Applying
Azure Well-Architected Framework
to
Azure Synapse Analytics

Datahai

The Microsoft Azure Well-Architected Framework is a **set of 5 pillars** which can be used to help an organisation **improve the quality** of their Azure workloads and infrastructure.

1 Cost Optimisation

2 Operational Excellence

3 Performance Efficiency

4 Reliability

5 Security

We can apply settings and features from the **Dedicated SQL Pools** and **Serverless SQL Pools** to these pillars

Cost Optimisation

Managing costs to maximize the value delivered

Monitor workloads to optimise service usage

Dedicated SQL Pools

- Assign compute using **DWUs** from DWU100 (60GB) to 30000 (18TB) to suit workload. Potentially use a higher DWU to load data then lower the DWU to query data during the day.
- Scale **Up** and **Down**, and also **Pause** using Azure Automation.
- 1 & 3 Year Reserve Pricing per DWU100 can **reduce pricing by up to 65%**

Serverless SQL Pools

- The cost is based on **Data Processed** on-demand which includes both **Reading** data from and **Writing** data to external storage.
- Best practice is to **optimise Data Types** and where possible **use Parquet**

Operational Excellence

Operations processes that keep a system running in production

Dedicated SQL Pools

- Database projects are supported and can be created using SSDT in Visual Studio and Azure Data Studio. A full CI/CD process can then be created using Azure DevOps integration to deploy code changes.
- Enable logging using **Azure Log Analytics**
- Use **Azure Monitor** to surface alerts and metrics to monitor DWU usage, Cache usage, RAM utilisation, and CPU usage.

Serverless SQL Pools

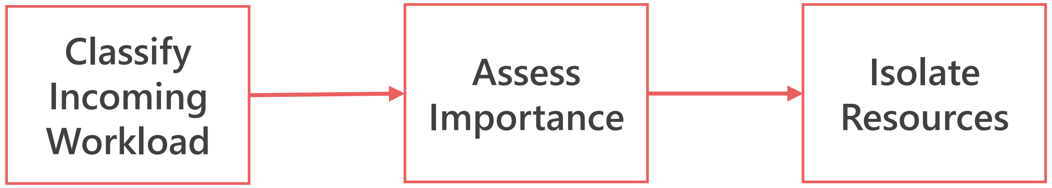
- The is currently no SSDT support, however SQL Scripts to create objects can be source controlled using **Synapse Studio Git integration**.
- System Views to track Data Processed volumes by **Day, Week, and Month**. If limits have been set, monitoring data processed usage is vital.
- **Azure Monitor** can also surface metrics to monitor data processed volumes, plus login attempts, and ended SQL requests

Performance Efficiency

The ability of a system to adapt to changes in load

Dedicated SQL Pools

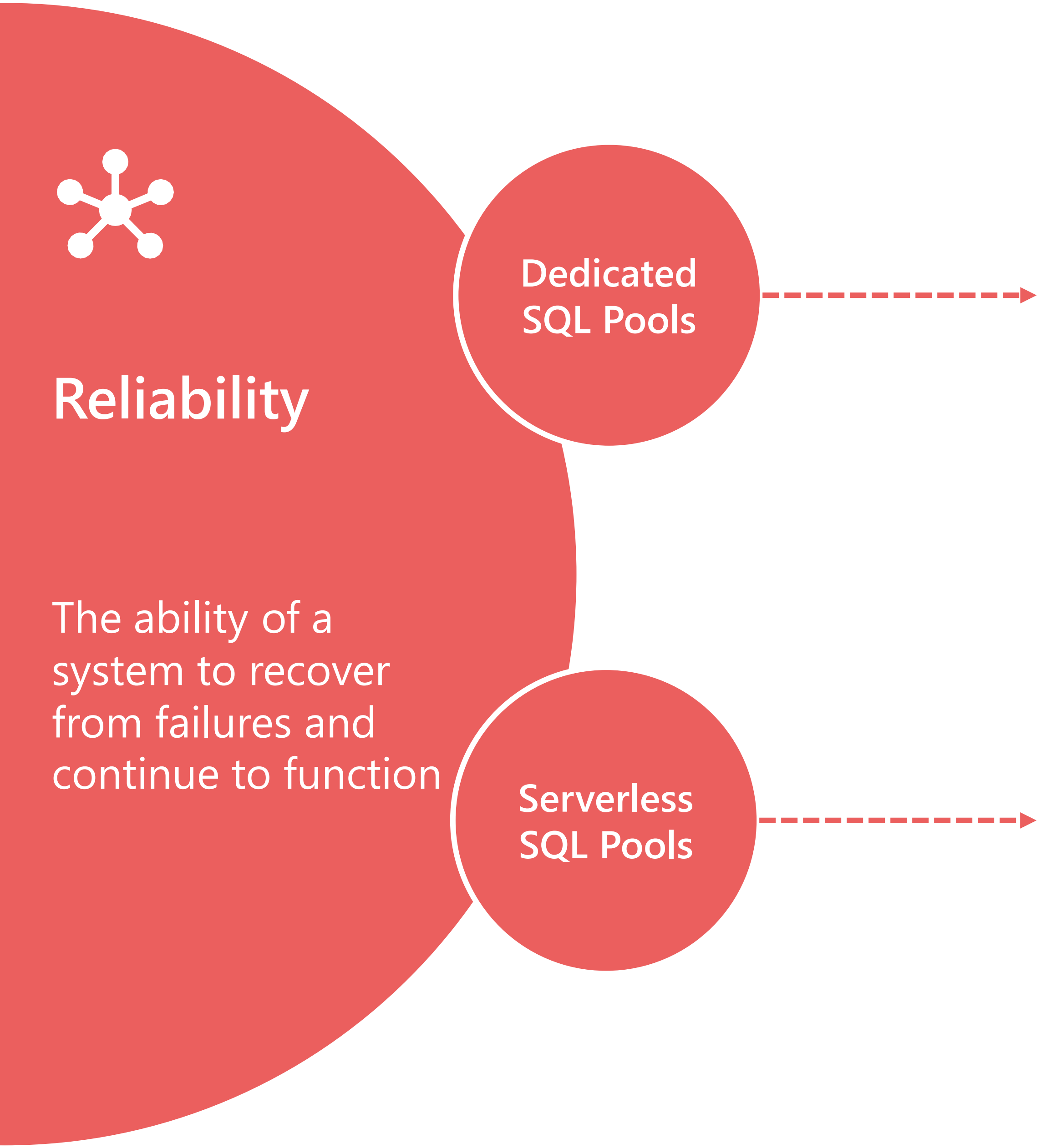
- Currently auto-scaling DWUs "online" is not supported so resizing compute is an offline process.
- The **Workload Classification, Importance, and Isolation** feature allows certain workload sizes to be allocated the required resources.



Classify Incoming Workload → Assess Importance → Isolate Resources

Serverless SQL Pools

- With the **Polaris engine** there is no need to set any workload settings or classification as the service will scale and allocate resources accordingly during **live workloads** execution.



Reliability

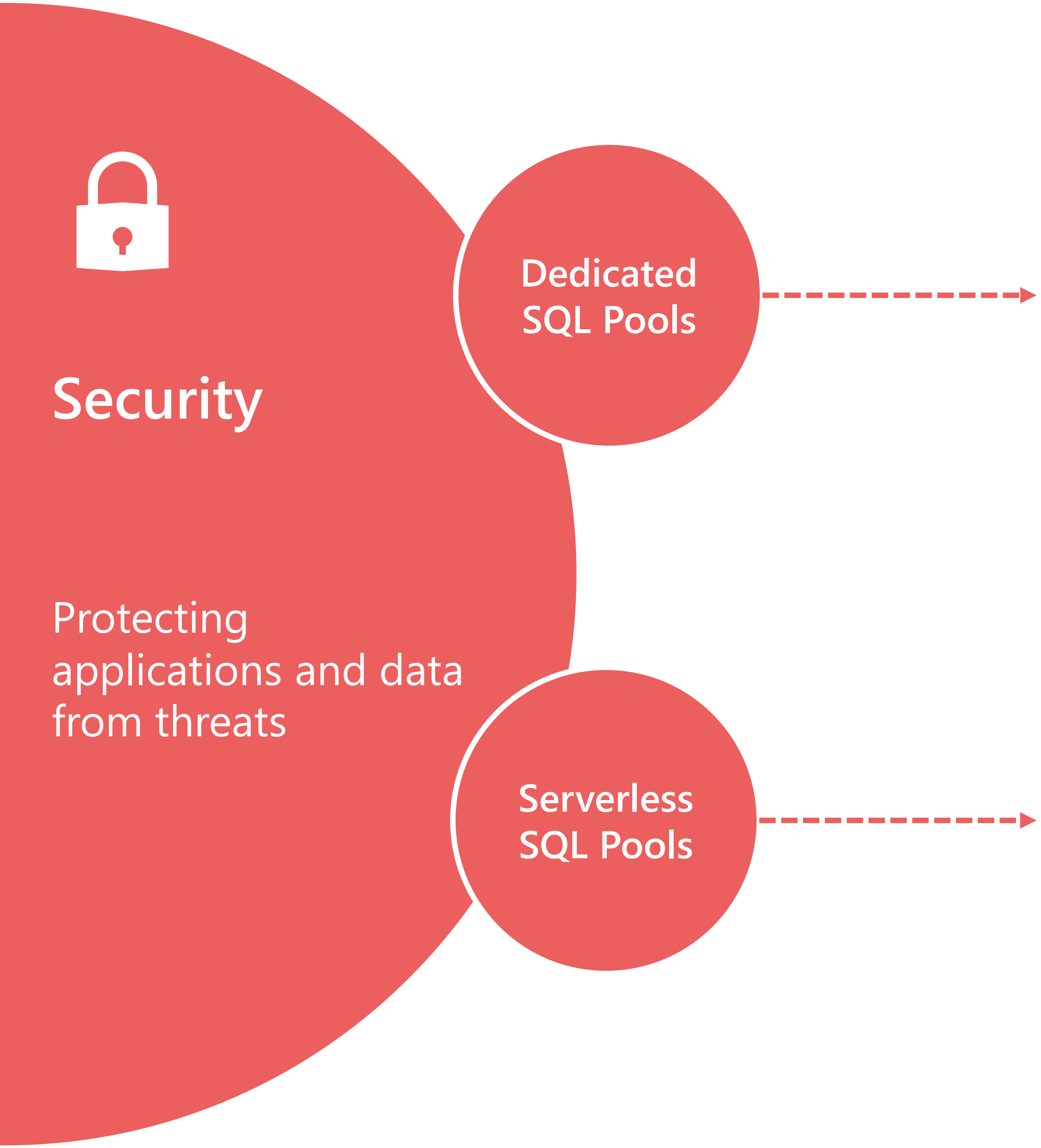
The ability of a system to recover from failures and continue to function

Dedicated SQL Pools

- When the Dedicated SQL Pool is running, **Automatic Restore Points** are created periodically during the day and are available for **7 days.** You can also create **User-Defined Restore Points** if you regularly scale/pause the service (limited to 7 Days retention).
- You can enable/disable **Geo-Backup** which will backup to a paired region, E.G. UK South and UK West.

Serverless SQL Pools

- Automatic Fault tolerance in Polaris engine with an automated query restart process. "Tasks" will be restarted automatically in the event of a failure and this is a seamless process to the user/query executor.



Security

Protecting applications and data from threats

Dedicated SQL Pools

- Use Azure Active Directory Groups and Users, and SQL Logins to secure database objects and data.
- Enable **Transparent Data Encryption** (TDE) to encrypt data at rest.
- Enable **Azure Defender** with periodic Vulnerability Assessment scans to determine current security state
- **Firewall rules** and **Private Endpoints** can be used to allow access to Synapse Analytics (across Synapse services).

Serverless SQL Pools

- As with Dedicated SQL Pools we can use Azure Active Directory Groups and Users, and SQL Logins to secure database objects such as **Views** and **External Tables**.
- Access to the data itself is based on Azure Storage permissions being allocated to the **AAD Group/User**, using time-based **SAS credentials**, and also using the **Managed Identity** of the Synapse service itself. With Cosmos DB we are authenticating using the database key.

References and Further Reading

Document	Link
Microsoft Azure Well-Architected Framework	https://docs.microsoft.com/en-us/azure/architecture/framework/
Azure Well-Architected Review	https://docs.microsoft.com/en-us/assessments/?mode=pre-assessment
Azure Advisor	https://docs.microsoft.com/en-us/azure/advisor/
Azure Advisor Score	https://docs.microsoft.com/en-us/azure/advisor/azure-advisor-score