Security

Best Practices

1. Hash all passwords
2. If possible, use MFA
3. Create regular backups of the database
4. Separate the database servers
5. Use firewalls
6. Use least-level of privilege needed for access
7. Ensure both data at-rest and data in-transit are encrypted

Implementation – follows Azure Security baseline

Data Encryption - Automatic

- Azure Database for MySQL secures your data by encrypting data in-transit with TLS
- Azure Database for MySQL uses the FIPS 140-2 validated cryptographic module for storage encryption of data at-rest.
- All data, including backups, are encrypted using an AES 256-bit algorithm

Network Security

- Ensure SSL connection is enabled

Diagnostic Settings, Audit Logging, Server Logs & Ingest Logs & Azure Activity Log

- What it is: monitors network resource configurations and detects changes for network resources related to your Azure Database
- What they do
  - Configure central security log management
  - Activity logs, which are automatically available, include event source, data, user, timestamp, source addresses, destination addresses, and other useful elements
  - Aggregates security data generated by your Azure Database for MySQL instances
  - Monitors attempts to access deactivated credentials
  - Creates alerts for when changes take place to production instances of Azure Database for MySQL
  - Detects changes for network resources
- Topics Covered
  - Network Security
  - Logging and Monitoring
  - Data Protection
  - Identity and Access Control
  - Threat protection

More information: https://docs.microsoft.com/en-us/azure/azure-monitor/essentials/activity-log

Data Recovery

- Ensure regular automated backups
  - You can use point-in-time restore to recover a server to an earlier state, as far back as 35 days

Database Team Members Responsibility

Identity and Access Control

- Maintain an inventory of administrative accounts
- Regularly review and reconcile user access

Data Recovery

- Ensure regular automated backups
  - You can use point-in-time restore to recover a server to an earlier state, as far back as 35 days
  - You can use musqldump to copy a database
  - We should backup our keys (through key vault) as well
  - General Things to Know
    - All backups are encrypted using AES 256-bit encryption
    - Server backups are automatically created and locally stored but they cannot be exported
    - For our subscription, a full database snapshot is performed daily & transaction log backups occur every 5 minutes
- https://docs.microsoft.com/en-us/azure/mysql/concepts-backup
- https://docs.microsoft.com/en-us/powershell/module/azurerm.keyvault/backup-azurekeyvaultkey

Consider Moving Forward

1. Use Azure role-based access control
   a. https://docs.microsoft.com/en-us/azure/role-based-access-control/role-assignments-portal

Options we are not implementing

- Not free - Advanced Threat Protection
- Not free - Can enable/Configure DDoS Protection Standard to guard against DDoS attacks
  - Protects against “Man in the Middle” attacks

- Not free - Provide security incident contact details and configure alert notifications for security incidents
    - Security incident contact information will be used by Microsoft to contact you if the Microsoft Security Response Center discovers that the customer's data has been accessed by an unlawful or unauthorized party
    - https://docs.microsoft.com/en-us/azure/security-center/security-center-provide-security-contact-details
- Too much work for scope of project - Options to use a manual key for data encryption
- Not necessary for our scope/project - Can create custom roles - https://docs.microsoft.com/en-us/azure/role-based-access-control/custom-roles
- Not many benefits with scope of project - Apply tags to Azure Database for MySQL instances and other related resources giving metadata to logically organize them into a taxonomy
    - https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/tag-resources
    - Used to quickly group and view resources with the same 'Tag' for easier management. Basically a way to easily organize and categorize resources
- Not necessary for our project - Can create custom azure policies
    - https://docs.microsoft.com/en-us/azure/governance/policy/tutorials/create-and-manage
    - Helpful in tracking compliance & personalizing policies for your specific company/product

Passwords

- Passwords should be hashed, not encrypted
- Algorithms such as Sha256 or Sha512 should not be used with passwords because they were not created to do so
- Algorithms such as Argon2, bcrypt, and scrypt should be used instead
- We will be implementing bcrypt