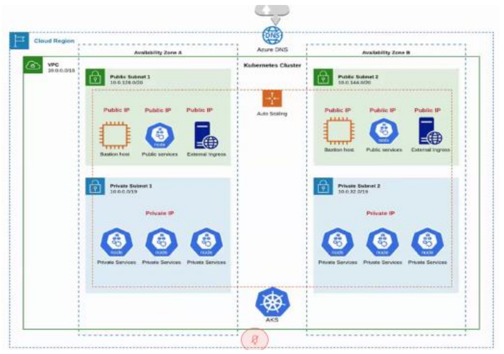
Customer is currently exploring Azure cloud services and evaluating if the below PaaS solutions meet their needs

- Kubernetes
- PostgreSQL
- Redis

Customer has used these open source products available on AWS and is technically aware of the products and features however would like to understand how these PaaS solutions are offered by Azure as a managed service.

High level use case

End devices will need to communicate in real time to a cloud infrastructure where a proprietary solution is deployed as a microservice in Kubernetes (Pls see backup slide).



Azure Database for PostgreSQL Questions

Pricing
What are the pricing options available?

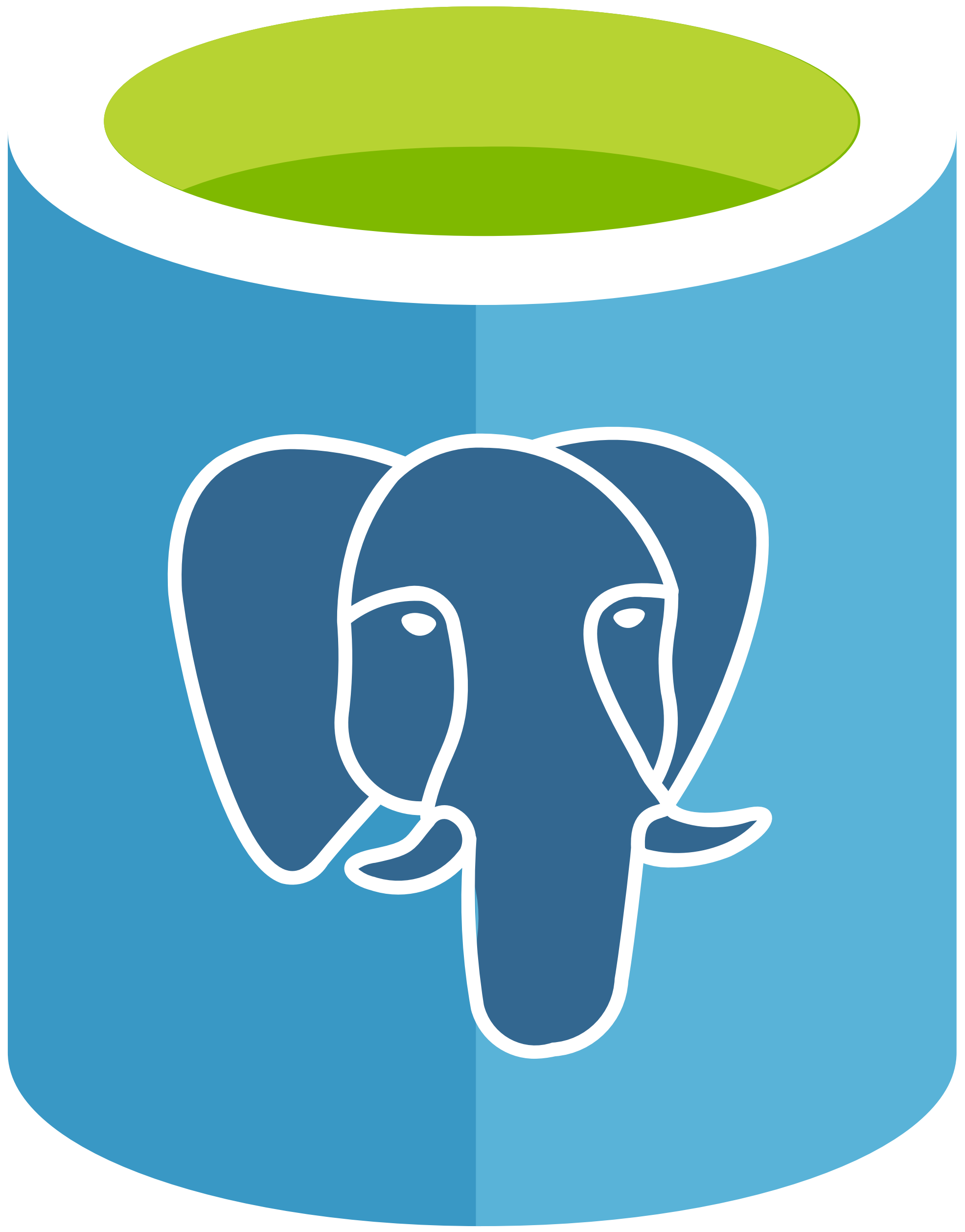
High Availability
High availability with no additional cost (99.99% SLA), need details?

Upgrades
How are upgrades handled as a managed service?

DB Schema Versioning
How is DB schema handled?

Optimization
How is optimization handled on the database?

Maintenance (Backup & Restore)
How is backup and restore handled?



Pricing

Deployment
☐Single Server
☐Hyperscale

+

Tiers
☐Basic
☐General Purpose
☑Memory Optimized

+

Options
❑Compute
❑Storage
❑Backup
❑High Availability
❑Read Replicas

=

TOTAL PRICE

Basic:
- Light compute & IO Performance
- Example: Dev/Test/Small scale infrequent applications

General Purpose:
- Most business workloads
- Example: Web/Mobile applications, Enterprise applications

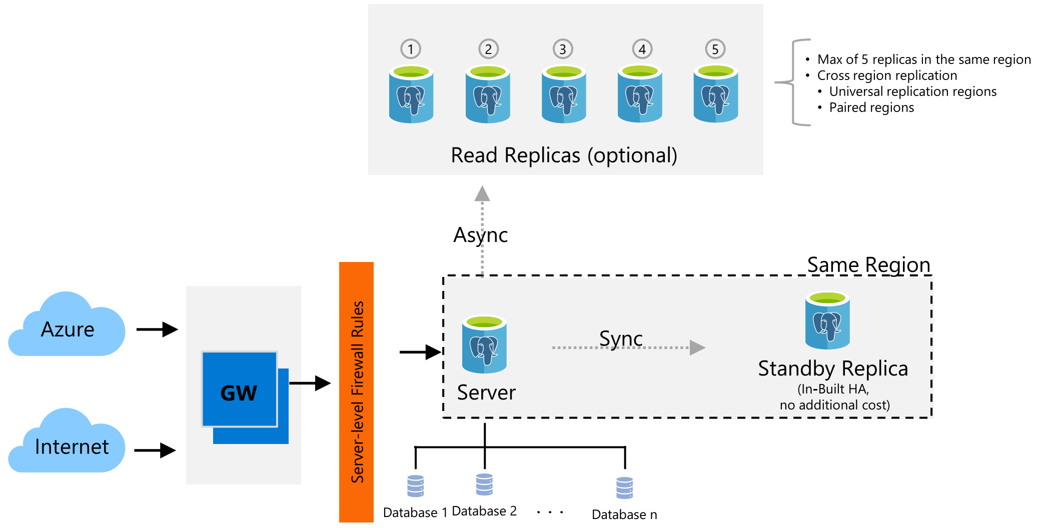
Memory Optimized:
- High performance database workloads
- Example: Realtime data, Transactional or Analytic apps

Recommendation:
☑ Memory Optimized

To get an estimate of the price please refer to below link
https://azure.microsoft.com/en-us/pricing/details/postgresql/server/

Deployment Options

Single Server



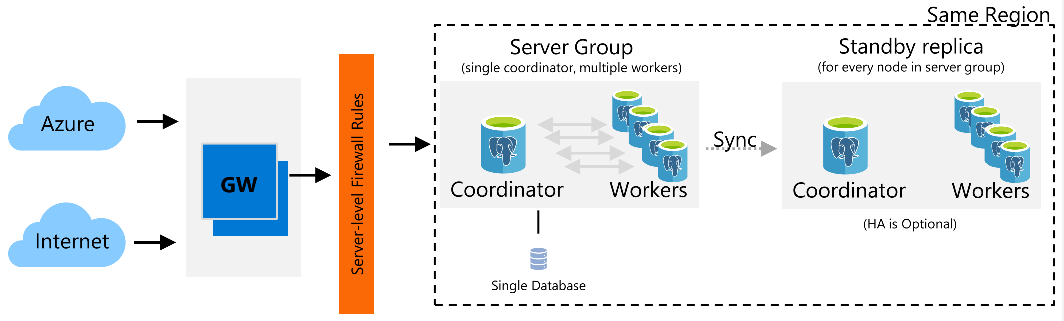
In a nutshell:

- Enterprise-grade security and compliance
- Built-in high availability, no additional cost
- Vertical scale as needed
- Secured to protect data at-rest and in-motion
- Automatic backups and point-in-time restore
- Monitoring and alerting to assess your server

Workloads:

- Transactional and operational analytics workloads
- Apps requiring JSON, geospatial support, or full-text search
- Cloud-native apps built with modern frameworks

Hyperscale



In a nutshell:

- Enterprise-grade security and compliance
- High availability (optional), cost will double if enabled
- Horizontal scaling across multiple machines using Sharding
- Query parallelization across the servers
- Excellent support for multi-tenant applications, Realtime operational analytics, and high throughput transaction workloads (>100GB of data)
- Monitoring and alerting to assess the nodes in your server group

Workloads:

- Multi-tenant SaaS applications
- Real-time operational analytics
- High-throughput transactional applications

High Availability

Single Server (Built-in)

Financially backed service level agreement (SLA) 99.99%

standby replicas of single node

Note: Built-In HA with no additional cost

Hyperscale (Optional)

MSFT documentation will be updated shortly

Financially backed service level agreement (SLA) 99.95%

standby replicas of every node in a server group

Note: Cost doubles the cluster price if HA is turned on

NOTE:

At all times, changes made to an Azure Database for PostgreSQL database server occur in the context of a transaction. Changes are recorded synchronously in Azure storage when the transaction is committed. If a node-level interruption occurs, the database server automatically creates a new node and attaches data storage to the new node. Any active connections are dropped, and any inflight transactions are not committed.

It is important that PostgreSQL database applications are built to detect and retry dropped connections and failed transactions.

Read Replica (☑Single Server, ☒Hyperscale)

When to use
Read replica 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

Highlights

- Up to 5 replicas
- Each replica is billed separately
- Uses PostgreSQL asynchronous replication
- Cross-region replication is supported and can help in scenarios like disaster recovery or bringing data closer to the consumers
  - Universal replica regions
  - Paired regions
- Replications can be monitored

Limitation

- Asynchronous replication, replica will eventually become consistent with the master but there will be a delay
- Paired regions may not be available in a region. Some regions are paired in one direction only
- If replication is stopped the replica becomes a standalone server. Standalone server cannot be made into a replica again
- There is no automated failover between master and replica servers

Upgrades

Microsoft aims to support n-2 versions of the PostgreSQL engine in Azure Database for PostgreSQL.

Current supported PostgreSQL versions: 11, 10, 9.6, 9.5

Minor Version

- Automatic minor version upgrades supported

Major Version

- Automatic major version upgrades not supported

DB Schema Versioning

- Not Supported
- PostgreSQL does not support DB schema versioning
- Need to use a 3rd party solution

Optimization

Query Store (opt-in feature)

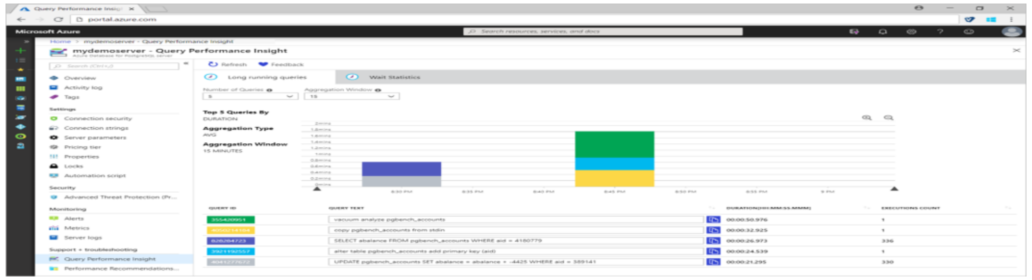
- Tracks query performance over time
- Automatically captures query history and runtime stats
- Database usage pattern

Query Performance Insight

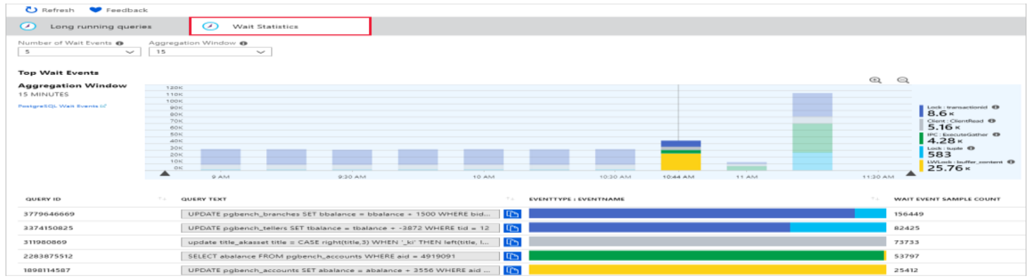
- Visualize long running queries
- Visualize wait statistics

Performance recommendation

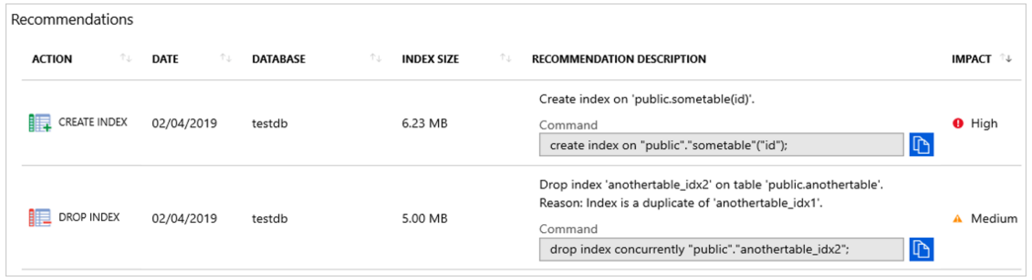
- Create index suggestions
- Drop index suggestions



QUERY ID	QUERY TEXT	DURATION(HH:MM:SS.MMM)	EXECUTIONS COUNT
355420951	vacuum analyze pgbench_accounts	00:00:50.976	1
4050214104	copy pgbench_accounts from stdin	00:00:32.925	1
828284723	SELECT abalance FROM pgbench_accounts WHERE aid = 4180779	00:00:26.973	336
3921192557	alter table pgbench_accounts add primary key (aid)	00:00:24.539	1
4041277672	UPDATE pgbench_accounts SET abalance = abalance + -4425 WHERE aid = 389141	00:00:21.295	330



QUERY ID	QUERY TEXT	EVENTTYPE : EVENTNAME	WAIT EVENT SAMPLE COUNT
3779646669	UPDATE pgbench_branches SET bbalance = bbalance + 1500 WHERE bid...		156449
3374150825	UPDATE pgbench_tellers SET tbalance = tbalance + -3872 WHERE tid = 12		82425
311980869	update title_akasset title = CASE right(title,3) WHEN '_ki' THEN left(title, l...		73733
2283875512	SELECT abalance FROM pgbench_accounts WHERE aid = 4919091		53797
1898114587	UPDATE pgbench_accounts SET abalance = abalance + 3556 WHERE aid ...		25412


Recommendations

ACTION	DATE	DATABASE	INDEX SIZE	RECOMMENDATION DESCRIPTION	IMPACT
CREATE INDEX	02/04/2019	testdb	6.23 MB	Create index on 'public.sometable(id)'. Command: create index on "public"."sometable"("id");	High
DROP INDEX	02/04/2019	testdb	5.00 MB	Drop index 'anothertable_idx2' on table 'public.anothertable'. Reason: Index is a duplicate of 'anothertable_idx1'. Command: drop index concurrently "public"."anothertable_idx2";	Medium

Monitoring

Azure Monitor

- Azure Monitor collects and aggregates data from a variety of sources into a common data platform where it can be used for analysis, visualization, and alerting.

Service Health

- View notifications
- Setup alerts
- View scheduled maintenance

Metrics

- The percentage of CPU in use
- The percentage of memory in use
- The percentage of IO in use
- The percentage of storage used out of the server's maximum
- The amount of storage in use
- The maximum storage limit for server
- The percentage of server log storage used
- The amount (Bytes) of server log storage used
- The maximum server log storage for server
- The number of active connections to the server
- The number of failed connections to the server
- Network ingress across active connections
- The amount of backup storage used
- The lag in bytes between the master and the most lagging replica (master server only)
- The time since the last replayed transaction (replica servers only)

Backup & Restore (☑Single Server, ☒Hyperscale)

Redundancy options

- Local redundancy (basic tier)
- Geo redundancy (local + paired region)

Retention period

- 7 days (default)
- Max 35 days (optional config)

Backup frequency

- Full backup – weekly
- Differential backup – twice a day (upto 4TB) *or* Snapshot backup – once a day (upto 16TB)
- Transaction log – every 5 minutes

Restore

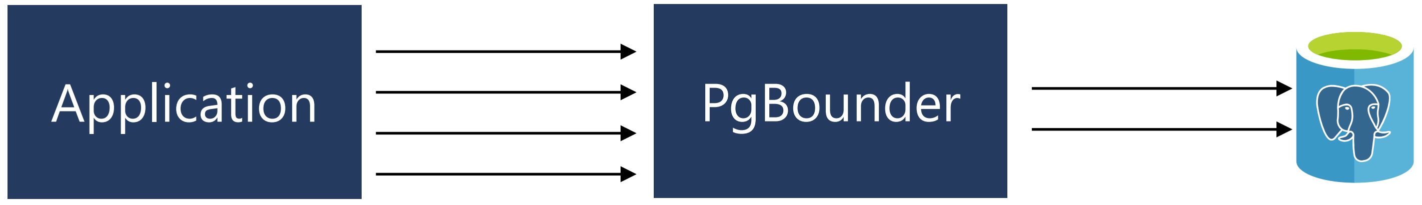
- Point-in-time restore (same region)
- Geo restore (only available if configured)

Note:

Configuring locally redundant or geo-redundant storage for backup is only allowed during server creation. Once the server is provisioned, you cannot change the backup storage redundancy option.

Azure Database for PostgreSQL provides up to 100% of your provisioned server storage as backup storage at no additional cost. Typically, this is suitable for a backup retention of seven days. Any additional backup storage used is charged in GB-month. For example, if you have provisioned a server with 250 GB, you have 250 GB of backup storage at no additional charge. Storage in excess of 250 GB is charged.

PgBouncer



Assumption: Application is containerized and running on Azure Kubernetes Service (AKS). An Azure Database for PostgreSQL server is created and blinded to the AKS cluster.

Highlights

- It is highly recommended to use a connection pool while running applications against Azure Database for PostgresSQL
- PgBouncer can run as a sidecar proxy, PgBouncer sidecar proxy image is available in Microsoft container registry
- PgBouncer, built-in retry logic further ensures connection resilience, high availability, and transparent application failover during the planned or unplanned failover of the database server

Advantages

- Improved throughput and performance
- No connection leaks by defining the maximum number of connections to the database server
- Improved connection resilience against restarts
- Reduced memory fragmentation

Best Practices

Single Server

Query performance tuning

- ➢ Enable *Query Store* (opt-in feature) this helps with identifying and tuning top expensive queries, identify and improve ad hoc workloads etc.,
- ➢ Use *Query Performance Insight* feature to help you to quickly identify what your longest running queries are, how they change over time, and what waits are affecting them
- ➢ Use *Performance Recommendations* feature to analyze workloads across your server to identify indexes with the potential to improve performance

Connectivity errors

- ➢ Transient connection errors should be handled using retry logic
- ➢ Connection *retry pattern*
  - Wait for 5 seconds before your first retry
  - For each following retry, increase the wait exponentially up to 60 seconds
  - Set a max number of retries at which point your application considers the operation failed

Monitoring

- ➢ Review the health of your apps with *Azure Monitor*. Use Azure Monitor to monitor metrics and configure alerts
- ➢ Enable server logs
- ➢ *Azure Service health* helps you stay informed and act when Azure service issues like outages and planned maintenance affects you. Setup alerts to be notified

Connection pooling

- ➢ Microsoft Container Registry provides a lightweight containerized *PgBouncer* that can be used in a sidecar to pool connections from AKS to Azure Database for PostgreSQL

Hyperscale

Multi-tenant apps

- ➢ Partition distributed tables by a common tenant_id column
- ➢ Convert small cross-tenant tables to reference tables
- ➢ Restrict filter all application queries by tenant_id

Real time Apps

- ➢ Choose a column with high cardinality as the distributed column (e.g., user, hosts or devices)
- ➢ Choose a column with even distribution
- ➢ Distribute fact and dimension tables on their common columns
- ➢ Change some dimension tables into reference tables

Time-series data

- ➢ Don’t choose a timestamp as the distributed column
- ➢ Use PostgreSQL table partitioning for time instead (break a large table of time-ordered data into multiple inherited tables with each table containing different time ranges

Questions

- PagerDuty Integration
  - https://www.pagerduty.com/docs/guides/azure-integration-guide/
- InfluxDB
  - Alternate, is to use TimescaleDB, as a PostgreSQL extension
- Grafana
  - Use Azure Monitor as a data source using a plugin

Microsoft

Q&A