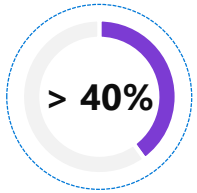
Microsoft

# Azure Database Services for MySQL, PostgreSQL, and MariaDB

Andrei Fateev
Cloud Solution Architect
Microsoft

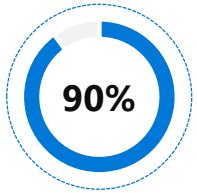

October 2018

# WHY OPEN SOURCE?




**>40%** of CIOs **have open source** as primary software **strategy**


**>60%** of developers **want** to see **cloud providers involved in OSS**




**90% of enterprises deploy open source** in some form

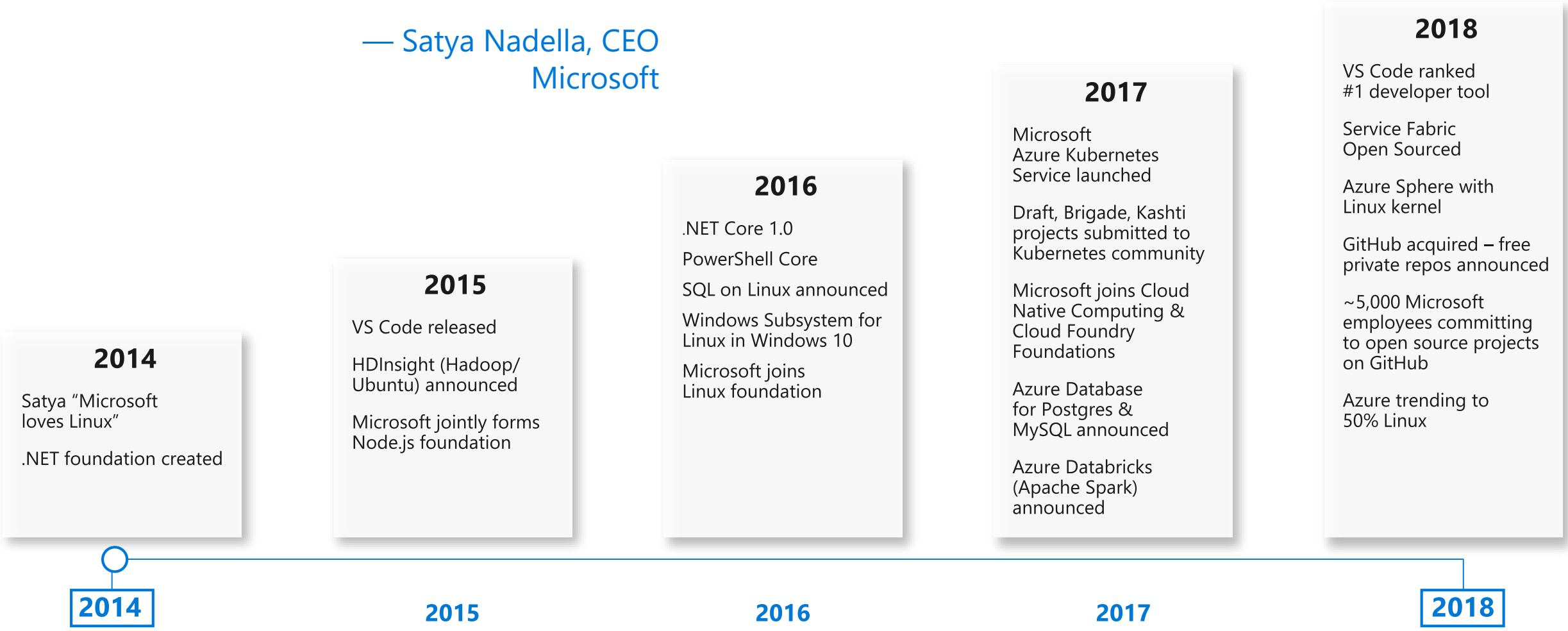
*Forrester 2017, Microsoft CMR, Altoros 2016-2017 trends in Open Source*

Azure

Open Source

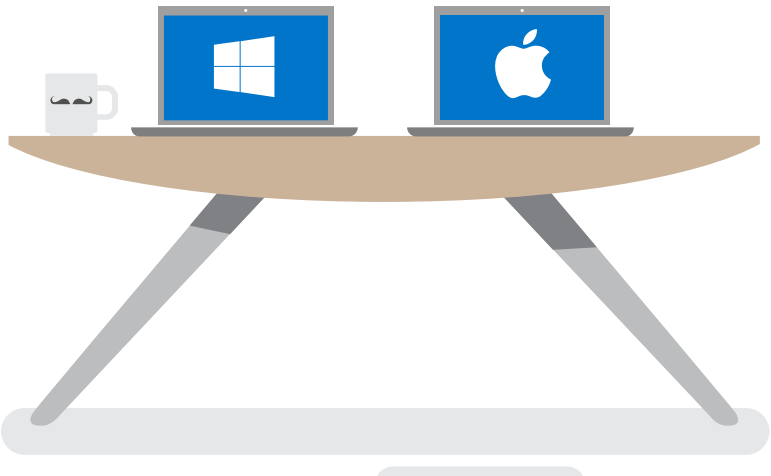
"Judge us by the actions we have taken in the recent past, our actions today and in the future."

— Satya Nadella, CEO
Microsoft



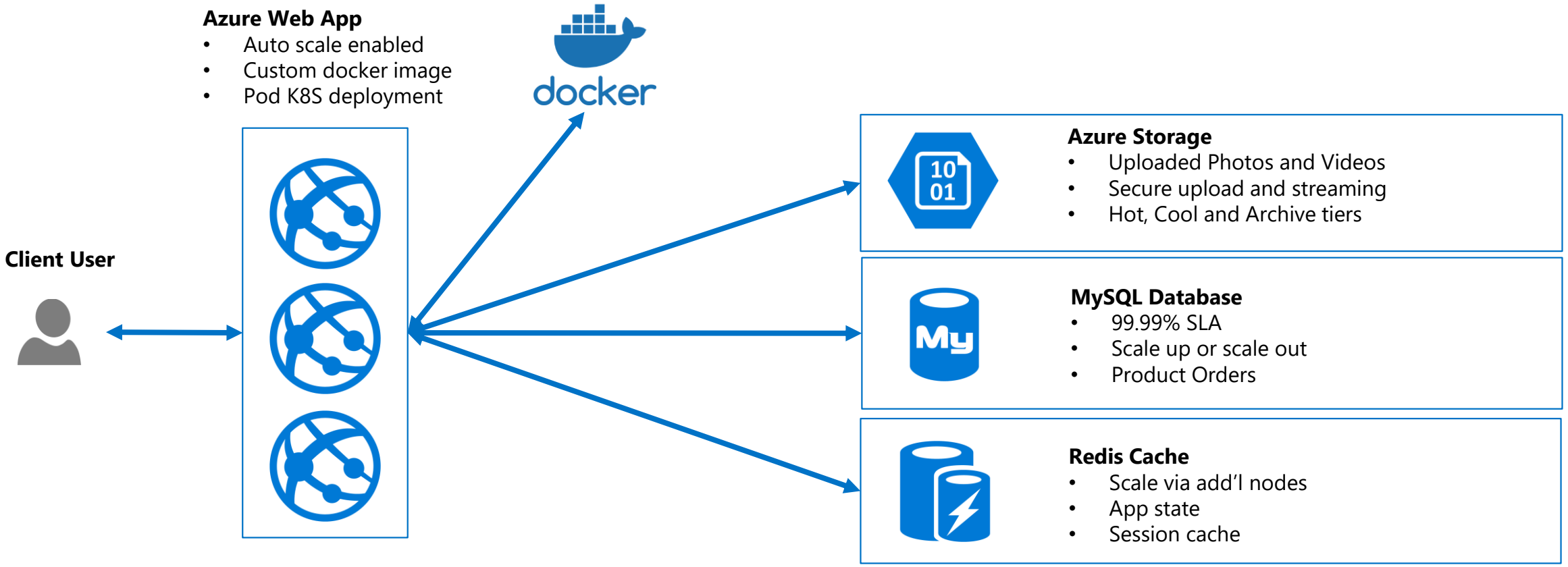
# Cloud App OSS Data Backend

| When you need.... | Because... | But not for... | Use ... |
|---|---|---|---|
| Web frameworks, transactions | Transactions, joins, structured data, familiar SQL query | Quickly changing data schemas | **MySQL, PGSQL, MariaDB** |
| Key-value pair store for heavy read-write, events | Low-cost, fast, massive scale | Rich query | **Tables** |
| Hierarchical, web, mobile | Flexible schema, familiar SQL query, low latency | Complex joins | **CosmosDB** |
| Heavy read-write, events, IoT | Open-source, integration with Hadoop analytics | Operational simplicity | **HBase on HDInsight** |
| Game state, User Sessions | Increasing speed of an app | Primary data store | **Redis Cache** |
| Search service | Integrating search into an app | Primary data store | **Azure Search** |

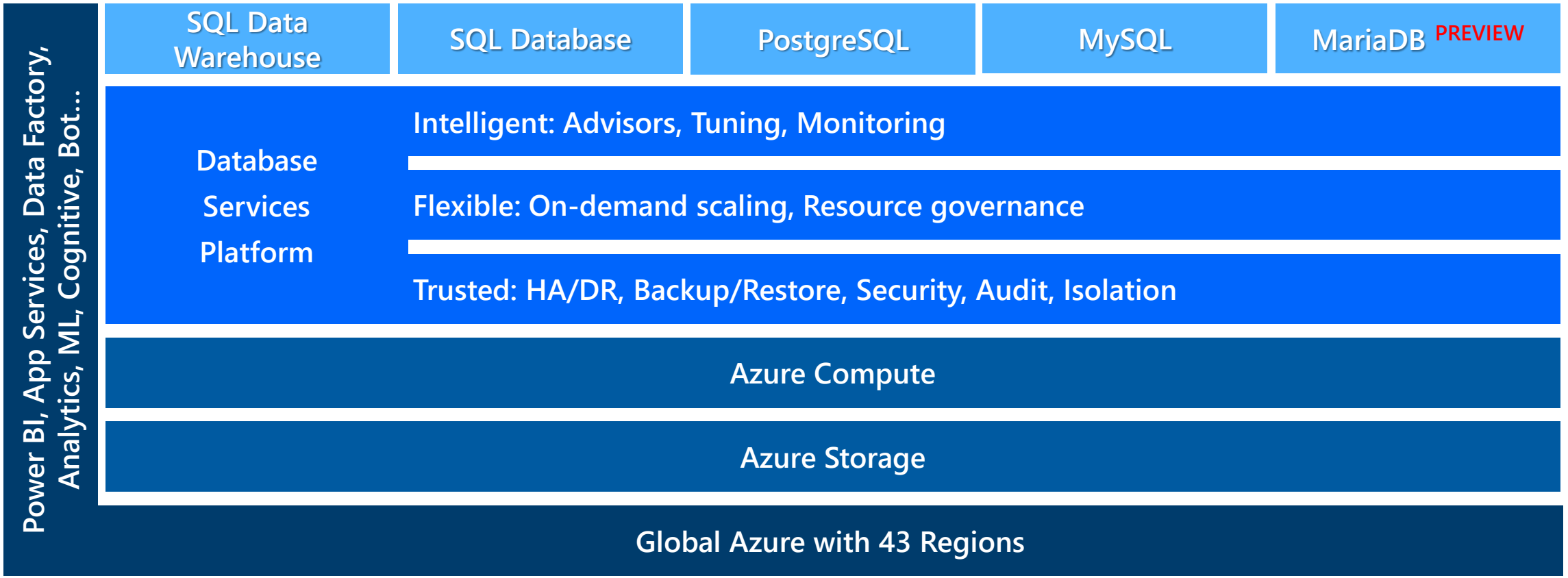
# Demo Wordpress on K8S and MySQL

https://docs.microsoft.com/en-us/azure/app-service/containers/tutorial-multi-container-app#connect-to-production-database



# The journey deepens…


# AZURE RELATIONAL DATABASE PLATFORM



# Service tiers and offerings

# SERVICE TIERS

| Service Tier | Basic | |
|---|---|---|
| Intended Use Case | Built for workloads with light compute needs and variable IO performance | |
| vCore | 1 | 2 |
| Compute Generation | Gen 4, Gen 5 | |
| Storage | 5 GB – 1 TB<br>Magnetic Media | |
| IOPS | Variable | |

# SERVICE TIERS

| Service Tier | Basic | | General Purpose<br>Balanced IO and Compute | | | | |
|---|---|---|---|---|---|---|---|
| Intended Use Case | Built for workloads with light compute needs and variable IO performance | | Ideal for most business workloads offering balanced and scalable compute and storage options | | | | |
| vCore | 1 | 2 | 2 | 4 | 8 | 16 | 32 |
| Compute Generation | Gen 4, Gen 5 | | Gen 4, Gen 5 | | | | |
| Storage | 5 GB – 1 TB<br>Magnetic Media | | 5 GB – 4 TB<br>Remote SSD | | | | |
| IOPS | Variable | | 100-6000 IOPS | | | | |

# SERVICE TIERS

| Service Tier | Basic | General Purpose<br>Balanced IO and Compute | Performance Optimized<br>Memory Optimized |
|---|---|---|---|
| Intended Use Case | Built for workloads with light compute needs and variable IO performance | Ideal for most business workloads offering balanced and scalable compute and storage options | Cache more data for faster transaction processing and higher concurrency |
| vCore | 1, 2 | 2, 4, 8, 16, 32 | 2, 4, 8, 16 |
| Compute Generation | Gen 4, Gen 5 | Gen 4, Gen 5 | Gen 5 only |
| Storage | 5 GB – 1 TB<br>Magnetic Media | 5 GB – 4 TB<br>Remote SSD | 5 GB – 4 TB<br>Remote SSD |
| IOPS | Variable | 100-6000 IOPS | 100-6000 IOPS |

## Supported Versions

MySQL: 5.6, 5.7, 8.0 (coming soon)

PostgreSQL: 9.5, 9.6, 10

# Service features

- Security
- High availability
- Scale up/down on-the-fly
- Backup and restore
- Monitoring and alerting
- MySQL replication
- Migration options

# SECURITY BUILT IN

Identity

- Native authentication
- Threat detection

Control access

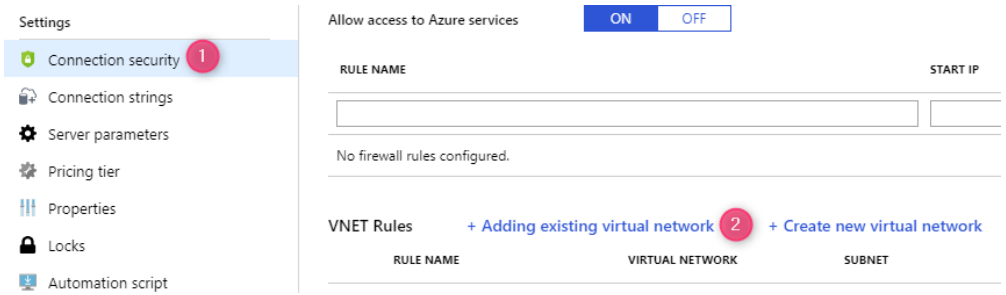
- Secure SSL connectivity
- Server firewall rules
- Virtual Network (SE)

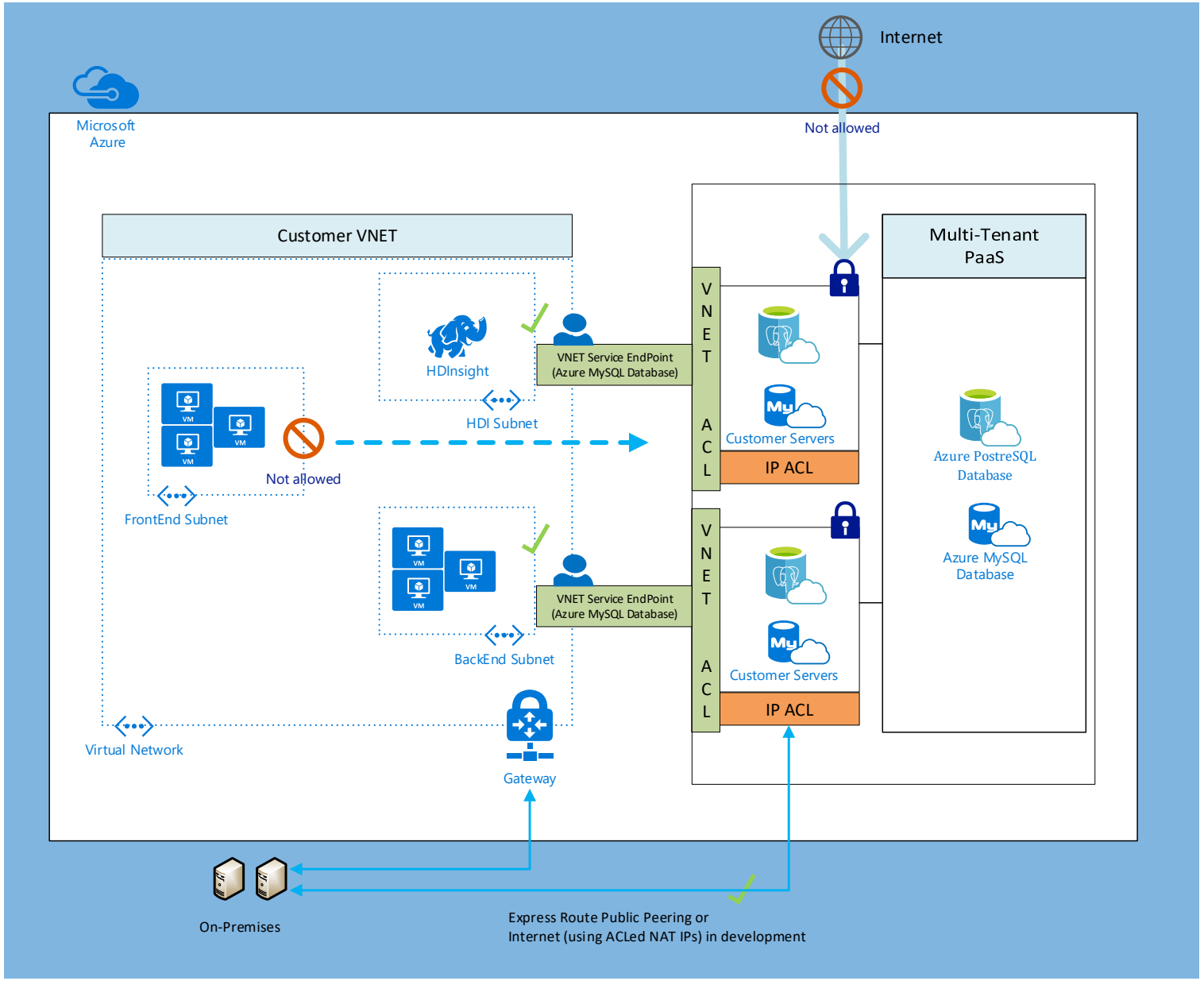
Protect data

- Built-in encryption at-rest for data and backups

# VNET SERVICE ENDPOINT

- Extend VNET private address space to Azure Database for MySQL and PostgreSQL
- Easy to configure (uses Microsoft.SQL service tag)
- Each VNET service endpoint applies to only one Azure region

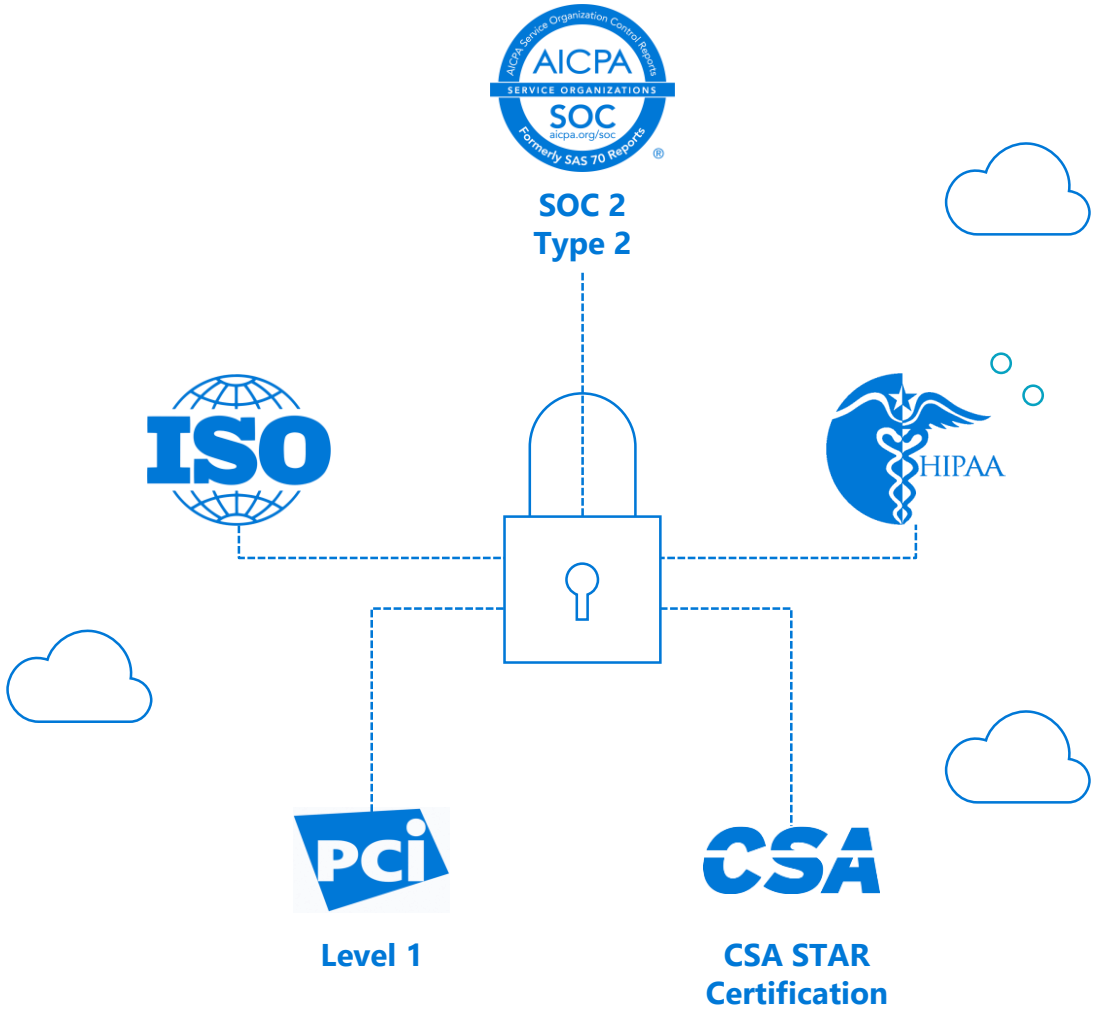


# SECURE AND COMPLIANT

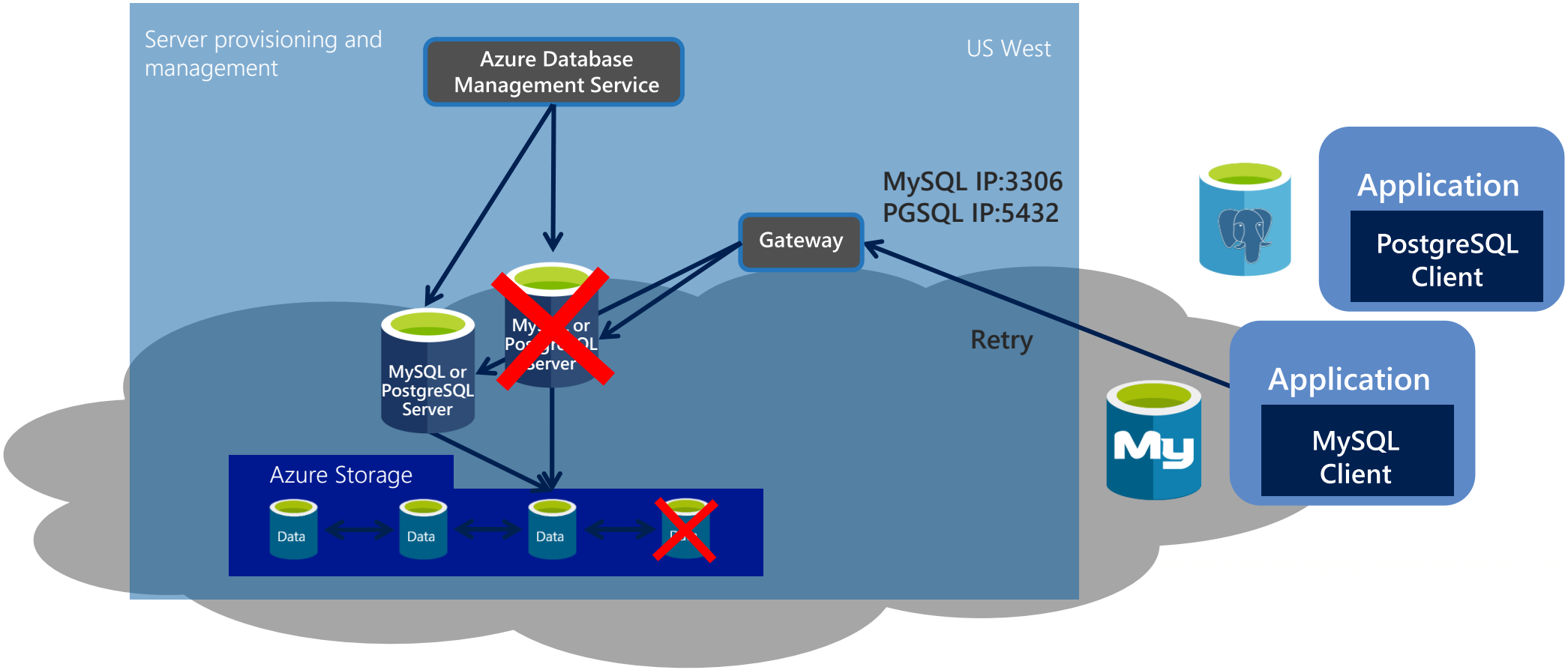
Protect your data with up-to-date security and compliance features with the Azure IP Advantage.

- SOC1 – Compliant
- SOC2 – Compliant
- SOC3 – Compliant
- ISO 27001:2013 – Compliant
- ISO 27018:2014 – Compliant
- CSA STAR Certification – Compliant
- HIPAA / HITECH Act – Compliant
- PCI DSS Level 1 – Compliant
- ISO 27017:2015 – Compliant
- ISO 27018:2014 – Compliant
- ISO 9001:2015 – Compliant
- ISO/IEC 20000-1:2011 – Compliant
- ISO 22301:2012 – Pending



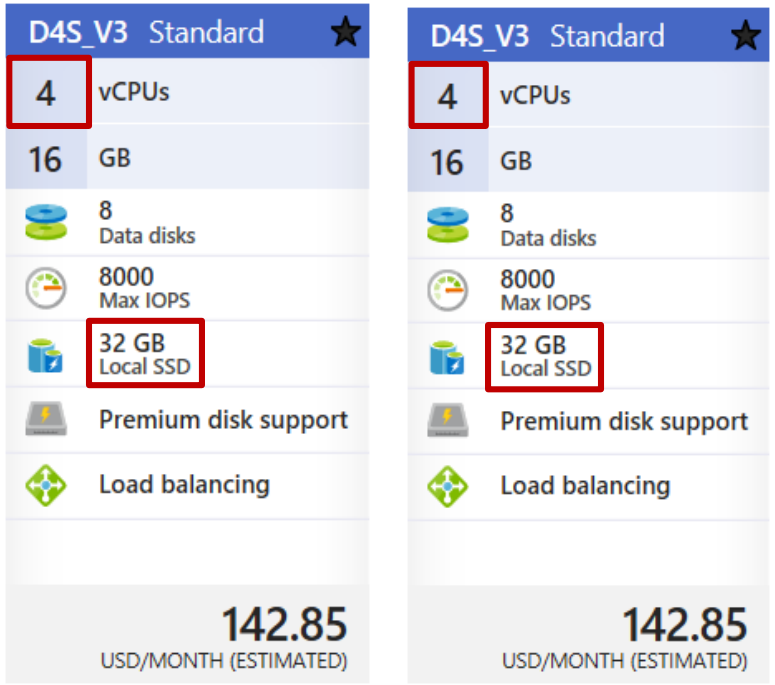
# BUILT-IN HIGH AVAILABILITY

Elastically scale your compute up or down
Independently scale up storage as needed seamlessly
Use replicas only if you need to!



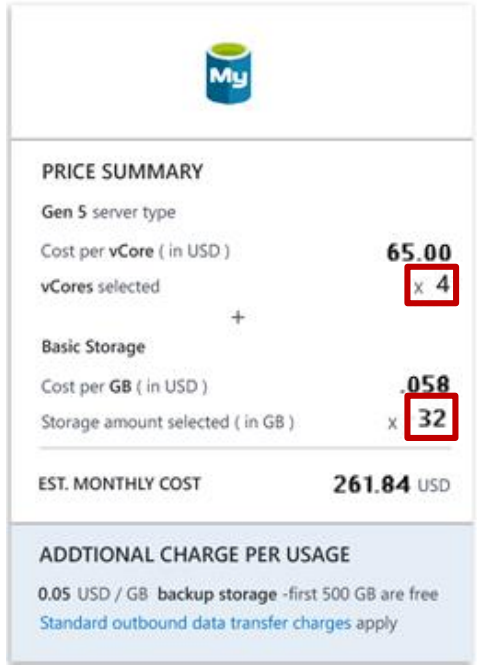
# HIGH AVAILABILITY AND SCALE

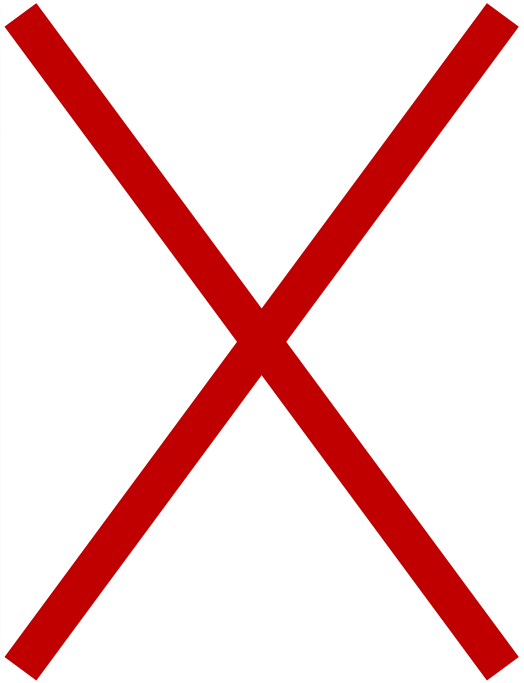
High Availability



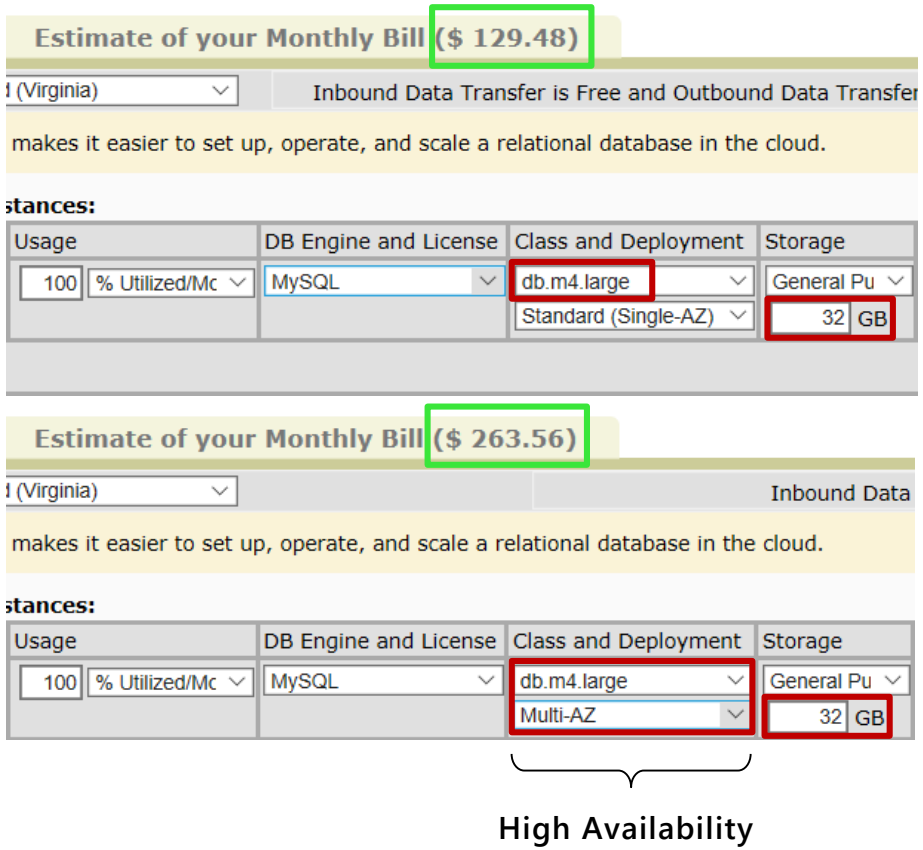

= $285 vs $262 =

High Availability

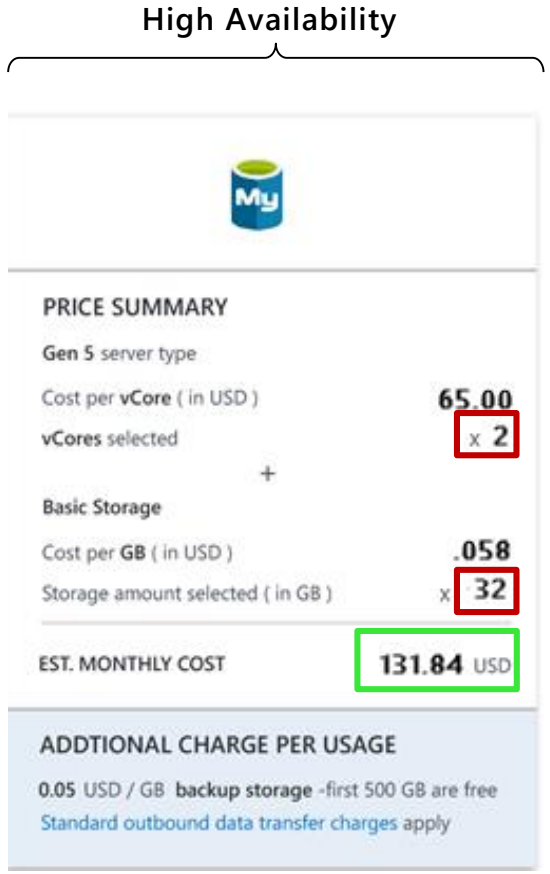


# HIGH AVAILABILITY IN AWS RDS VS. ADS

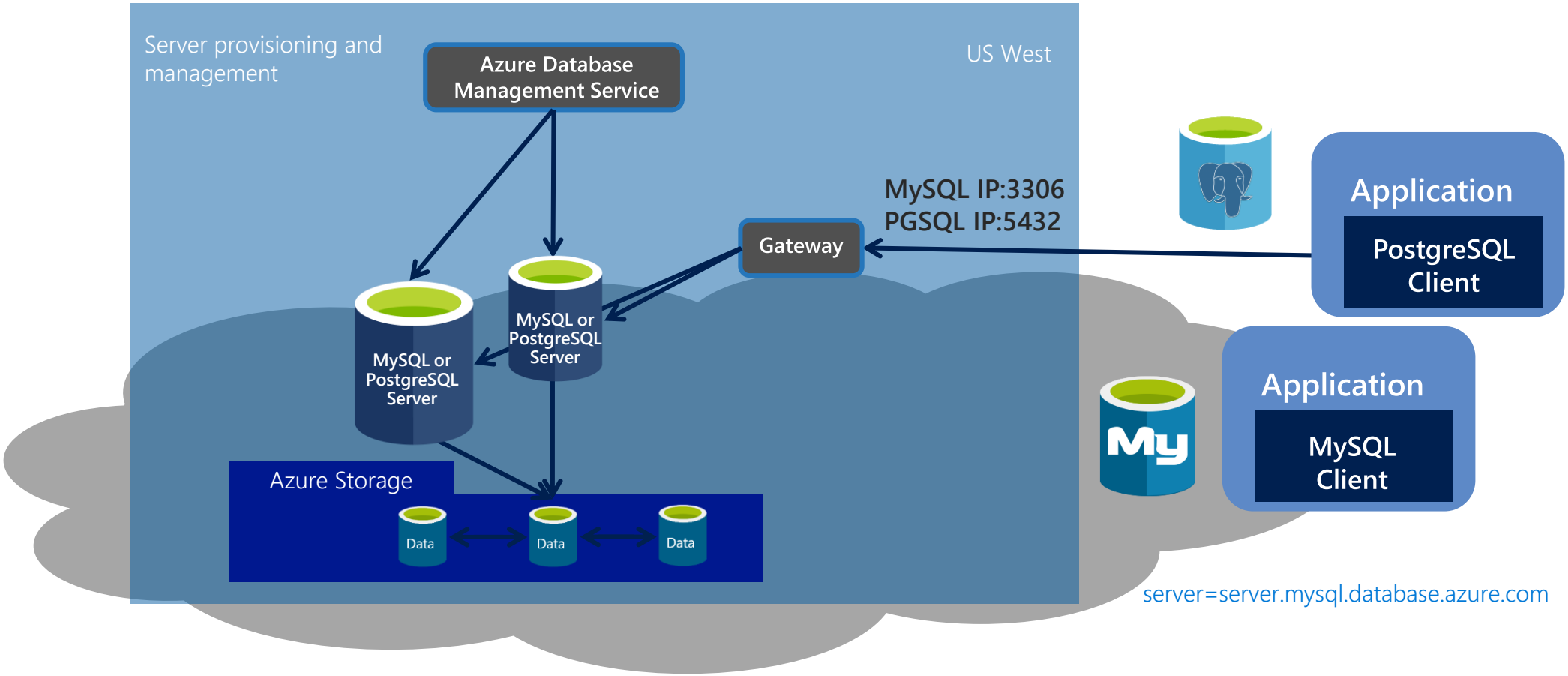



**AWS RDS with a 99.95% SLA is 2x more expensive* than Azure Database for MySQL/PostgreSQL**



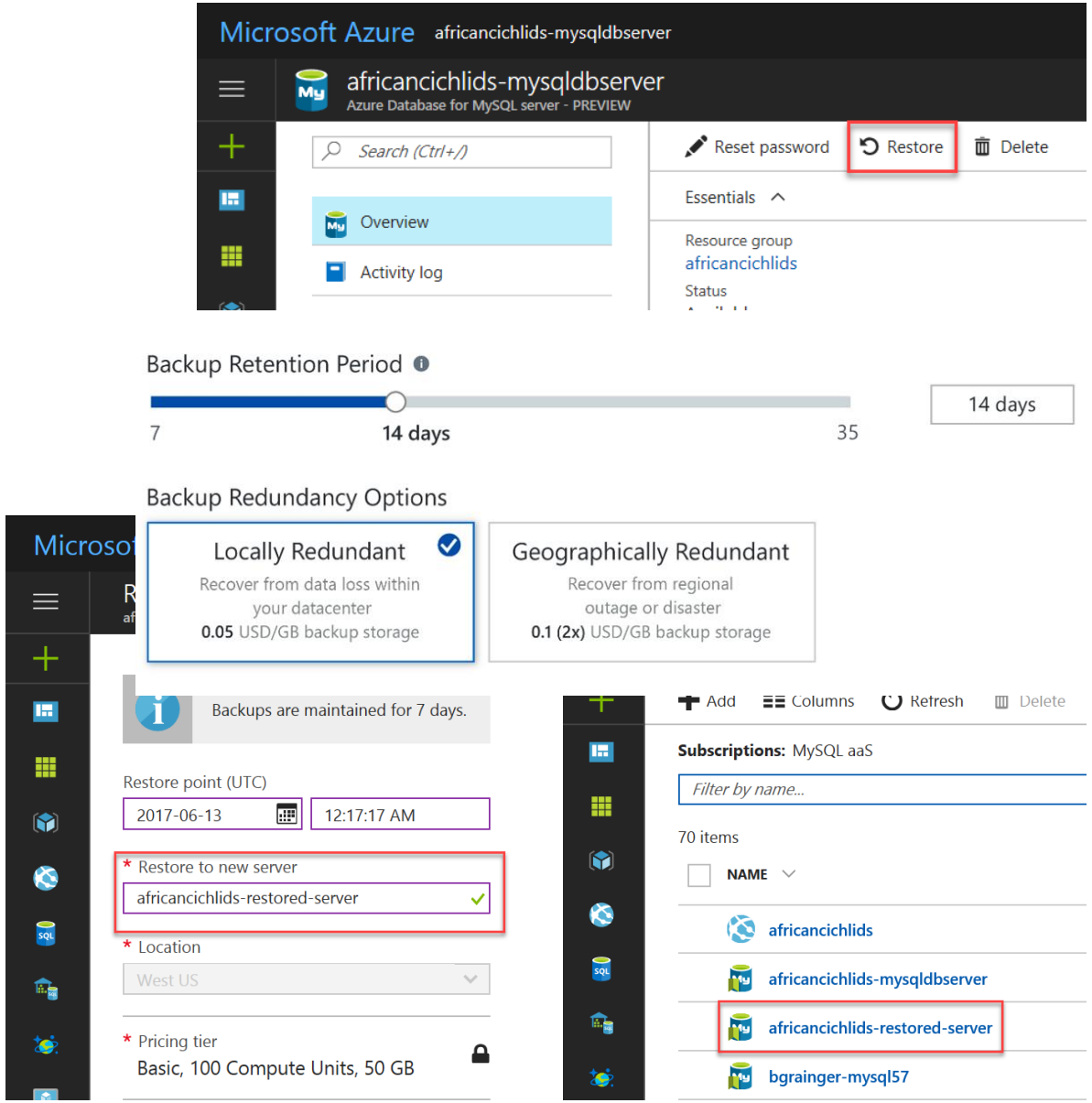

* as of June 2018

# SCALE PERFORMANCE ON THE FLY



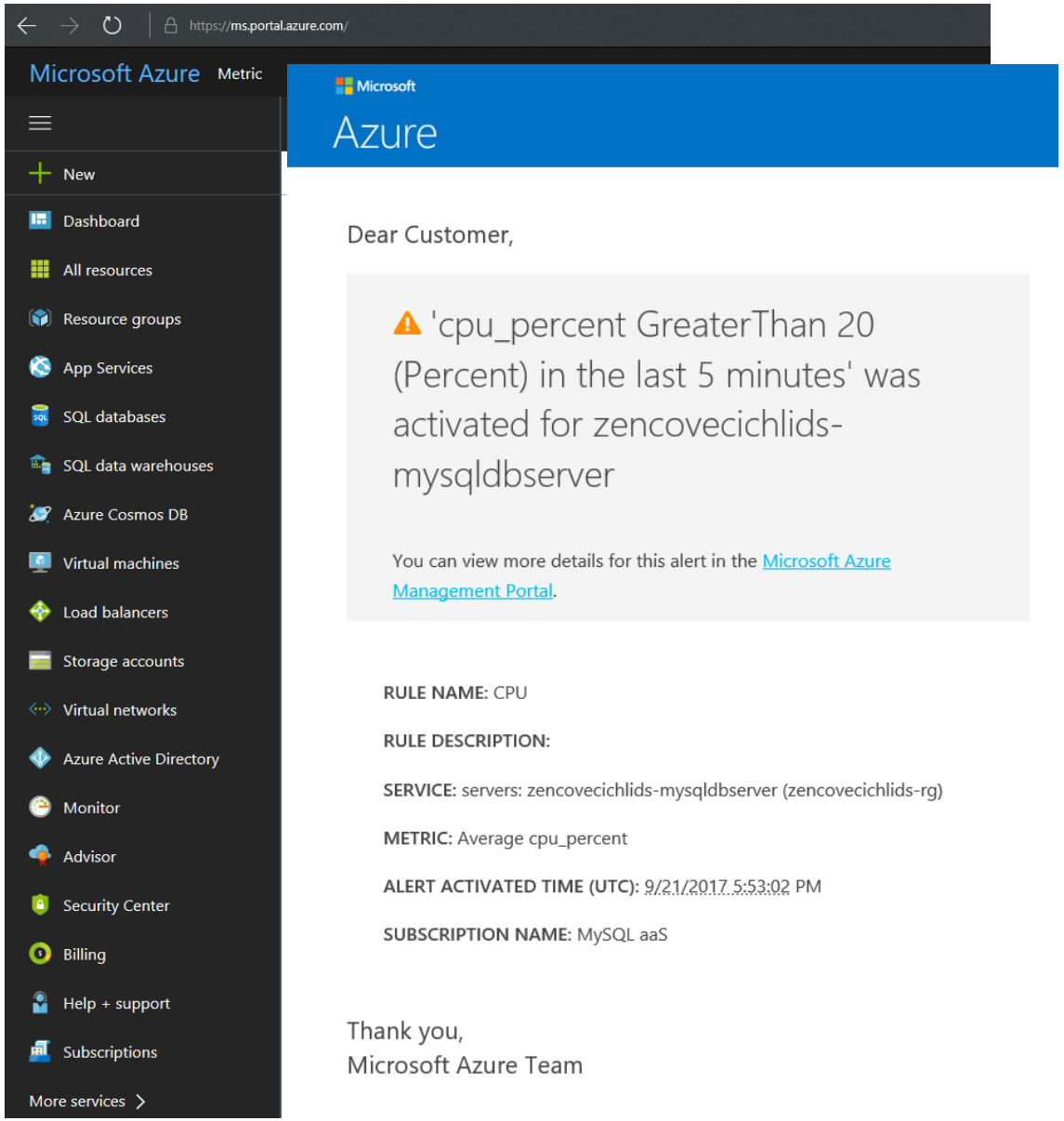
# BACKUP AND RESTORE

- Built-in backups
- Choose LRS or GRS
- Restore from geo-redundant backups for disaster recovery (RPO <= 1 hr)
- 1x Backup storage included
- PITR up to 35 days (min. 7 days)



# MONITORING AND ALERTING

- Built-in monitoring
- Configurable alerts
- Auto notifications
- Enabled for database engine monitoring by default

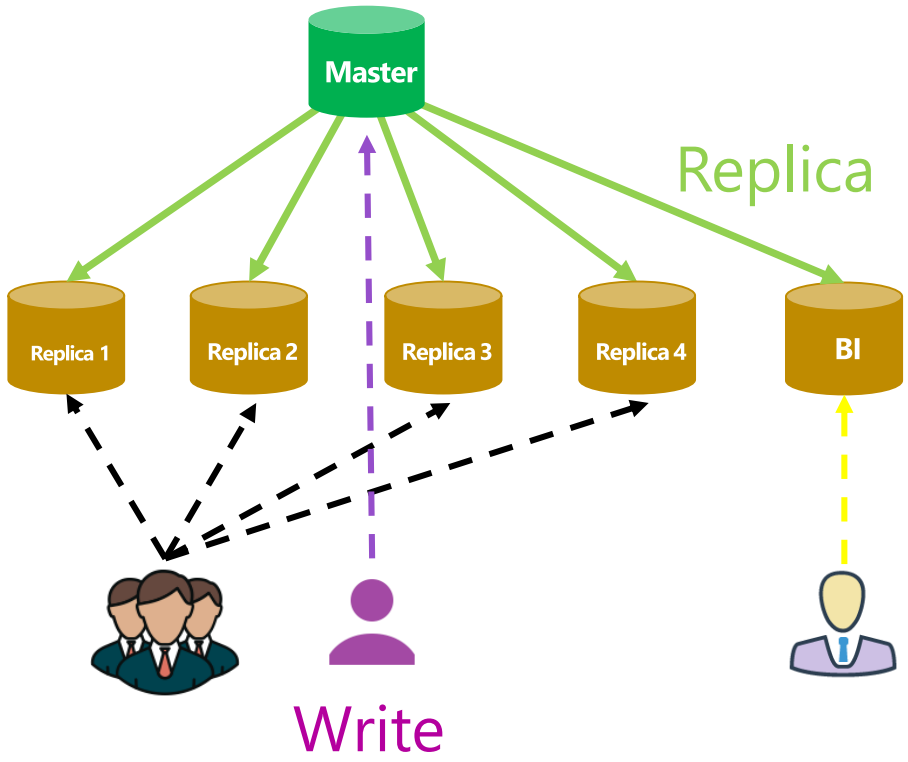


# MYSQL DATA-IN REPLICATION

- Data-in Replication (for migration scenarios)
  - Replica in Azure, master in VM, on-premises server or other cloud provider (with binlog replication)
  - For hybrid, multi-cloud synchronization or migration

# MYSQL AND POSTGRESQL REPLICATION

- Same region replica (for read scale-out)
  - Up to 5 replicas - both master and replica in the same Azure region
- Each replica is a new MySQL/PostgreSQL server
- MySQL: Native replication
- PostgreSQL: In preview



# MIGRATION METHODS

1. MySQL dump and PG dump

   Native commands, no additional setup

2. Migration wizard

   Native or 3rd party tools

- Native way, no additional setup
- Use when application can tolerate downtime

3. Replication

   Attunity, near-zero down time migration

- Third-party solution, needs setup
- Use when application ***cannot*** tolerate downtime

4. Azure Database Migration Service

- Fully-orchestrated migration service
- General Availability (H1 2018)
- Sign up for preview scenarios here

# KEY LEARNINGS FOR ACHIEVING BEST PERFORMANCE

- Network latency
  - Measure with SELECT 1 tests from the client using mysql shell
  - Average should be < 2msec

- Best practices
  - Ensure application and database server is in the same region, created within the same resource group
  - Configure VM (hosting the application) with accelerated networking

# KEY LEARNINGS FOR ACHIEVING BEST PERFORMANCE

- Storage
  - Monitor “IO Percent” and “Storage Percent” using Azure portal or Azure CLI
  - Use mysql (performance_schema) or postgres (pg_stat_statements) to determine queries taking longer times
- Best practices
  - Evaluate/tune indexes and queries as necessary.
  - Ensure adequate IOPS is provisioned for the instance.
  - IOPS scale at approximately 3 IOPS per GB.
  - Rule of thumb is to not exceed 80% of available IOPS (IO Percent) on average.
  - Configure alerts on “IO Percent” and “Storage Percent” as necessary using the Azure portal or CLI
  - Consider Memory Optimized tier

# KEY LEARNINGS FOR ACHIEVING BEST PERFORMANCE

- CPU
  - Monitor “CPU percent” using Azure portal or Azure CLI.
  - Use mysql (performance_schema) or postgres (pg_stat_statements) to determine queries taking longer times
  - High CPU usage is not the only indicator of CPU bottleneck. Do you have a single-threaded app?
- Best practices
  - Strongly recommend using General Purpose
  - Validate/tune indexes and queries as necessary
  - Ensure server log parameters are tuned for production workload
  - Tune database connections from the application. Use connection pooling or front end or application caching (if suitable). Consider Memory Optimized.
  - Configure alerts on “CPU percent” as necessary using Azure portal or CLI