Microsoft

Azure Database for PostgreSQL Flexible Server Resilient Architecture

WHITEPAPER V1.0

SAQLAIN TAHIR – PRINCIPAL CLOUD SOLUTION ARCHITECT

Table of Contents

All the components of the resilient architecture for Azure Database for PostgreSQL Flexible Server as per the given structure for Regional Outage, Zonal Outage, and Service Level Outage.

Regional Outage

Regional Disaster Protection

In the event of a region-wide disaster, Azure offers robust protection by leveraging disaster recovery mechanisms across different regions. This approach ensures resilience against regional or large-scale geographic disasters, providing a reliable fallback to another region. For a detailed understanding of Azure's disaster recovery architecture, refer to the Azure to Azure Disaster Recovery Architecture.

Business Continuity with Flexible Server

Azure Database for Postgres Flexible Server is designed to protect data and minimize downtime for mission-critical databases during both planned and unplanned events. Built on the resilient Azure infrastructure, flexible server incorporates features that ensure high availability and fault tolerance. These features are essential for maintaining business continuity and include fault-protection mechanisms, rapid recovery capabilities, and measures to minimize data loss.

When architecting applications, it is crucial to consider the following objectives to ensure business continuity:

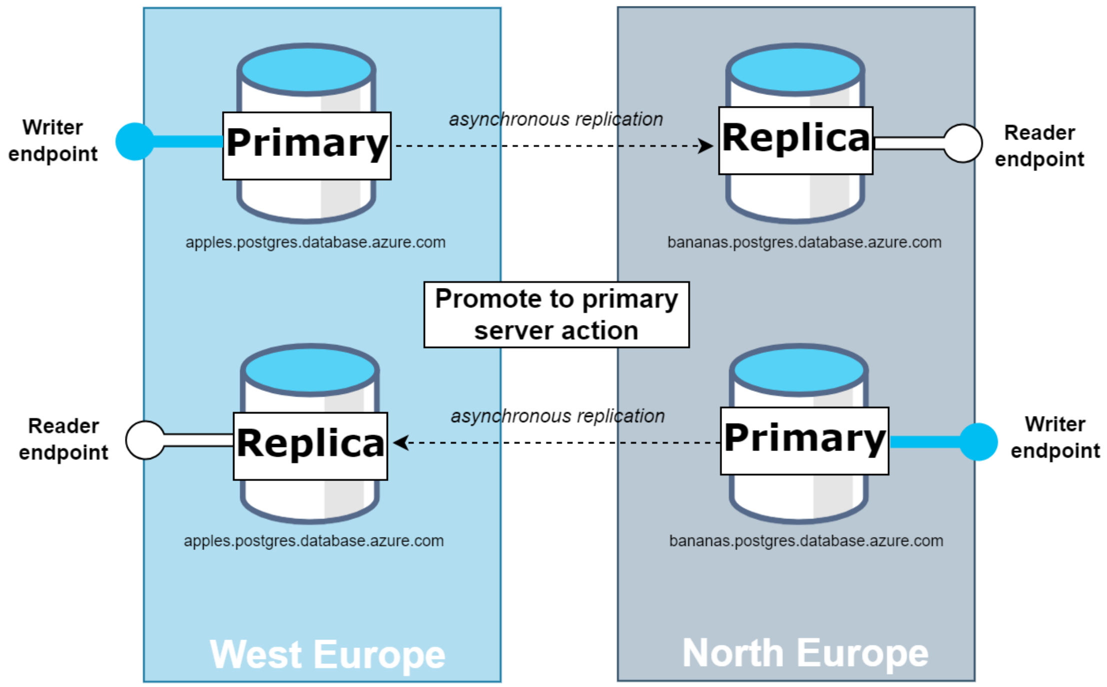
- Recovery Time Objective (RTO): This refers to the maximum acceptable amount of time an application can be offline. Different applications have varying tolerance levels for downtime; for instance, a business-critical database demands much stricter uptime compared to a test database.
- Recovery Point Objective (RPO): This indicates the maximum acceptable amount of data loss measured in time. It is vital to assess how much data loss your business can tolerate in case of a disruption.

Geo-Redundant Backup and Restore

Geo-redundant backup and restore allow you to restore your server in a different region in case of a disaster, providing 16 nines of durability for backup objects over a year. This serves as a cost-effective disaster recovery solution. Customers benefit from not having to pay for computing and disaster recovery until they initiate a restore. This feature must be configured at server creation, and it asynchronously copies backup data and transaction logs to the paired region.

Read Replicas

You can create replicas of the primary server within the same region or across different global Azure regions where Azure Database for PostgreSQL flexible server is available. In region replicas can provide replication options over default HA solutions. In region Read Replica can reduce read latency. Cross-region read replicas protect databases from region-level failures, using PostgreSQL's physical replication technology. They are available in general purpose and memory-optimized compute tiers and are updated asynchronously, which may result in some lag behind the primary server.


Writer endpoint
Primary
asynchronous replication
Replica
Reader endpoint
apples.postgres.database.azure.com
bananas.postgres.database.azure.com
Promote to primary server action
Reader endpoint
Replica
asynchronous replication
Primary
Writer endpoint
apples.postgres.database.azure.com
bananas.postgres.database.azure.com
West Europe
North Europe

Architecture Guidance

Action:

- In the event of a regional failure, you would need to manually promote a read replica in another region to be the new primary server.
- This process involves changing the read replica's role, which then takes over handling the write operations as the new primary server.

Implementation:

- Deploy instances across multiple Azure regions.
- Set up cross-region replication for disaster recovery.
- Implement geo-redundant configurations.

Impact:

- In Azure Database for PostgreSQL Flexible Server, when you make changes to keys or permissions on the primary server, these changes are typically replicated to any read replicas automatically. This replication includes changes to roles, permissions, and other security settings. Read replicas typically provide near-real-time updates from the primary server, but heavy, persistent write activities can lead to increased replication lag and higher storage usage on the primary due to retained WAL files

Mitigation:

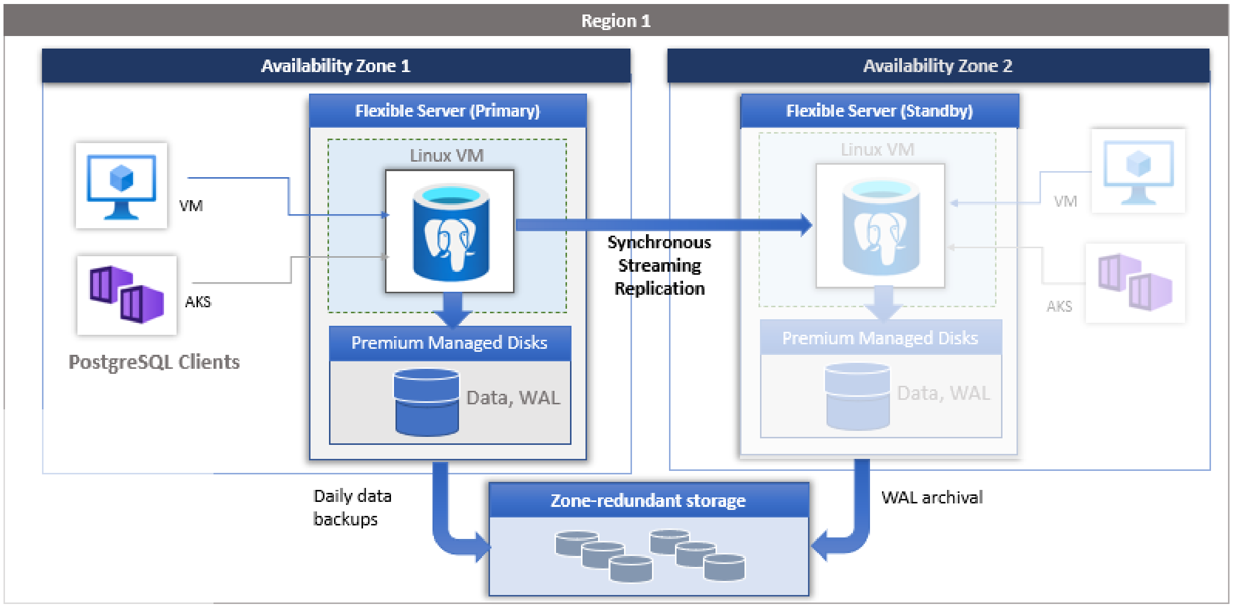
- Regularly test failover and disaster recovery plans.
- Ensure automated backups with configurable retention periods.
- Create Virtual Endpoints. These are read-write and read-only listener endpoints, that remain consistent irrespective of the current role of the Azure Database for PostgreSQL flexible server instance
- **Scenario 1:**
  - Using Virtual EndPoint with Read Replica
    - The read-only endpoint will point to the new replica which was the former primary
  - Using Virtual EndPoint without Read Replica
    - If we don't have read replica
- **Scenario 2:**
  - Use Virtual EndPoint in Parity with Read Replica to failover and failback between replicas
- With geo-redundant backup, you can perform a geo-restore in the paired region during an outage, creating a new server with the last available data. Cross-region read replicas can also be promoted to standalone read-write servers during regional failures, with an RPO of up to 5 minutes.

Zonal Outage

Azure Database for PostgreSQL - Flexible Server supports both zone-redundant and zonal models for high availability configurations. Both high availability configurations enable automatic failover capability with zero data loss during both planned and unplanned events.

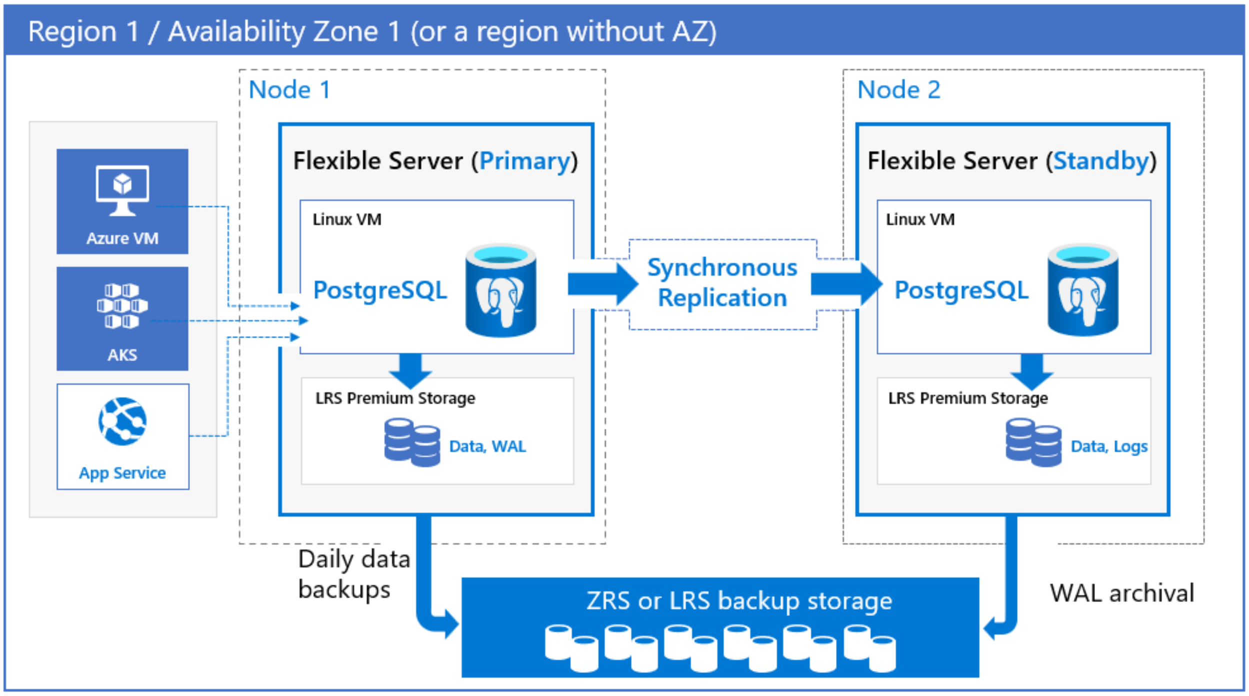
Zone-redundant

Zone redundant high availability deploys a standby replica in a different zone with automatic failover capability. Zone redundancy provides the highest level of availability but requires you to configure application redundancy across zones. For that reason, choose zone redundancy when you want protection from availability zone level failures and when latency across the availability zones is acceptable. Zone-redundancy model offers uptime SLA of 99.99%



Zonal

Choose a zonal deployment when you want to achieve the highest level of availability within a single availability zone, but with the lowest network latency. Zonal model offers uptime SLA of 99.95%



In zone-redundant and zonal models, automatic backups are performed periodically from the primary database server. At the same time, the transaction logs are continuously archived in the backup storage from the standby replica. If the region supports availability zones, backup data is stored on zone-redundant storage (ZRS). In regions that don't support availability zones, backup data is stored on local redundant storage (LRS)

Architecture Guidelines

Planned events such as scale computing and scale storage happen on the standby first and then on the primary server. Currently, the server doesn't failover for these planned operations.

Action:

- PostgreSQL Flexible Server will failover to another node within the same region if the zone hosting the primary instance is impacted.

Implementation:

- Utilize zone-redundant configurations.
- Configure automatic failover within the same region.
- Automatically creates a Standby.
- Flexible server health monitoring periodically checks for both the primary and standby health. If health monitoring detects that a primary server isn't reachable, the service then initiates an automatic failover to the standby server.

Impact*:*

- This setup does not affect networking/DNS, as high availability is managed through automated service healing.
- The health monitoring algorithm is based on multiple data points to avoid false positive situations.

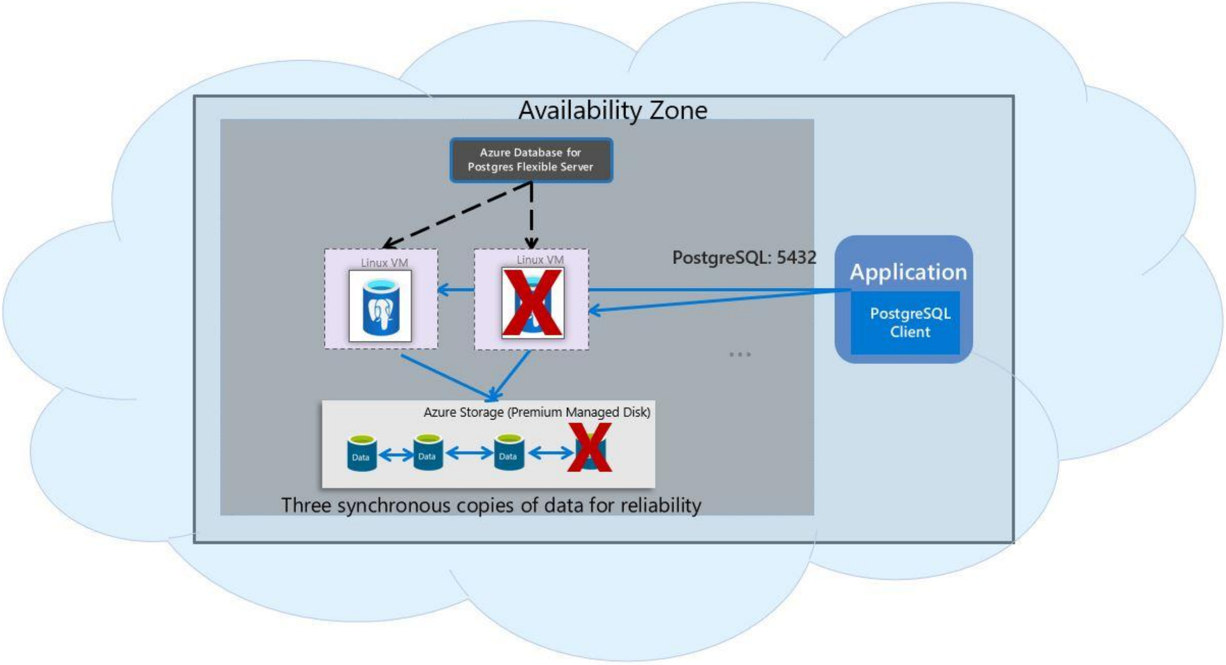
Mitigation:

- To recover from a zone-level failure, you can perform point-in-time restore using the backup. You can choose a custom restore point with the latest time to restore the latest data.
- Point-in-restore is recommended for user errors on the primary server like accidental drop of a table or incorrect data updates, from the backup. A new database server is restored as a single-zone flexible server with a new user-provided server name. Possible use cases could be:
  - Use restored server for production
  - Restore an object
  - Clone your database server to testing and development
- Implement monitoring and alerts to ensure timely detection and response.
- Use Virtual Network (VNet) integration or Private Link for secure and seamless failover.

VM Level Outage

Resilience without Availability Zone

For flexible servers configured without high availability, the service provides local redundant storage with three copies of data, zone-redundant backup (in regions where it's supported), and built-in server resiliency to automatically restart a crashed server and relocate the server to another physical node. Uptime SLA of 99.9% is offered in this configuration


Availability Zone
Azure Database for Postgres Flexible Server
Linux VM
Linux VM
PostgreSQL: 5432
Application
PostgreSQL Client
...
Azure Storage (Premium Managed Disk)
Data
Data
Data
Three synchronous copies of data for reliability

Service Level Outage

Azure Database for PostgreSQL Flexible Server offers various levels of resilience, but it's still important to understand the potential service level outages that might affect its availability and performance. Here are some possible service level outages for Azure Database for PostgreSQL Flexible Server resilience:

- **Infrastructure Outages**
  - **Hardware Failures**: Failures in the underlying hardware components such as CPUs, memory, disks, or network interfaces.

   - **Data Center Outages**: Entire data center outages due to power failures, natural disasters, or other catastrophic events.

- **Network Outages**
   - **Network Connectivity Issues**: Loss of network connectivity between the client and the server, or within the Azure data centers.
   - **DNS Failures**: Issues with DNS resolution that prevent access to the database.

- **Service Outages**
   - **Service Maintenance**: To mitigate the impact of maintenance-related downtime, we offer a couple of options:
      - **Leverage Flexible Maintenance Windows**: This allows you to choose a maintenance window that best fits your business operations and minimizes disruption.
      - **Choose Between 'System-Defined Schedule' and 'Custom-Defined Schedule'**:
         - A **System-Defined Schedule** is predetermined by our systems based on overall usage patterns and is designed to impact the fewest customers.
         - A **Custom-Defined Schedule** gives you the flexibility to set specific times for maintenance, providing more control over when these activities occur.
   - **Service Interruptions**: Unexpected interruptions in the Azure PostgreSQL service due to bugs, misconfigurations, or other operational issues.

- **Resource Limitations**
   - **Resource Exhaustion**: Running out of allocated resources such as CPU, memory, or disk I/O, which can cause the database to become unresponsive or slow.
   - **Quota Limits**: Hitting quota limits set by Azure for the number of resources that can be provisioned.

- **Software Failures**
   - **Database Engine Bugs**: Bugs in the PostgreSQL engine itself that cause crashes or data corruption.
   - **Operating System Issues**: Problems with the underlying operating system that affect the database performance or availability.

- **Configuration Errors**
  - **Misconfiguration**: Incorrect configuration settings that lead to suboptimal performance or security vulnerabilities.
  - **Deployment Errors**: Errors during the deployment process that prevent the server from starting or functioning correctly.
- **Security Incidents**
  - **Denial of Service (DoS) Attacks**: Malicious attacks aimed at making the service unavailable by overwhelming it with requests.
  - **Security Breaches**: Unauthorized access to the database, leading to potential data loss or corruption.
- **Geographical Outages**
  - **Regional Outages**: Outages that affect an entire Azure region, impacting all services within that region.
  - **Availability Zone Failures**: Failures within a specific availability zone that do not affect the entire region but impact services in that zone.

A detailed look at how failover is managed and the considerations involving Azure Key Vault (AKV) with private endpoints:

Within-Region Failover

1. **High Availability Configuration**: Azure PostgreSQL Flex allows you to configure a standby replica of your primary server within the same region but in a different availability zone. This setup ensures that in the event of a zone failure, the standby can automatically take over with minimal downtime.
2. **Automatic Failover**: The service automatically handles failover to the standby server if the primary server becomes unavailable due to maintenance or outages.
3. **DNS Caching**: When using private endpoints, DNS caching can affect how quickly applications can reconnect to the database post-failover. It’s crucial to configure your DNS settings properly to handle the updated pointers that direct to the new primary server.

Cross-Region Failover

1. **Geo-Redundancy**: Cross-region failover can be facilitated by setting up a geo-redundant standby server in a different region. This is particularly important for disaster recovery scenarios where regional continuity is at risk.
2. **Manual Failover**: Typically, cross-region failover is not automatic and requires manual intervention to promote a standby server in another region to become the new primary server.
3. **DNS and Endpoint Configuration**:
    a. Using AKV with private endpoints complicates cross-region failover due to the need for proper DNS management. Since AKV uses regional DNS zones, you need to ensure that your DNS setup can handle the change in regions without significant delays.
    b. Consider implementing traffic manager profiles or using Azure DNS to manage DNS failover and health checks, ensuring that the DNS can quickly resolve to the new primary server's location.

Integration with Azure Key Vault

When integrating Azure PostgreSQL Flex with AKV for managing encryption keys and secrets, consider the following:

1. **AKV Resilience**: Rely on AKV's built-in high availability and redundancy features to ensure that it remains accessible even during regional outages.
2. **DNS Requirements**: For AKV, especially when accessed via private endpoints, ensure that your DNS setup aligns with AKV's failover mechanism to maintain connectivity during and after failovers. This might involve configuring DNS settings to support rapid changes in endpoint addresses.
3. **Cross-Region Considerations**: If your DR strategy includes cross-region failover, ensure that your AKV setup can replicate secrets and keys across regions or that you have equivalent key vaults in multiple regions with synchronized data.

By addressing these considerations, you can enhance the resilience and availability of your Azure Database for PostgreSQL Flexible Server deployments, ensuring they remain robust even in complex scenarios involving regional outages and failovers.

Architecture Guidelines

By understanding these potential outages and employing the appropriate resilience strategies, you can ensure a more robust and reliable Azure Database for PostgreSQL Flexible Server deployment.

Action:

- In case of a primary region network outage, failover Azure Database for PostgreSQL Flexible Server to the secondary region.

Implementation:

- Deploy instances across multiple regions with a Traffic Manager for routing.
- Configure DNS to support failover scenarios.
- Azure Private DNS offers a reliable and secure DNS service specifically for your virtual network, enabling domain name resolution within the network without requiring a custom DNS configuration.
- When using private network access within an Azure virtual network, providing Private DNS zone information is essential for DNS resolution. For new Azure Database for PostgreSQL flexible servers created with private network access, Private DNS zones must be configured to enable private access to these servers.
- When using a private DNS zone in a different subscription, that subscription **must** have the Microsoft.DBforPostgreSQL resource provider registered as well, otherwise your deployment of Azure Database for PostgreSQL flexible server won't complete.

Impact:

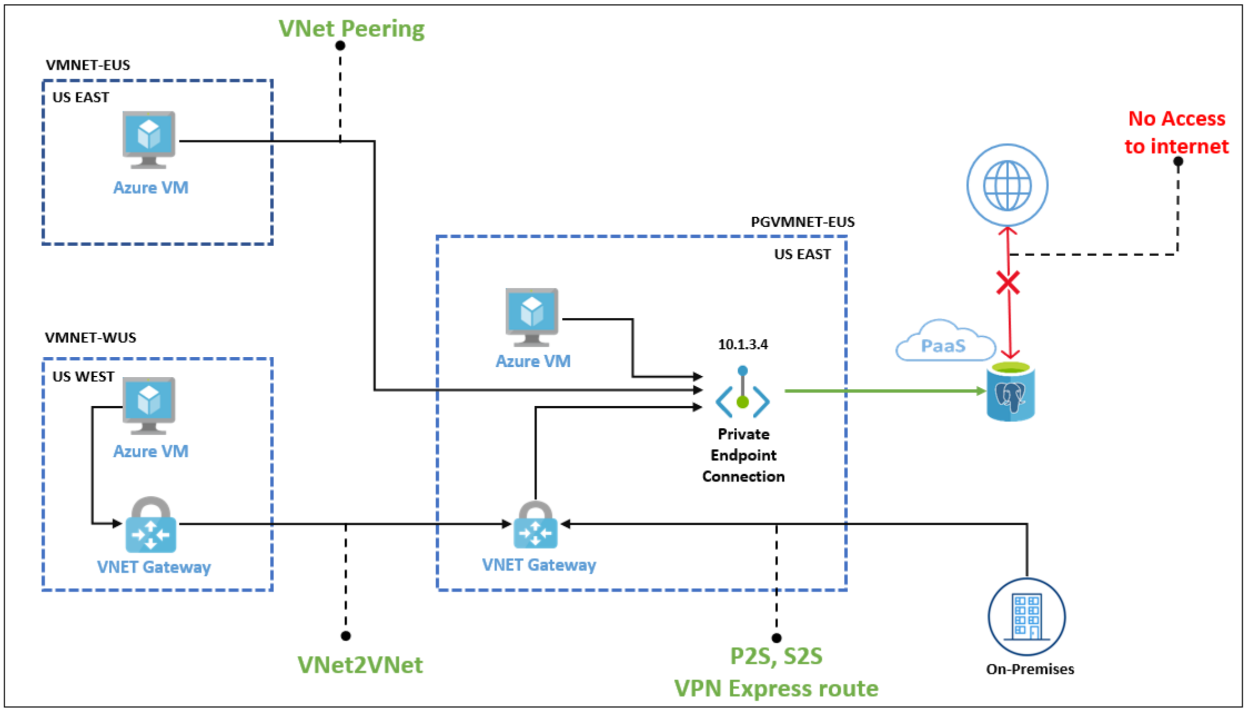
- The traffic manager will route requests to the operational pool, ensuring full functionality.

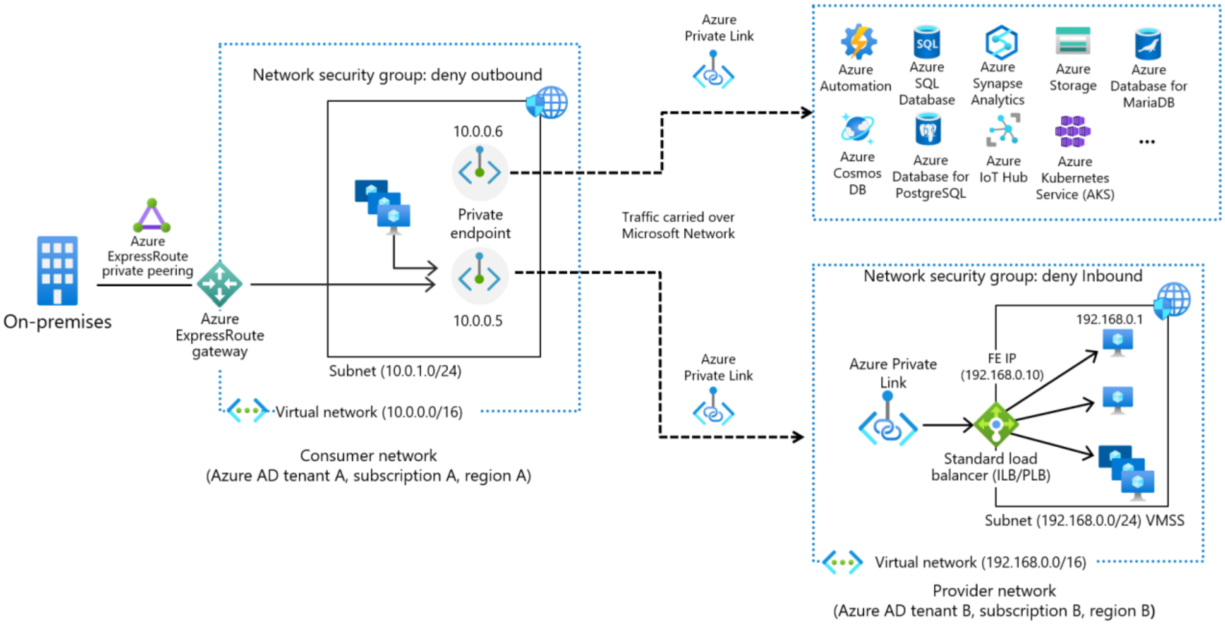
Mitigation:

- Use Azure Traffic Manager to route requests to healthy endpoints.
- Implement retry logic and circuit breaker pattern in the application layer.
- High Availability: Configurations that support automatic failover within the same region or across multiple regions.
- Geo-Redundancy: Creating read replicas or backups in different geographic regions to protect against regional outages.
- Automated Backups: Regular automated backups to ensure data can be restored in case of corruption or loss.

- Monitoring and Alerts: Using Azure Monitor and setting up alerts to quickly respond to issues.
- Scaling Resources: Dynamically scaling resources to handle increased load and avoid resource exhaustion.
- Security Best Practices: Implementing security measures such as firewalls, encryption, and regular updates to protect against security incidents.

Private Link Support for Azure PostgreSQL Flex:




Azure Private Link
Azure Automation
Azure SQL Database
Azure Synapse Analytics
Azure Storage
Azure Database for MariaDB
Azure Cosmos DB
Azure Database for PostgreSQL
Azure IoT Hub
Azure Kubernetes Service (AKS)
...
Network security group: deny outbound
10.0.0.6
Private endpoint
10.0.0.5
Subnet (10.0.1.0/24)
Virtual network (10.0.0.0/16)
Consumer network
(Azure AD tenant A, subscription A, region A)
Azure ExpressRoute private peering
Azure ExpressRoute gateway
On-premises
Traffic carried over Microsoft Network
Azure Private Link
Network security group: deny Inbound
192.168.0.1
Azure Private Link
FE IP (192.168.0.10)
Standard load balancer (ILB/PLB)
Subnet (192.168.0.0/24) VMSS
Virtual network (192.168.0.0/16)
Provider network
(Azure AD tenant B, subscription B, region B)

Resilience Architecture Best Practices

Levels of Resiliency Designs

- Encryption: Data at rest and in transit using AES-256 encryption.
- Hardware Security Module (HSM): Use Azure Key Vault with HSM for managing keys.
- Redundancy: Zone-redundant and geo-redundant configurations.
- Automated Backups: Regular automated backups with configurable retention.
- Point-in-Time Restore (PITR): Restore data to any point within the retention period.

List of Gaps/Risks

- Single Region Failure: Even with zone-redundant setup, a complete region failure can impact availability.
- Latency: Cross-region replication can introduce latency.
- Cost: Higher costs associated with redundant setups and advanced security measures.
- Complexity: Increased complexity in managing and maintaining the architecture.

Cost Estimates for Each Design

- Single Region Setup: $
- Zone-Redundant Setup: $$
- Geo-Redundant Setup: $$$
- Enhanced Security with HSM: $$

Resiliency Ranking

- Geo-Redundant Setup: 1
- Zone-Redundant Setup: 2
- Single Region Setup: 3
- Enhanced Security with HSM: 4
- Standard Security: 5

Developer Guidance

- Application Layer Recommendations:
- Implement retry logic for transient failures.
- Use connection pooling to manage database connections efficiently.
- Implement exponential backoff strategies.
- Retry Logic: Built-in retry mechanisms for common errors.
- Circuit Breaker Pattern: To prevent cascading failures.

DNS Considerations

- Customer DNS: Flexibility in naming conventions and existing infrastructure integration.
- Private DNS (Azure): Enhanced security, reduced latency, and easier management within Azure ecosystem.

Security Hardening Best Practices:

1. **Physical Security**: PostgreSQL Flexible Server benefits from the robust physical security of the Microsoft Azure cloud platform. Azure datacenters are designed, built, and operated to strictly control physical access to areas where your data is stored.

2. **Network Security**: Deploy your flexible server into an Azure virtual network (VNet). Each VNet is isolated from others and the internet, ensuring that your network traffic is not accessible to other Azure customers or external entities. Use Network Security Groups (NSGs) for basic network-level access control based on IP addresses and protocols. Azure Virtual Endpoints can also create a facade layer, protecting the primary server from direct exposure. It is also recommended to use Azure Private Link for creating a private endpoint. It can also enhance security by ensuring that the database is not accessible via the public internet, only through the VNet.

3. **Encryption in Transit**: Encrypt client traffic to your PostgreSQL Flexible Server as it traverses the network. Use TLS, an industry-standard protocol, to ensure encrypted connections between your database server and clients. Azure Database for PostgreSQL - Flexible Server supports TLS version 1.2 and later. The Handshake Protocol manages cipher suite negotiation, server and optional client authentication, and session key exchange.

4. **Authentication**: Enhance authentication security by using Microsoft Entra ID, supported exclusively on the Azure platform. Benefits include:

a. Uniform user authentication across Azure services.
b. Centralized management of password policies and rotation.
c. Management of database permissions using external (Entra ID) groups.
d. Use of PostgreSQL database roles for identity authentication at the database level.
e. Support for token-based authentication for applications connecting to Azure Database for PostgreSQL.

5. **Limiting Access with Roles**: Grant permissions directly to database users by creating roles with specific sets of permissions based on minimum application and access requirements. Assign appropriate roles to each user to enforce a least privilege model for accessing database objects. Grant specific privileges to database users on a granular basis.

6. **Encryption at Rest with Customer Managed Keys (CMK)**: Data on disk is encrypted with an Azure internal key known as the Data Encryption Key (DEK). Secure key creation, storage, access control, and management are essential. Azure Key Vault (AKV) is the recommended key storage solution, providing a common management experience across services. Keys are stored and managed in AKV, and access can be granted to users or services. Azure Key Vault supports customer creation or import of keys for customer-managed encryption scenarios. Permissions to use keys stored in AKV for encryption and decryption can be granted to Azure Active Directory accounts.

References:

- Reference Architecture
- Security in Azure Database for PostgreSQL - Flexible Server | Microsoft Learn
- Data encryption with customer-managed key - Azure Database for PostgreSQL - Flexible server | Micros...
- Azure Data Encryption-at-Rest - Azure Security | Microsoft Learn
- Active Directory authentication - Azure Database for PostgreSQL - Flexible Server | Microsoft Learn
- Networking overview - Azure Database for PostgreSQL - Flexible Server | Microsoft Learn
- https://learn.microsoft.com/en-us/azure/postgresql/flexible-server/concepts-networking-private-link
- Data security and encryption best practices - Microsoft Azure | Microsoft Learn
- Virtual endpoints - Azure Database for PostgreSQL - Flexible Server | Microsoft Learn