Tales from Building a SQL Server Data Warehouse in Azure

Experiences & lessons learned from a migration to Azure

August 21, 2017

Lead Analytics Architect, SentryOne
Microsoft Data Platform MVP

Blog: www.sqlchick.com

Twitter: @sqlchick

Melissa Coates

Goals for This Session

1. Share decisions and lessons learned from a recent Azure implementation
2. Introduce key concepts for a deployment to Azure
3. Discuss items involved with building a DW environment in Azure

Azure services & features change frequently.

The information in the presentation is accurate as of mid-August 2017.

Agenda

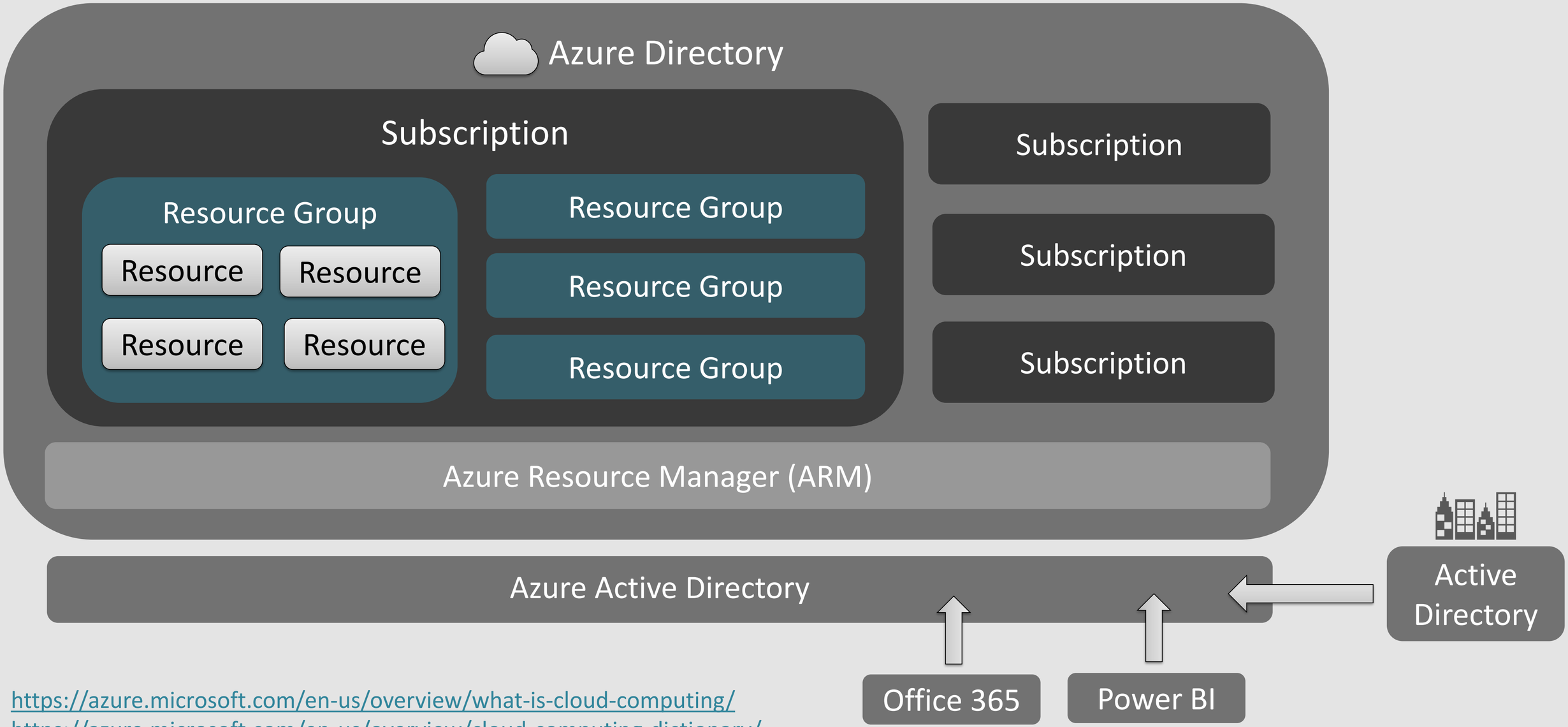
- ❑ Key Azure Concepts
- ❑ Deciding on Azure VM vs. Azure SQLDB vs. Azure SQLDW
- ❑ Additional Planning Considerations
- ❑ Building the Azure Environment
  - ❑ Structuring Dev, Test, & Prod Environments
  - ❑ Naming Conventions & Tags
  - ❑ Configuration Decisions
  - ❑ Automation & Scheduling
  - ❑ Monitoring the Solution
- ❑ Key Takeaways & Open Q&A

Out of scope:

- Linux deployments
- Azure Stack & private cloud deployments
- On-premises physical & virtualized deployments (including Fast Track DW & APS/PDW)
- Security in depth (though we touch on a few points)
- Networking & connectivity
- Troubleshooting, performance tuning & growth
- Details on how to install and configure SQL Server

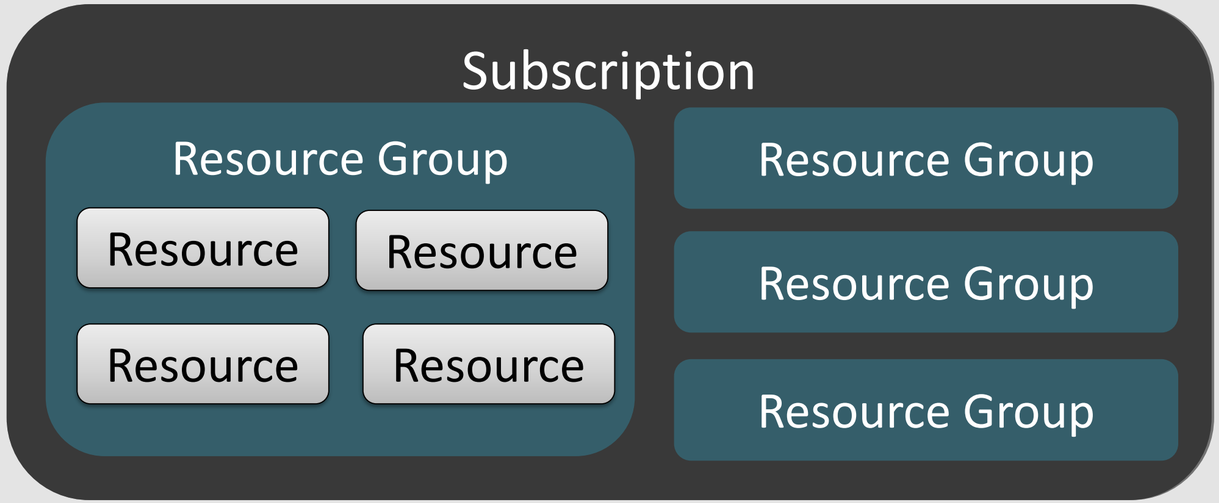
Key Azure Concepts

The Azure Lingo



https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/
https://azure.microsoft.com/en-us/overview/cloud-computing-dictionary/

Resource Groups



We have learned:

- ✓ Keep resource groups more narrow than broad
- ✓ Select the region (location) carefully

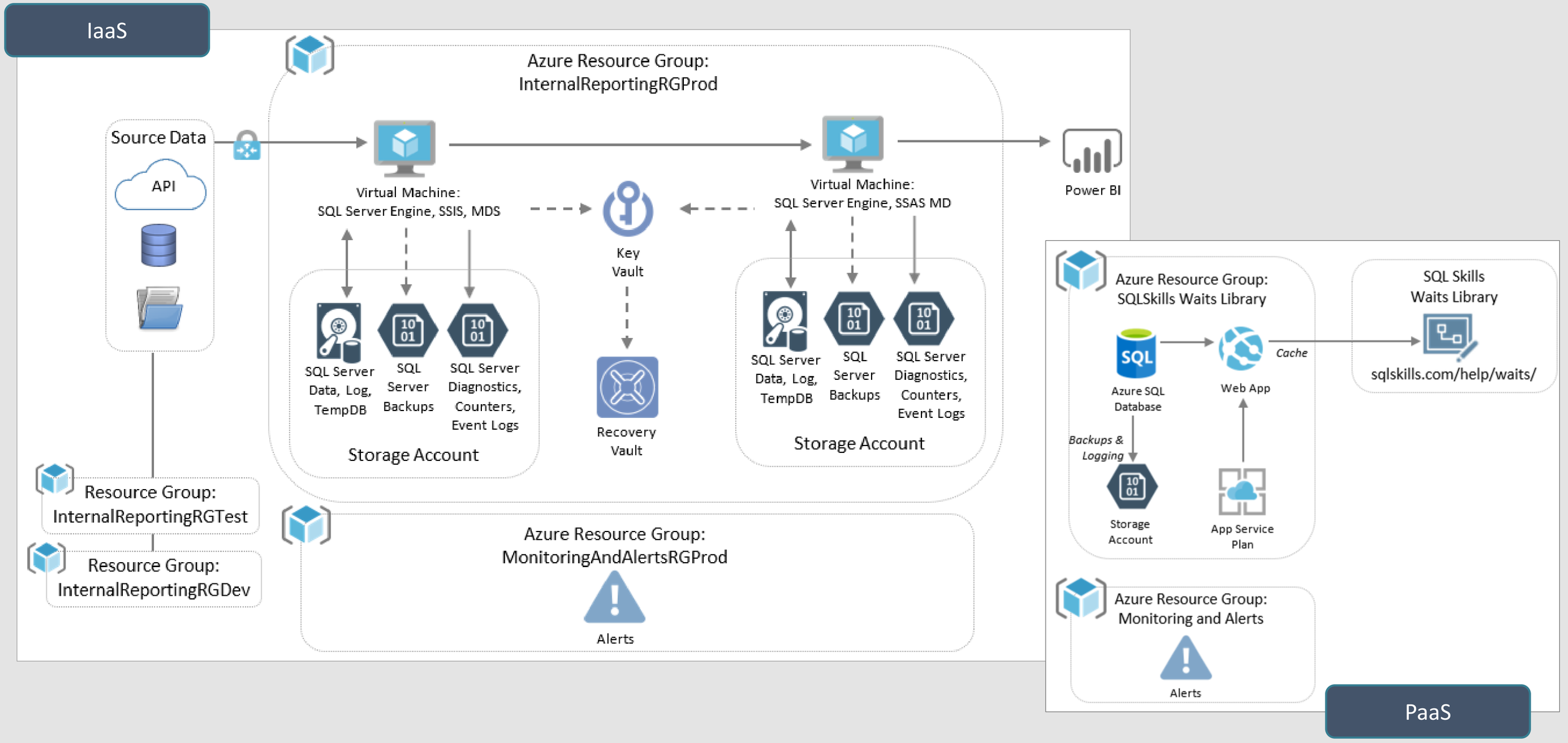
Planning for resource groups is critical

Focus on:

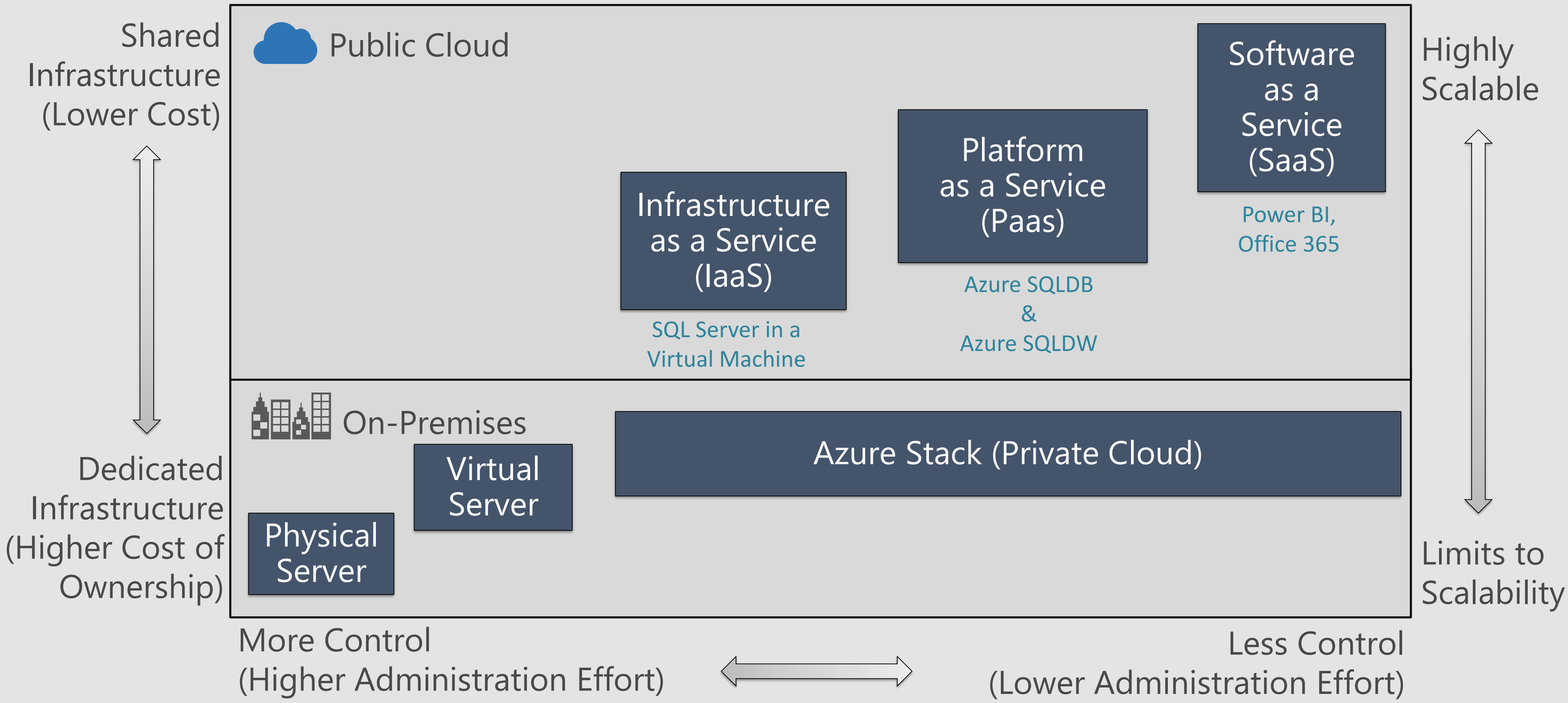
- Logical organization
- Permissions
- Policies

Scope of ARM automation scripts (exception: Resource Explorer)

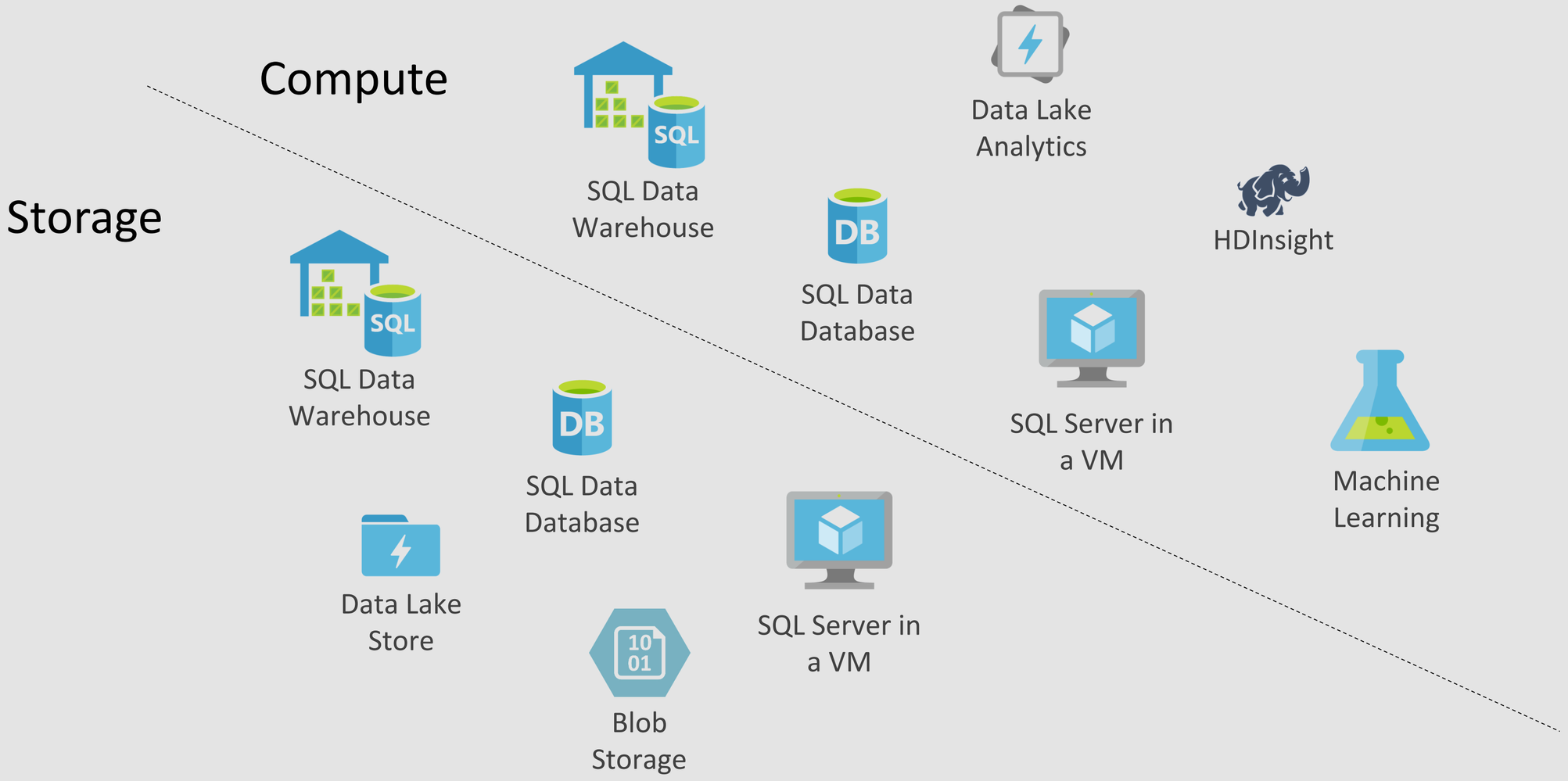
Examples – Resources & Resource Groups



IaaS vs. PaaS vs. SaaS



Compute vs. Storage



Some resources scale, or even pause, compute separately from storage.

Deciding on Azure VM
vs. Azure SQLDB
vs. Azure SQLDW

Comparing the SQL Offerings in Azure

SQL Server in a Virtual Machine (IaaS)

Run full workload within an Azure virtual machine, including SQL Server, SSIS, SSAS, SSRS, etc

Azure SQL Database (PaaS)

A relational database-as-a-service (DBaaS)

Non-Managed

A traditional Azure SQLDB deployment (isolated DB)

Managed Instance

Newer - closer feature parity to SQL Server (instance level features)

Azure SQL Data Warehouse (PaaS)

An data warehouse-as-a-service (DWaaS) optimized for performance and large-scale distributed workloads

MPP architecture (massively parallel processing)

Comparing the SQL Offerings in Azure (2/2)

SQL Server in a Virtual Machine (IaaS)

Best for:

- ✓ Migrating existing solutions
- ✓ Running any software and/or all SQL Server features
- ✓ Administering all aspects
- ✓ Bring your own license (Software Assurance)
- ✓ Isolated dev/test environments
- ✓ SLA: for the VM

Azure SQL Database (PaaS)

Best for:

- ✓ New database solutions (If a DW it should be < 4TB data size--sharding across DBs is not suitable for DW workloads)
- ✓ OLTP with scaling & pooling needs (unpredictable workloads)
- ✓ Reduced administration of DB, OS, HA, and DR
- ✓ SLA: for the database

Azure SQL Data Warehouse (PaaS)

Best for:

- ✓ DW with larger data volumes (bare min. of 1-4TB)
- ✓ Ability to scale compute up/down, or pause (elasticity)
- ✓ Data Lake Store integration (relational + nonrelational data)
- ✓ Reduced administration
- ✓ SLA: for the database

More info: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-paas-vs-sql-server-iaas

Key Differences with Azure SQL Database

Many features go first to Azure SQLDB (“cloud first”).
However, there are some key features
not available in SQLDB (PaaS):

- PolyBase
- R Services
- Change data capture
- CLR
- DB snapshots
- Some T-SQL syntax
- Profiler
- Non-primary filegroups

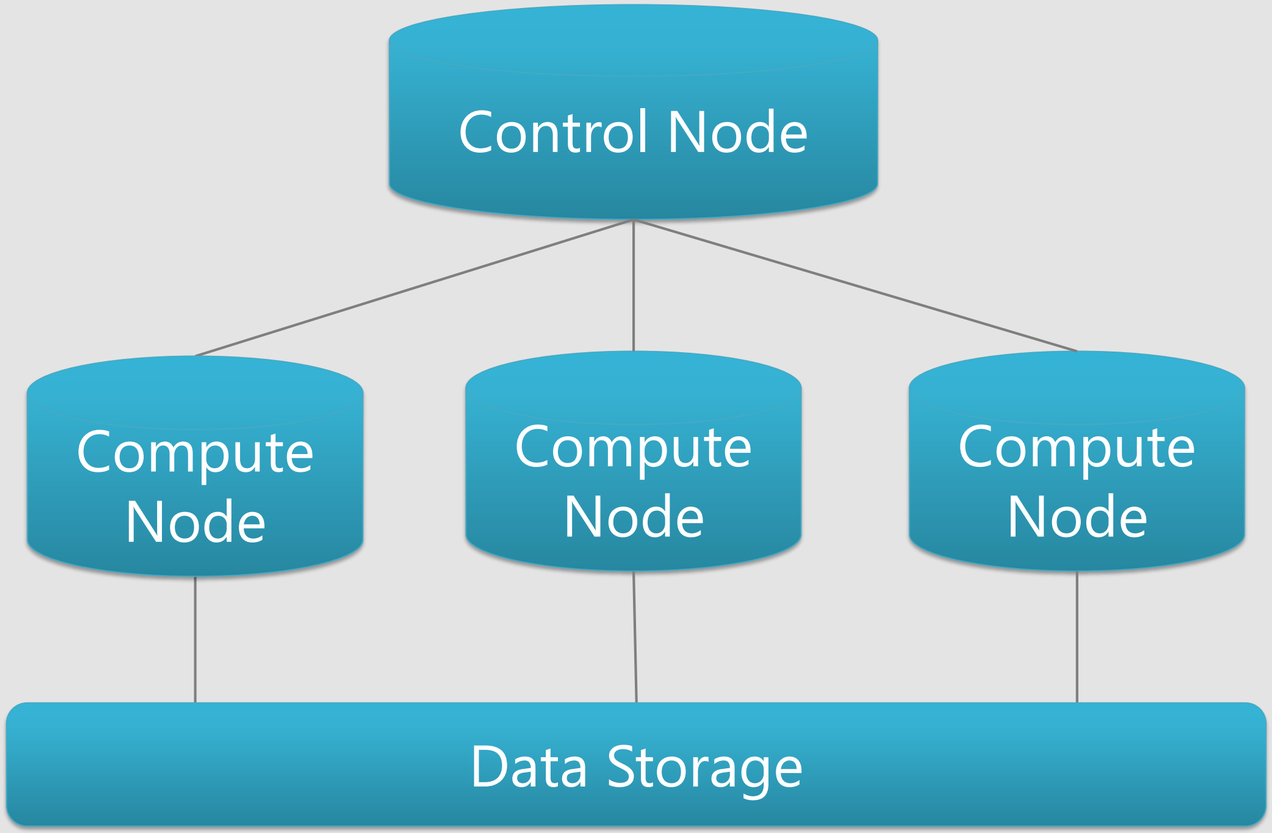
Also, some features rely on Premium edition:

- Columnstore indexes

Feature	SQL Server	Azure SQL Database
Active geo-replication	Not supported - see Always On Availability Groups	Supported
Always Encrypted	Supported	Supported - see Cert store and Key vault
AlwaysOn Availability Groups	Supported	Not supported - See active geo-replication
Attach a database	Supported	Not supported
Application roles	Supported	Supported
Auto scale	Not supported	Supported - see Service tiers
Azure Active Directory	Not supported	Supported
Azure Data Factory	Supported	Supported
Aud[illegible]	Supported	Supported

Full list:
https://docs.microsoft.com/en-us/azure/sql-database/sql-database-features

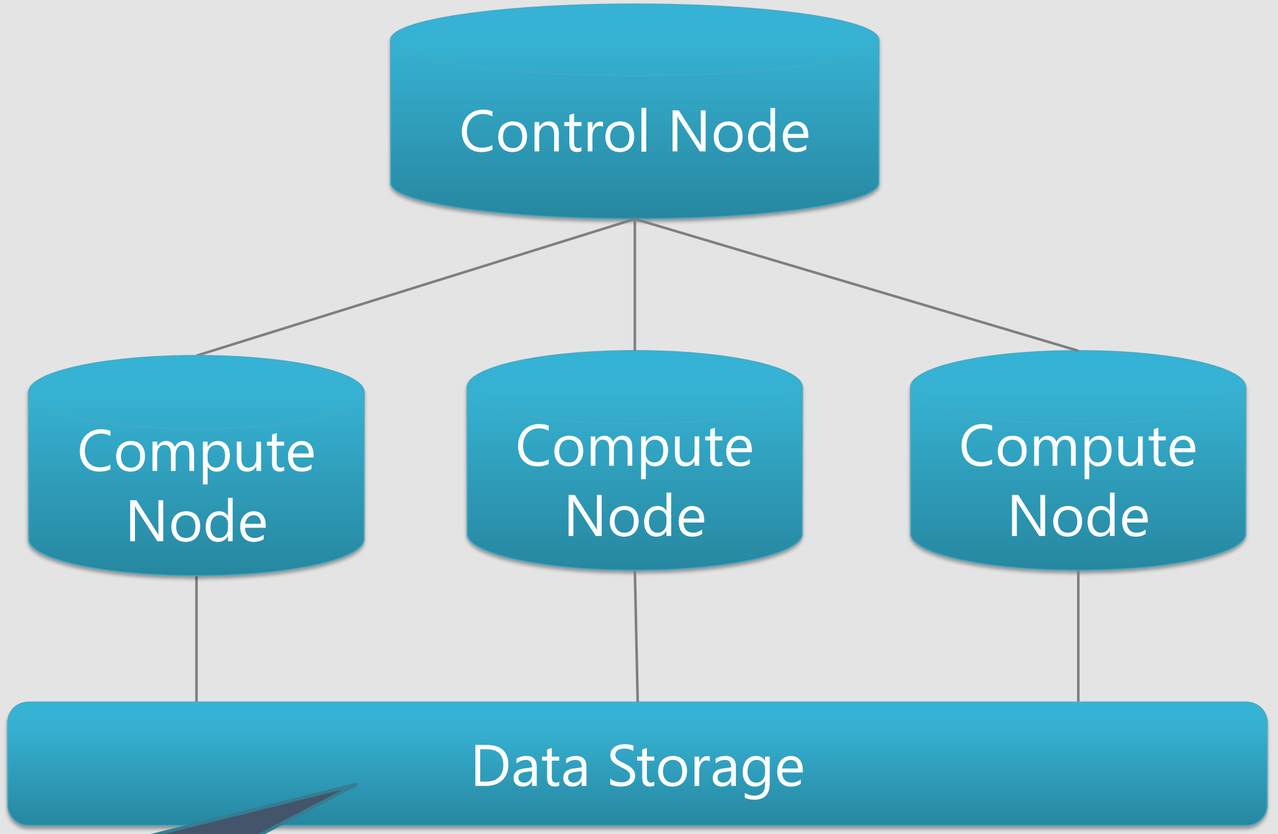
Key Differences with Azure SQL Data Warehouse (1/2)



- MPP Architecture
- Shared-Nothing Architecture
- Decoupled Storage & Compute
- Scale Up, Down, Pause
- PolyBase

Key Differences with Azure SQL Data Warehouse (2/2)

- Not All Features Supported
- Different Data Loading Patterns
- Distribution Keys are Critical
- Denormalized Data Model is Best



Take time to educate yourself on the key differences with the MPP architecture—it will affect the design & the data load processes

Our Decisions on What to Use

SQL Server in a Virtual Machine (IaaS)

We are using a VM for:

- ✓ SQL Server DW
- ✓ Integration Services
- ✓ Analysis Services (MD)
- ✓ Master Data Services
- ✓ R Services

We need VMs for SSIS and SSAS anyway, so we couldn't justify migrating the relational DW to a PaaS solution at this time

Azure SQL Database (PaaS)

We are using SQLDB for:

- ✓ A specific use case: public reporting solution via SQLSkills Waits Library
- ✓ This SQLDB is loaded from the DW (in SQL Server)

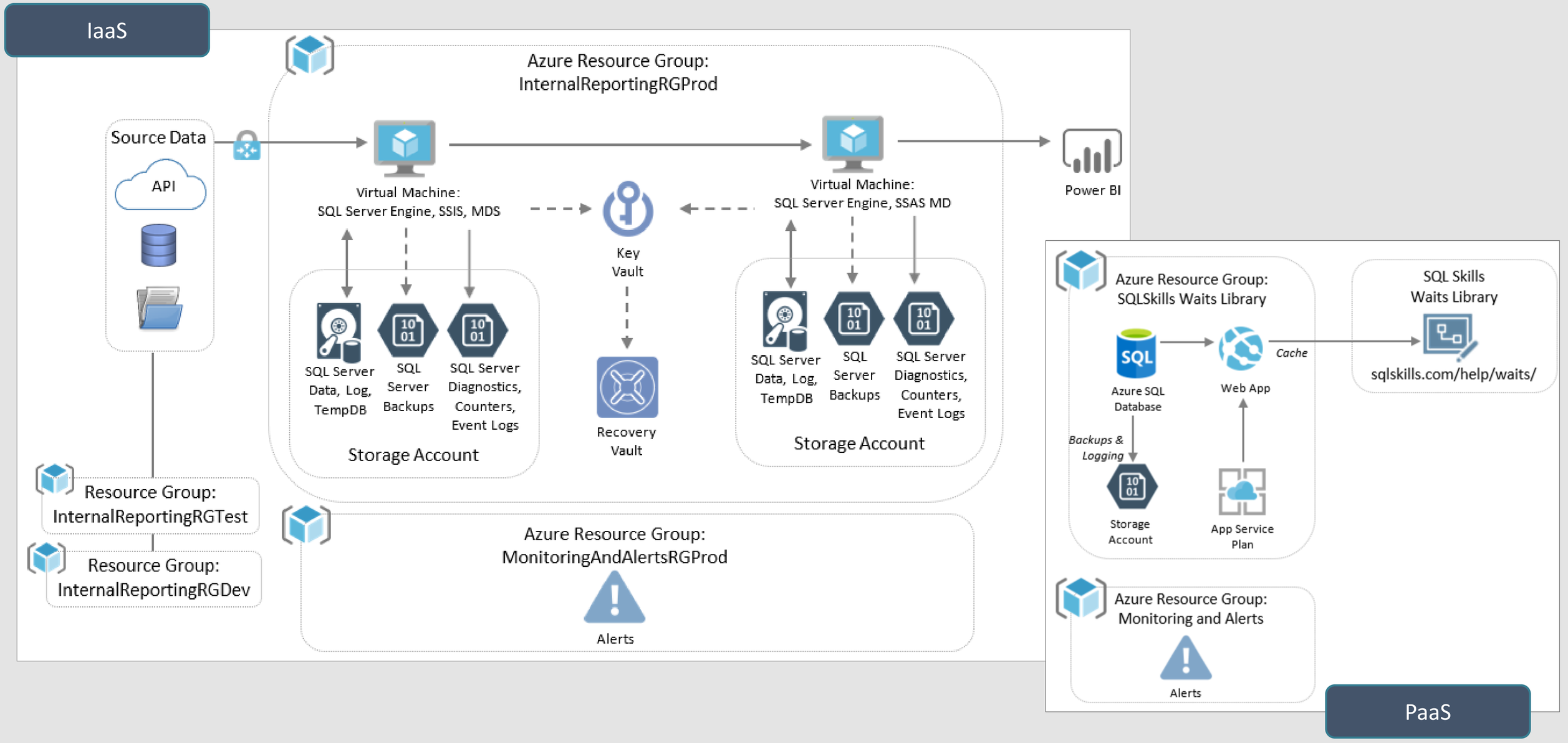
Azure SQL Data Warehouse (PaaS)

We put SQLDW on roadmap:

- ✓ Future data growth
- ✓ Future PolyBase integration with multi-structured data in Azure Data Lake Store

Requires some refactoring of the data load processes so we are planning the move to SQLDW strategically

Our Current State



Additional Planning Considerations

Our Goals & Requirements for the Move to Azure

Expand infrastructure to support future growth

Ensure ability for Analytics Team to manage environment independently

Support existing solutions with little to no redesign or refactoring

Secure connectivity via VPN

Minimize cost where practical

Acceptable performance of hourly ETL jobs

Think about trade-offs you're willing to make for cost, performance, security, regulatory compliance, DR/backups/redundancy, and simplicity

Initial Planning Before Provisioning Any Resources (1/3)

First Steps

- Licensing and Editions
- Full Cloud Implementation vs. Hybrid (Partially On-Premises)
- IaaS vs. PaaS decisions; feature comparison

Capacity Planning & Cost Estimates

- Big 3: Storage, memory, CPU
- Networking
- Scalability needs
- Cost estimates: https://azure.microsoft.com/en-us/pricing/calculator/

High Availability & Disaster Recovery

- Down time sensitivity (RPO, RTO)
- SLAs from Azure

Compliance & Security

- Compliance: https://www.microsoft.com/en-us/trustcenter/compliance/default.aspx

Initial Planning Before Provisioning Any Resources (2/3)

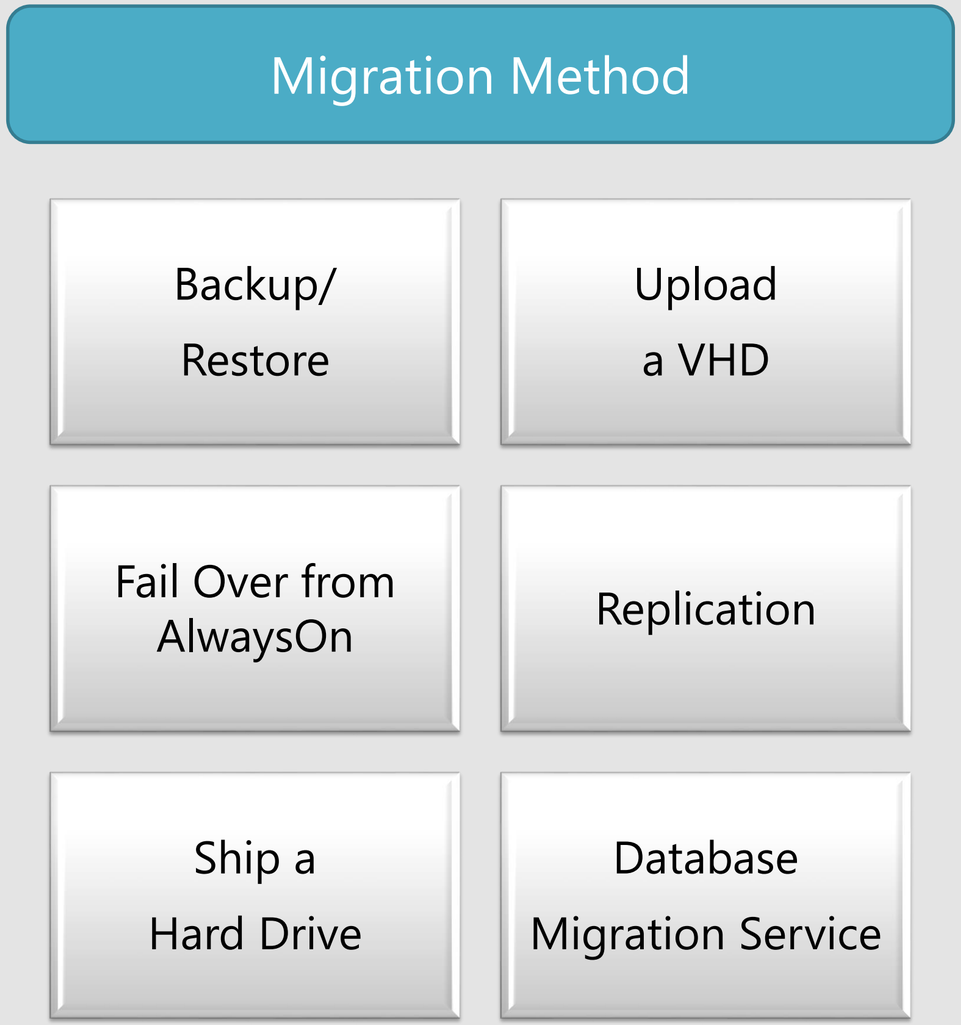
Domain service accounts & credentials

- How administrative & owner permissions will work if decentralized
- Read and/or write permissions for source or related systems
- One domain service account per service, per environment
- Sync to Azure Active Directory for domain users & groups
- Service principals for certain resources in Azure are very important

Azure region: primary location & data redundancy

- Geographic location of data
- Proximity to business users
- Co-location of related resources
- Minimizing latency
- Minimizing data egress charges (very inexpensive though)
- Not all resources/services are available in every region

Initial Planning Before Provisioning Any Resources (3/3)



We went with a backup/restore approach for the SQL Server files for the DW.

The exception to this was SSISDB custom auditing & logging objects. This DDL was deployed from SSDT after the SSISDB catalog was configured.

Info on migration techniques:
https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sql/virtual-machines-windows-migrate-sql

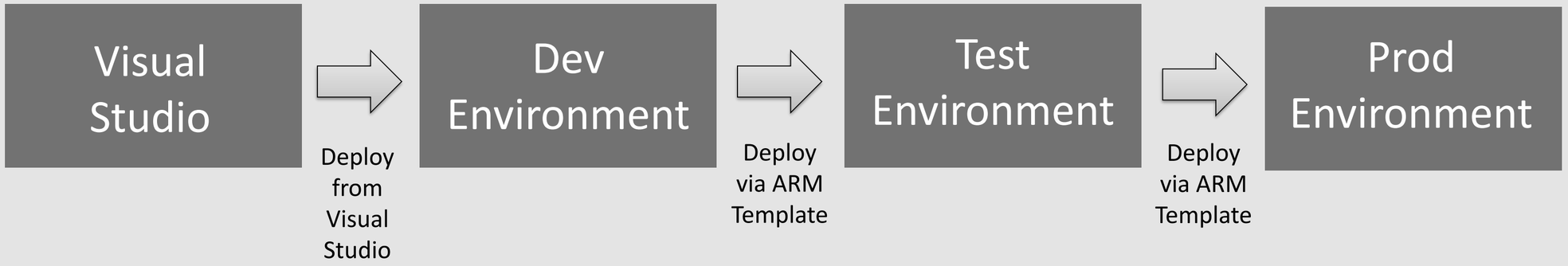
Azure SQL Data Warehouse migration utility:
https://docs.microsoft.com/en-us/azure/sql-data-warehouse/sql-data-warehouse-migrate-migration-utility

Structuring
Dev, Test & Prod
Environments

Separation of Dev, Test, Prod Environments

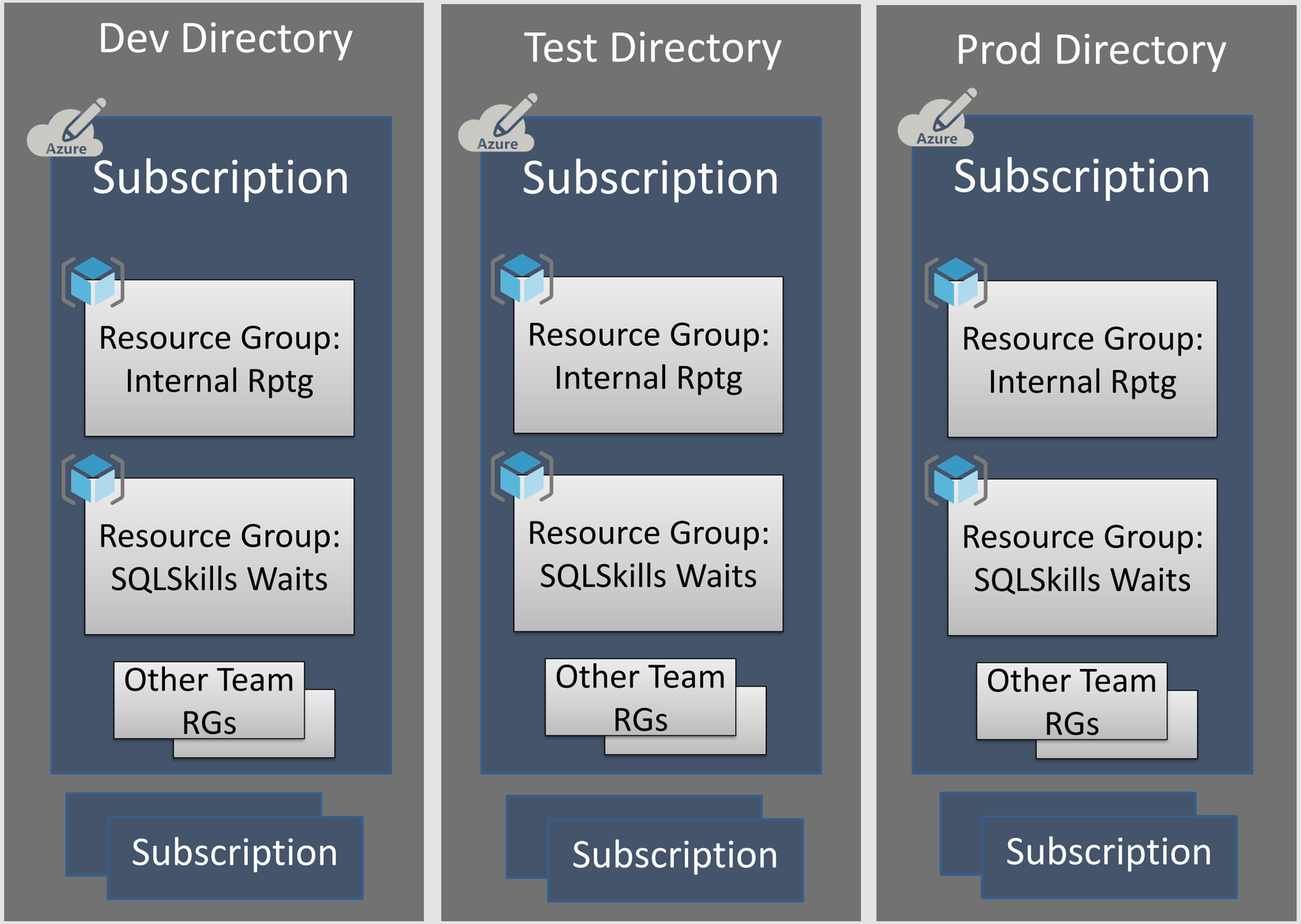
Most commonly environments are segregated by:

1. Resource Groups,
2. Subscription, or
3. Directory, or
4. A combination of 1 and 3, or 2 and 3



For large or multi-tenant implementations: be aware of Azure limits before deciding.

Option: Separate By Directory



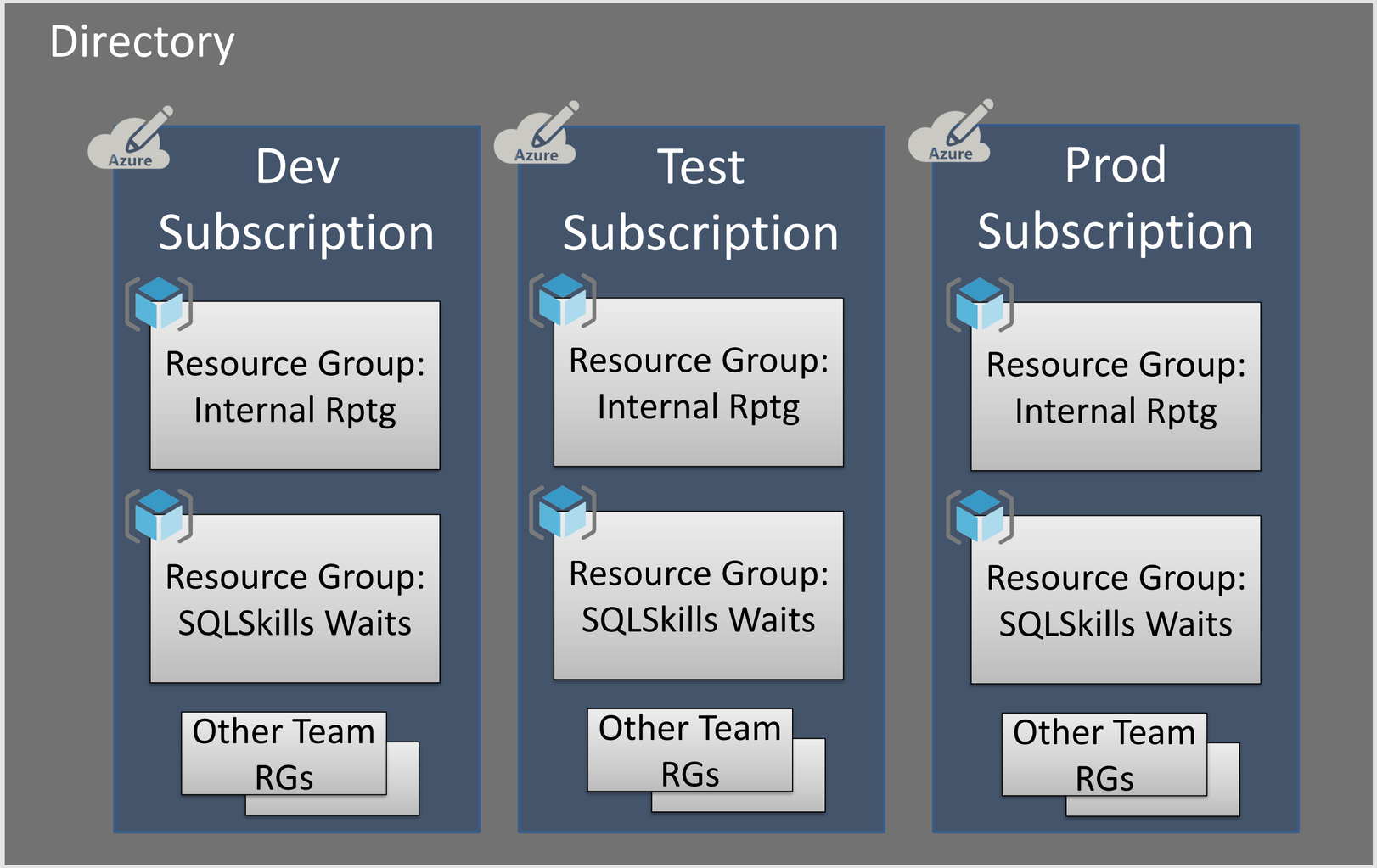
Pros:

- ✓ Clear boundary
- ✓ Offers the most scalability

Cons:

- ✓ More infrastructure to manage
- ✓ A lot of objects intermixed in a subscription - need clear resource group names and/or tags to tell what belongs to which team

Option: Separate By Subscription



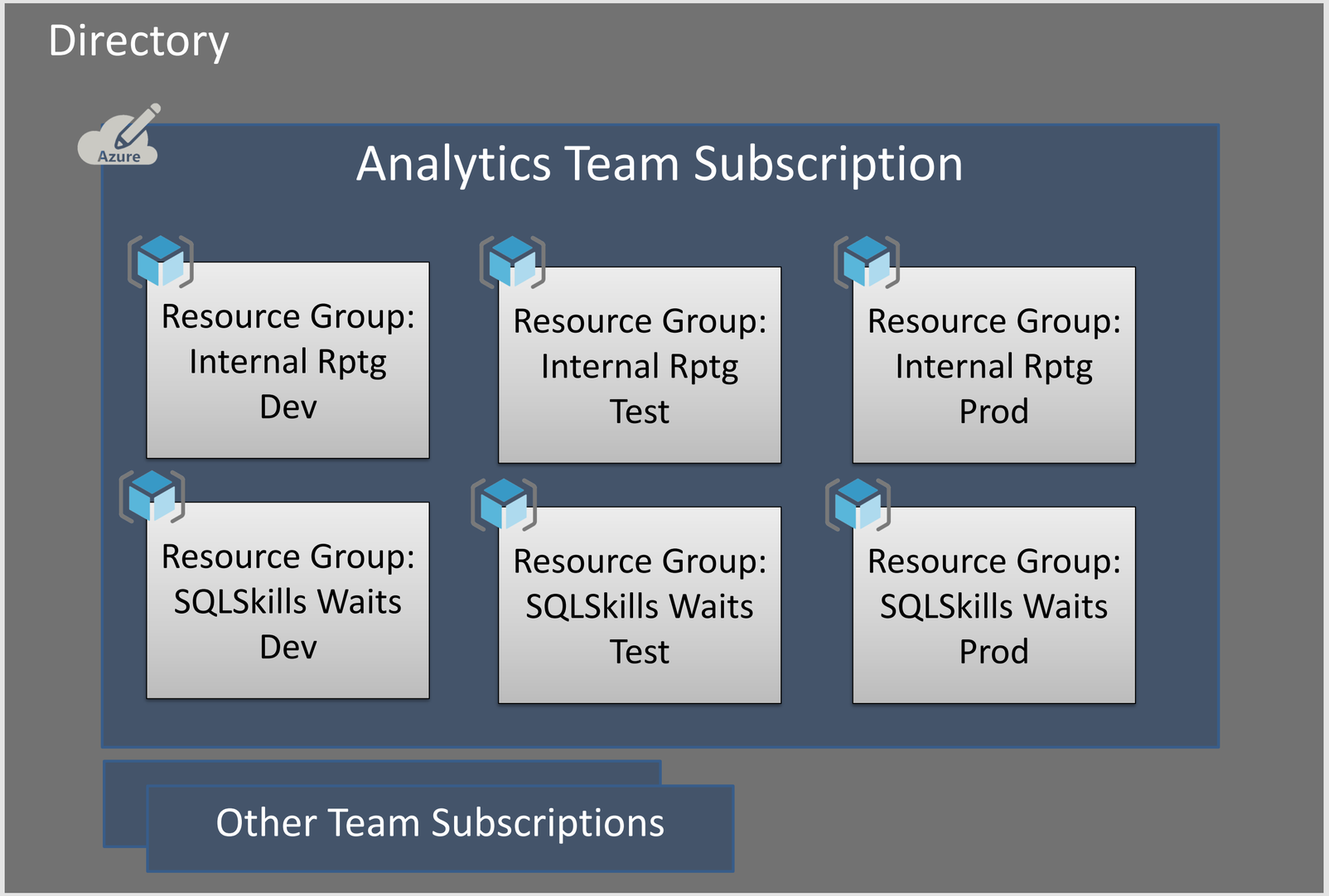
Pros:

- ✓ Clear boundary

Cons:

- ✓ Cannot always provision new resources if you're just an owner at the resource group level
- ✓ Separate virtual networks & VPN set up
- ✓ A lot of objects intermixed in a subscription - need clear resource group names and/or tags to tell what belongs to which team

Option: Separate by Resource Group



We are currently using this option.

Pros:

- ✓ Isolation of subscriptions by the team who owns/manages
- ✓ Set up of virtual network & VPN just once
- ✓ Co-admin privileges easier to delegate at the subscription level for each team
- ✓ Billing segregation
- ✓ Less likely to hit Azure limits

Cons:

- ✓ More risk because Prod is mixed in with Dev & Test (mitigate w/ diff IDs to access Prod resources)

Naming Conventions & Tags

Naming Conventions

Purpose → Type of Service → Environment

- ✓ Type of service in the name helps with logging/metrics in monitoring scenarios
- ✓ Environment as the suffix makes any concatenations easy within scripts
- ✓ Prod is enumerated because we work within one subscription
- ✓ No dashes since all services don't allow them
- ✓ Camel case if the service allows it; otherwise lower case

Not our "real" object names but close enough

Resource Groups

InternalReportingRGDev
InternalReportingRGTest
InternalReportingRGProd

Virtual Machines

BISQLVM1Dev
BISQLVM1Test
BISQLVM1Prod

Storage Accounts

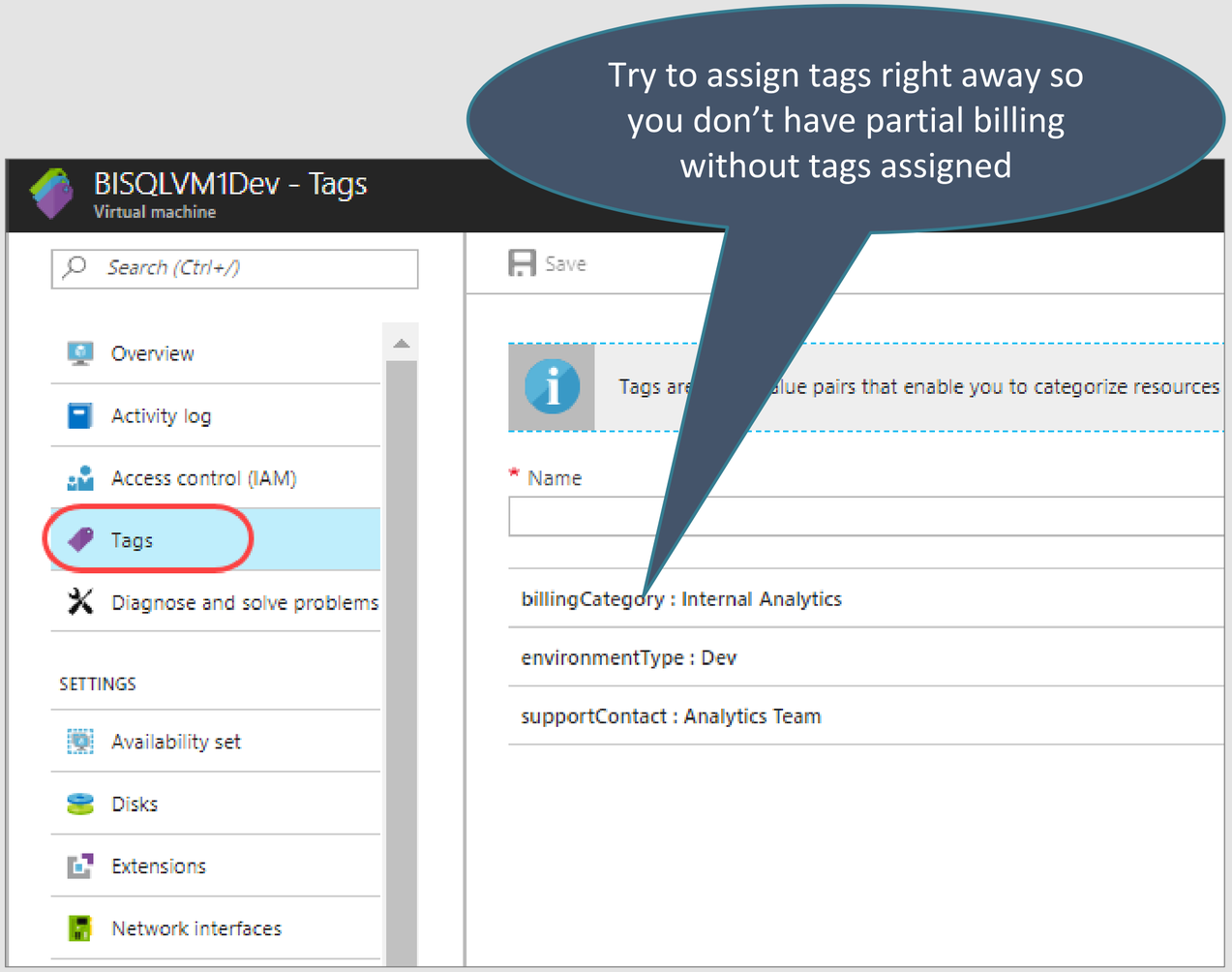
BISQLVM1DataStdStrgDev
BISQLVM1BckStdStrgDev
BISQLVM1DiagStdStrgDev

More info on naming conventions: http://www.sqlchick.com/entries/2017/6/24/naming-conventions-in-azure

Standard Tags

Tags are very helpful for:

- ✓ Billing or cost center categories
- ✓ Environment names
- ✓ Project or system
- ✓ Purpose or application
- ✓ Team, department, business unit
- ✓ Who owns or supports a resource
- ✓ Release or version #s (ex: testing infrastructure)
- ✓ Archival date (ex: if needed only temporarily)
- ✓ Patching or maintenance window or SLA
- ✓ Which customer it applies to (ex: if an ISV)
- ✓ etc…



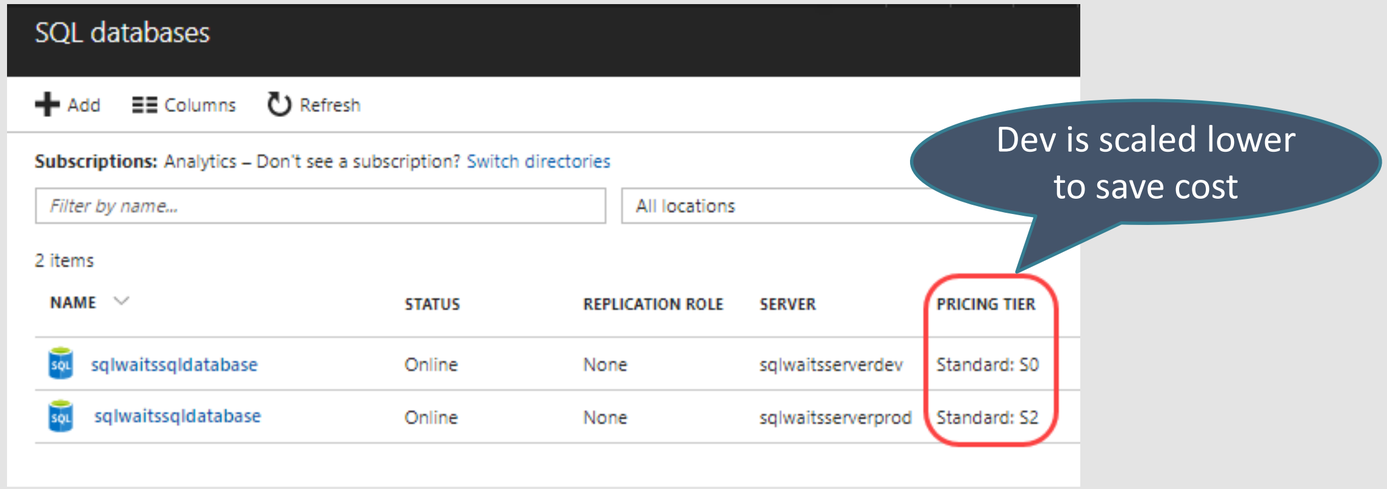
Meter Name	Consumed Service	Resource Group	Tags
"Standard IO - Block Blob Read Operation Units (in 10,000s)"	"Microsoft.Storage"	"internalreportingrgdev"	"{"billingCategory":"Internal Analytics","environmentType":"Dev","supportContact":"Analytics Team"}"
"Standard IO - Table Write Operation Units (in 10,000s)"	"Microsoft.Storage"	"internalreportingrgdev"	"{"billingCategory":"Internal Analytics","environmentType":"Dev","supportContact":"Analytics Team"}"
"Standard IO - Page Blob/Disk (GB)"	"Microsoft.Storage"	"internalreportingrgdev"	"{"billingCategory":"Internal Analytics","environmentType":"Dev","supportContact":"Analytics Team"}"
"Standard IO - Table (GB)"	"Microsoft.Storage"	"internalreportingrgdev"	"{"billingCategory":"Internal Analytics","environmentType":"Dev","supportContact":"Analytics Team"}"

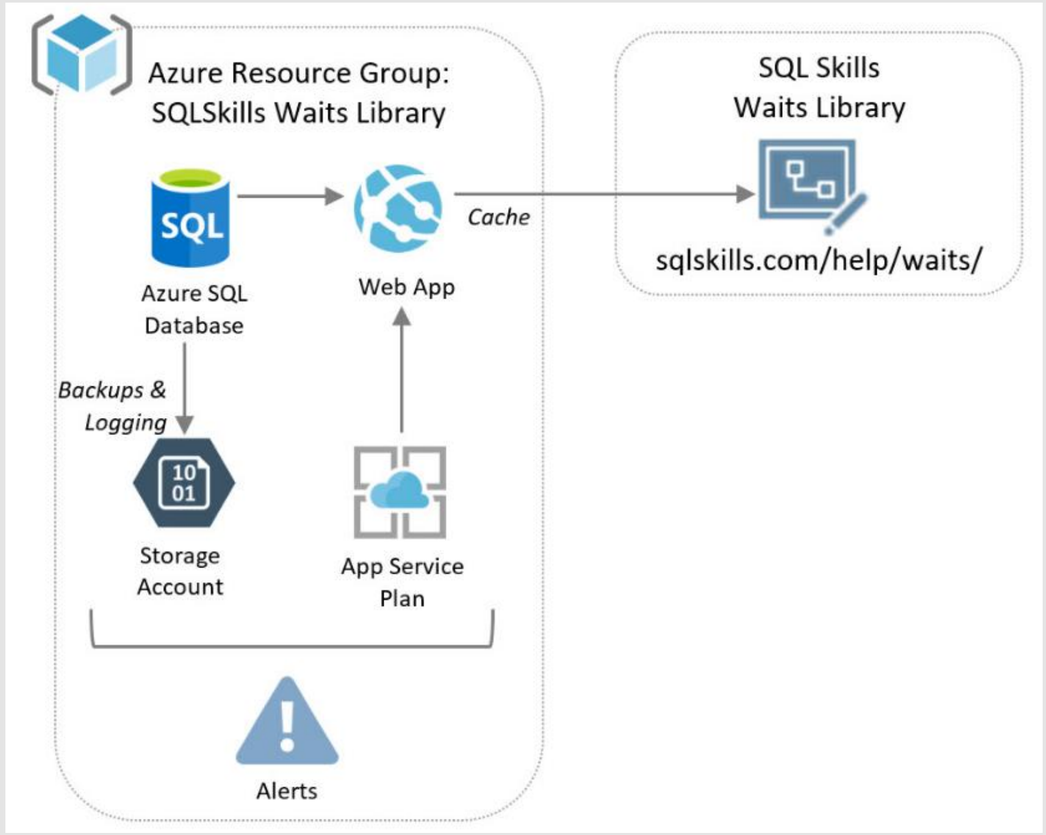
More info on tags: http://www.sqlchick.com/entries/2017/7/15/powershell-for-assigning-querying-tags-in-azure

Configuration Decisions: Azure SQL Database

Use of Azure SQLDB

We're using Azure SQL Database for a new solution just created recently.





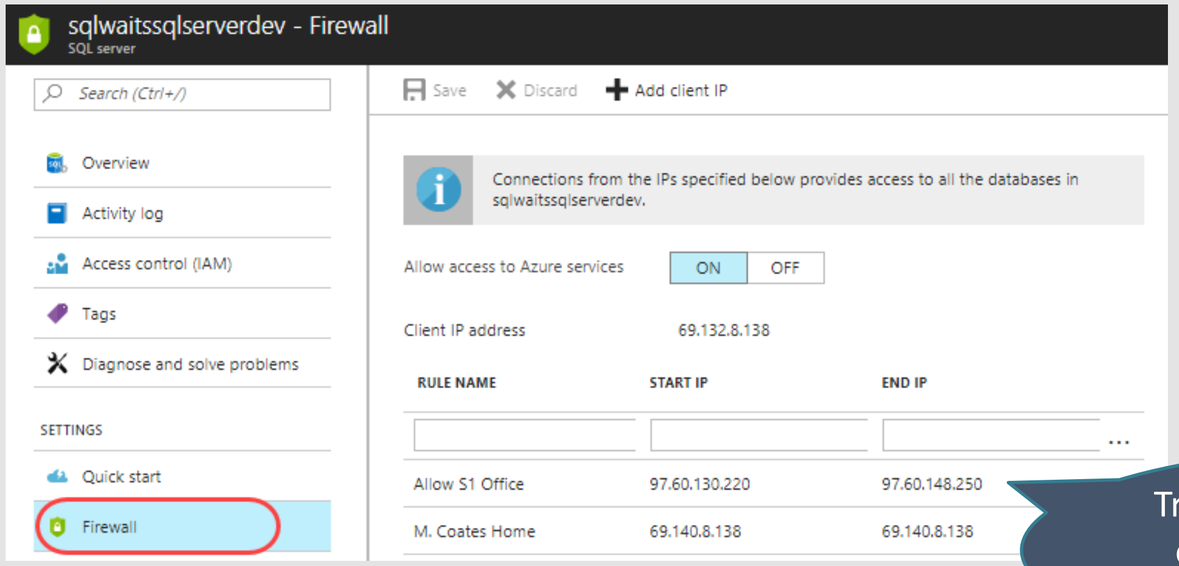
Our solution contains aggregate, anonymous data.

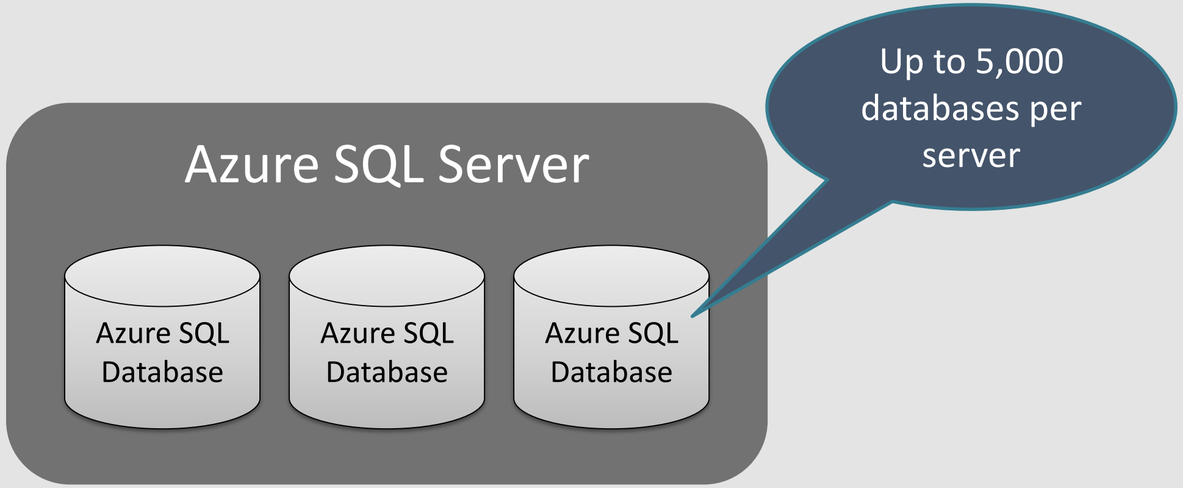
There is no Virtual Network/VPN functionality for SQLDB or SQLDW yet (coming in "Managed Instances" of Azure SQLDB).

Firewall for Azure SQLDB

A firewall can be set at the server level, or at the database level.

Server level firewall rules can be set in the portal:



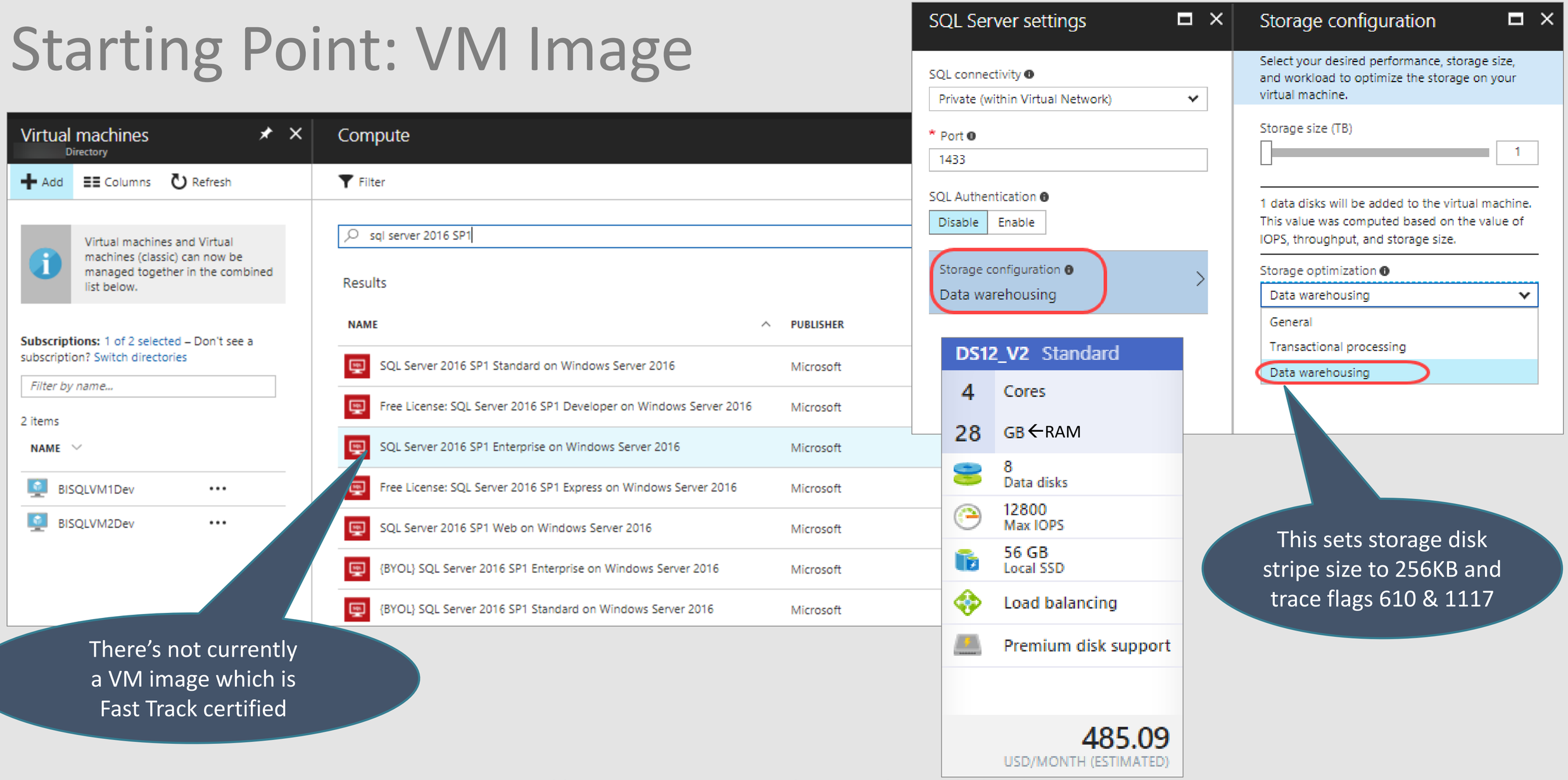


Database level firewall rules must be set with T-SQL. Although that makes them less visible, it is a better practice.

Try to assign a name to each entry. The generic default name isn't helpful.

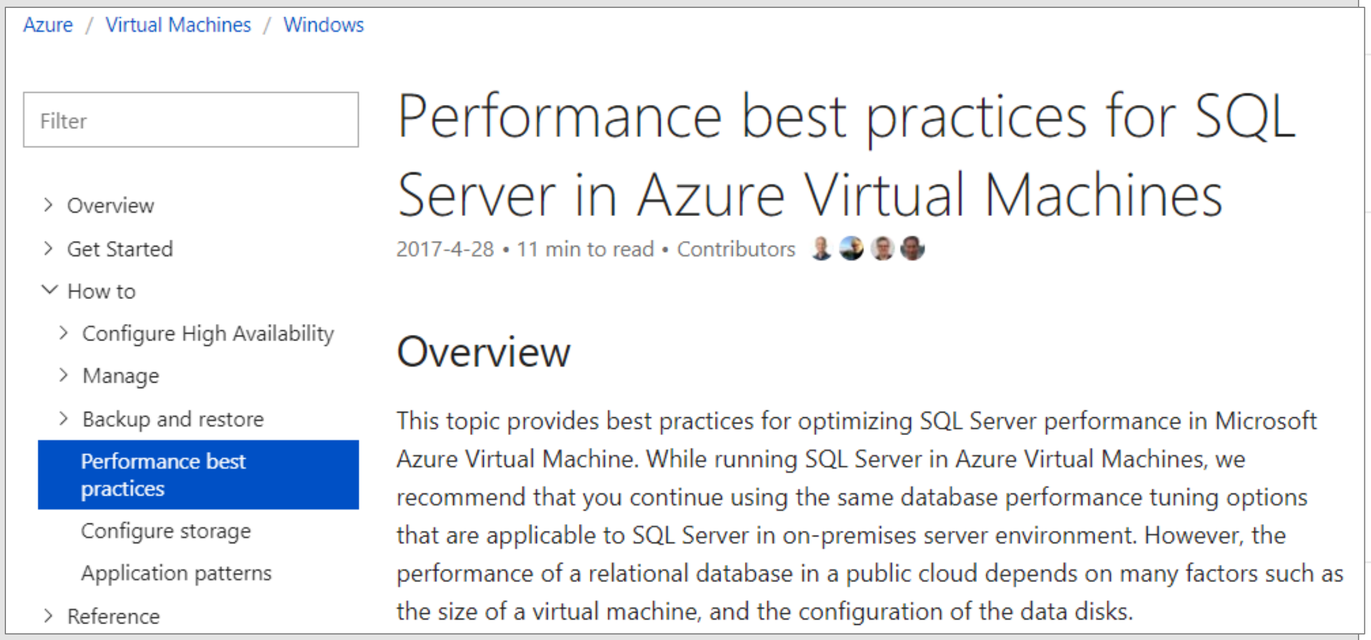
Configuration Decisions: SQL Server in Azure Virtual Machine

Starting Point: VM Image



If the VM scale you want isn't available: http://www.sqlchick.com/entries/2017/3/30/why-some-azure-vm-sizes-are-unavailable-when-resizing-in-the-portal

Best Practices Guidance



Area	Optimizations
VM size	DS3 or higher for SQL Enterprise edition. DS2 or higher for SQL Standard and Web editions.
Storage	Use Premium Storage. Standard storage is only recommended for dev/test. Keep the storage account and SQL Server VM in the same region. Disable Azure geo-redundant storage (geo-replication) on the storage account.
Disks	Use a minimum of 2 P30 disks (1 for log files; 1 for data files and TempDB). Avoid using operating system or temporary disks for database storage or logging. Enable read caching on the disk(s) hosting the data files and TempDB. Do not enable caching on disk(s) hosting the log file. Important: Stop the SQL Server service when changing the cache settings for an Azure VM disk. Stripe multiple Azure data disks to get increased IO throughput. Format with documented allocation sizes.
I/O	Enable database page compression. Enable instant file initialization for data files. Limit or disable autogrow on the database. Disable autoshrink on the database. Move all databases to data disks, including system databases. Move SQL Server error log and trace file directories to data disks. Setup default backup and database file locations. Enable locked pages. Apply SQL Server performance fixes.
Feature specific	Back up directly to blob storage.

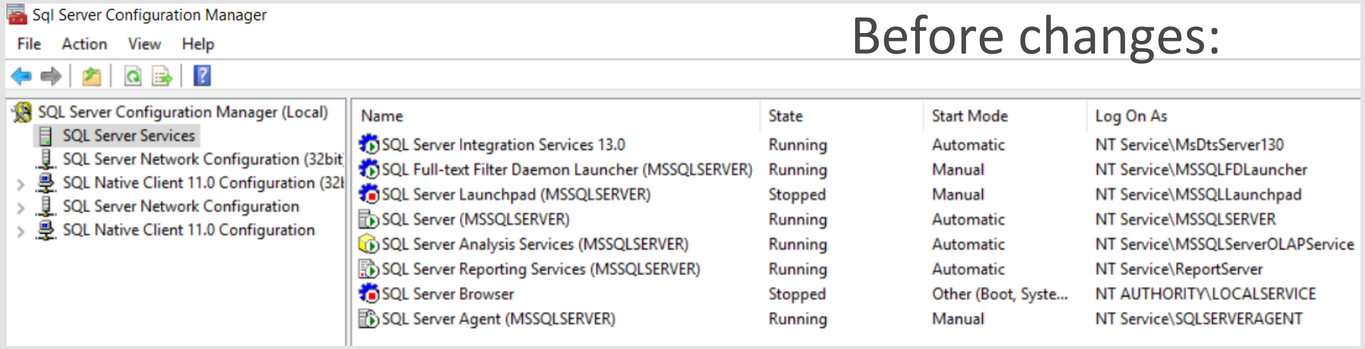
Most of the guidance in this Performance Best Practices article is implemented in the pre-built image.
https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sql/virtual-machines-windows-sql-performance

Changes Made to Pre-Built VM Image

We started with a pre-built image, and made customizations after it was domain-joined, such as:

- Domain service accounts for each service
- Disable unused services
- Disk structure + permissions
  - G: Data
  - L: Logs
  - T: TempDB
- Enable disk encryption
- Policies (volume maintenance, etc)

Before changes:



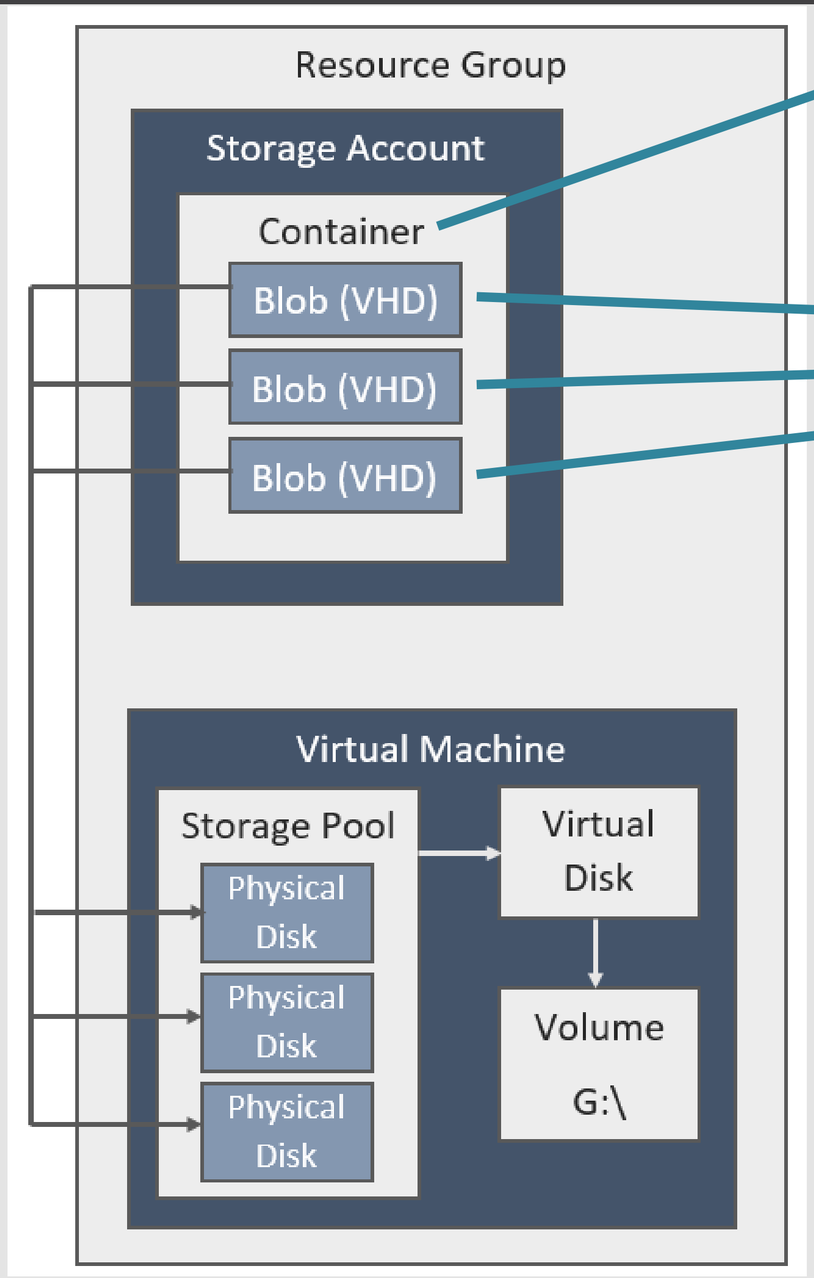
SQL Server changes for each new VM:

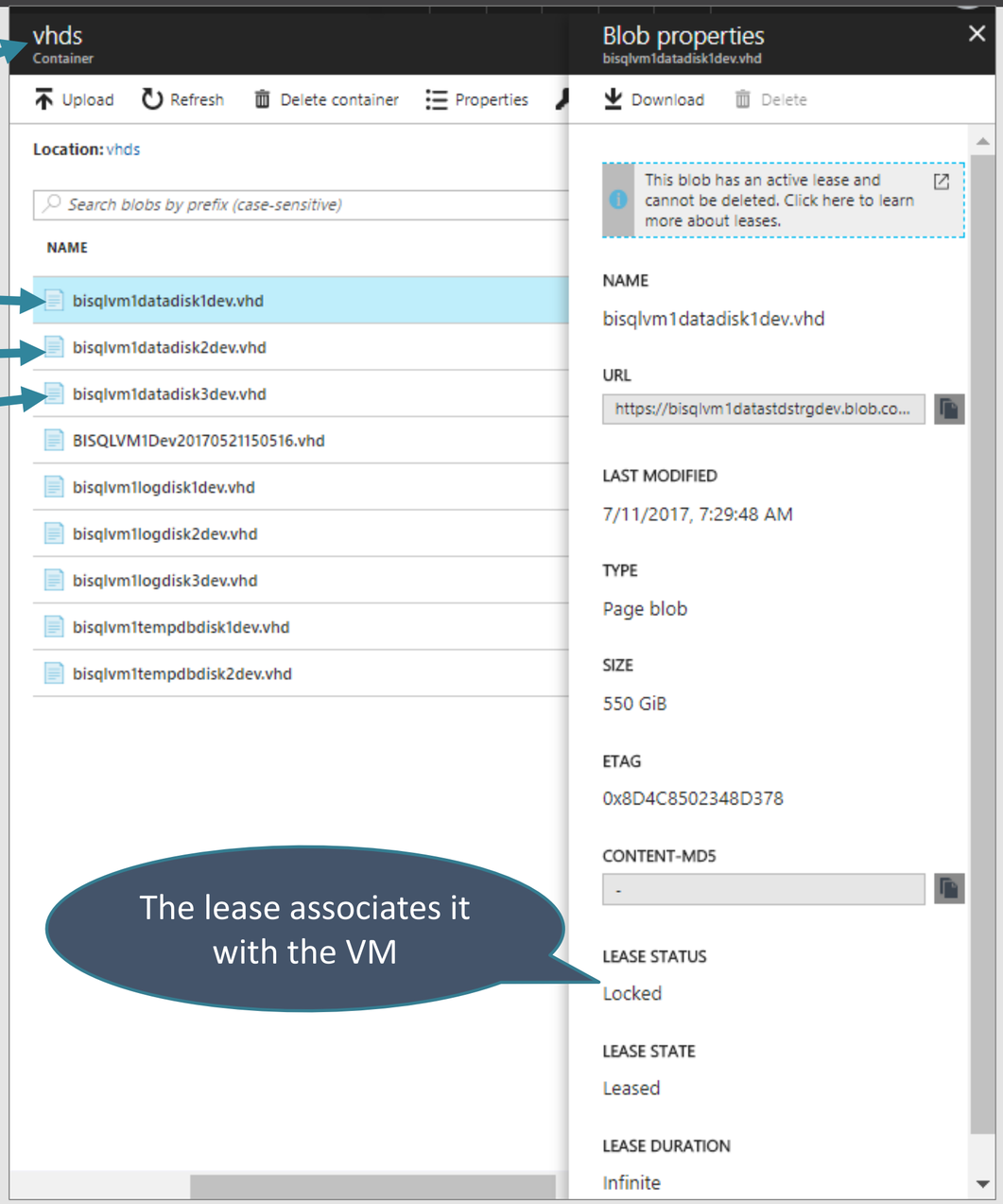
- Enable the remote DAC
- Default database file locations
- Fill factor settings at database level
- Optimize for ad hoc workloads
- Max degree of parallelism
- Cost threshold for parallelism
- Startup parameters & trace flags
- Expand # of TempDB files & relocate
- Time of SSIS maintenance job
- Default for backup checksum
- Limited account to act as DBOwner

  etc…

Storage

The VHDs (virtual hard disks) behind a VM are in Azure Storage.





Storage

Backups (“bck”), Data, and Diagnostic logging (“diag”) are separated

All data for a single VM is in the same storage account – it is a unit of recovery in case of failure

Associated with Azure VMs (IaaS)

Associated with Azure SQLDB (PaaS)

NAME	RESOURCE GROUP	LOCATION	SKU	BILLINGCATEGORY (TAG)	ENVIRONMENTTYPE (TAG)	SUPPORTCONTACT (TAG)
INTERNALREPORTINGRGDEV						
bisqlvm1bckstdstrgdev	InternalReportingRGDev	East US	Standard_RAGRS	Internal Analytics	Dev	BI & Analytics Team
bisqlvm1datastdstrgdev	InternalReportingRGDev	East US	Standard_LRS	Internal Analytics	Dev	BI & Analytics Team
bisqlvm1diagstdstrgdev	InternalReportingRGDev	East US	Standard_LRS	Internal Analytics	Dev	BI & Analytics Team
SQLSKILLSWAITSLIBRARYRGDEV						
sqlwaitsbckstdstrgdev	SQLSkillsWaitsLibraryRGDev	East US	Standard_RAGRS	SQLSkills Waits Library	Dev	BI & Analytics Team
sqlwaitsdiagstdstrgdev	SQLSkillsWaitsLibraryRGDev	East US	Standard_LRS	SQLSkills Waits Li…	Dev	BI & Analytics Team

Geo-redundancy (GRS) for backup files

Managed/Unmanaged

We chose to use unmanaged storage as shown above (managed was released to preview during our implementation). Managed VM storage doesn’t display in the portal.

Premium/Standard

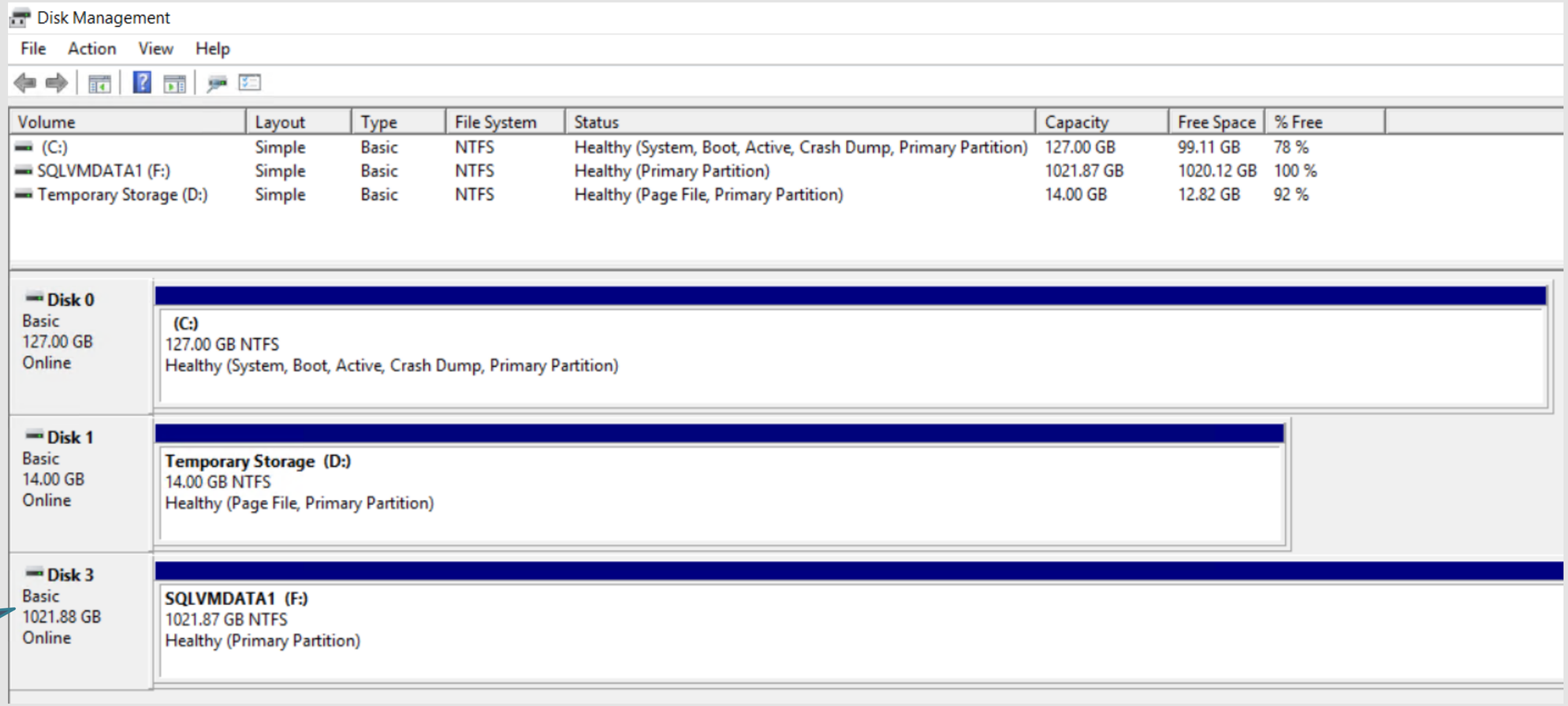
We are using standard storage for VMs in Dev and Test, and Premium in Production.

VM Disks

Default disks provisioned by the pre-built VM image.

We omitted the F: drive (after moving TempDB to its proper drive).

The default data disk size is 1TB



To release the lease on the storage blob:

1. Delete the volume, and
2. Detach disk from the VM

Don’t forget to delete the file in Azure Storage as well.

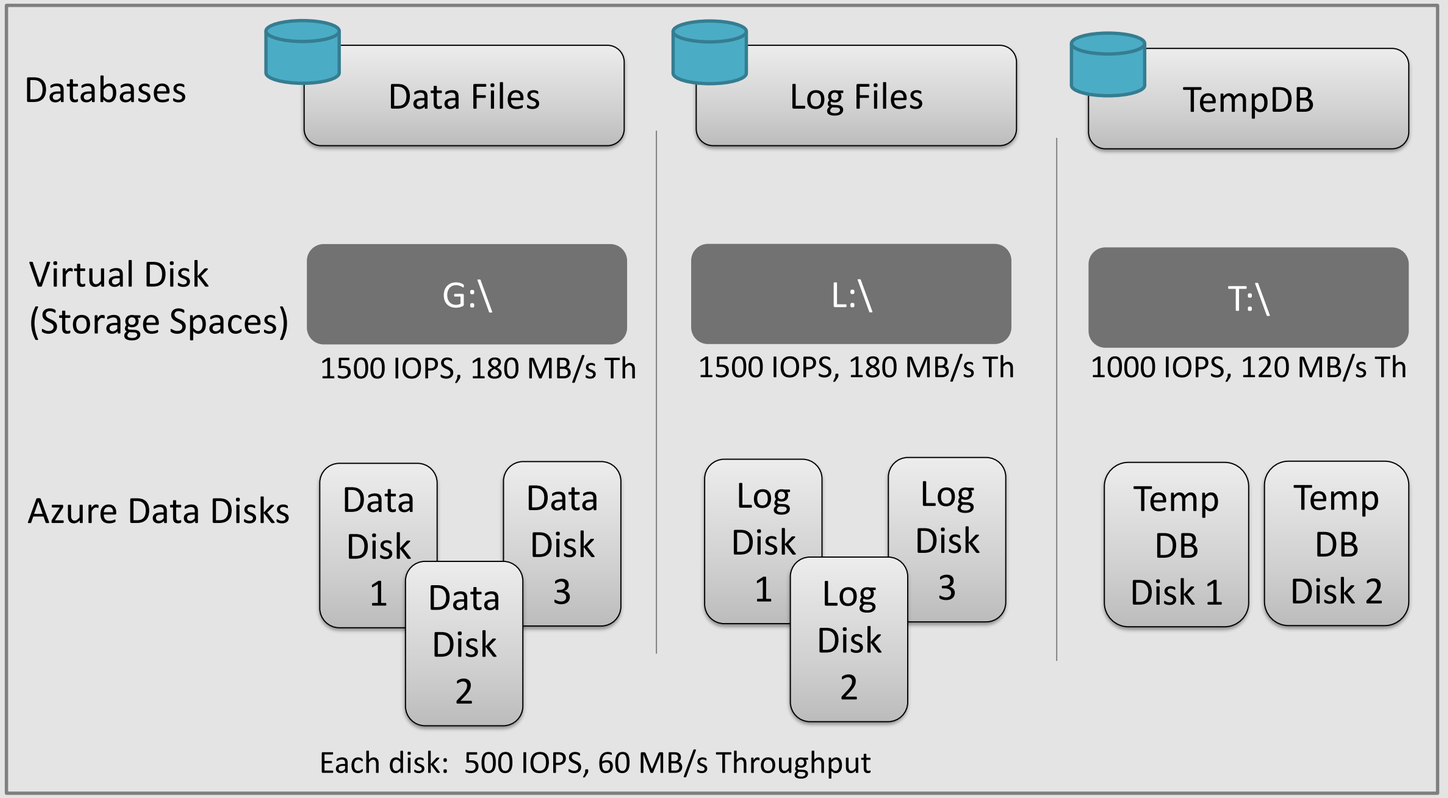
VM Disks

We are using Storage Spaces (aka Virtual Disks): multiple Azure data disks in a storage pool to collectively share throughput and IOPS, limits.

This aggregation of disks is helpful due to lower Standard limits on IOPS & throughput.

(Another alternative to increase disk performance is to spread separate database files on different disks.)

IOPS = # of requests per second (OLTP)
Throughput (Bandwidth) = IOPS x I/O Size per specified interval (DW)



Adapted from:
https://blogs.msdn.microsoft.com/sqlcat/2013/06/17/performance-guidance-for-sql-server-in-windows-azure-virtual-machines/

VM Disks

Step 1: Check max # of disks which can be attached to the VM

Step 2: Add disks to VM (Azure portal or PowerShell)

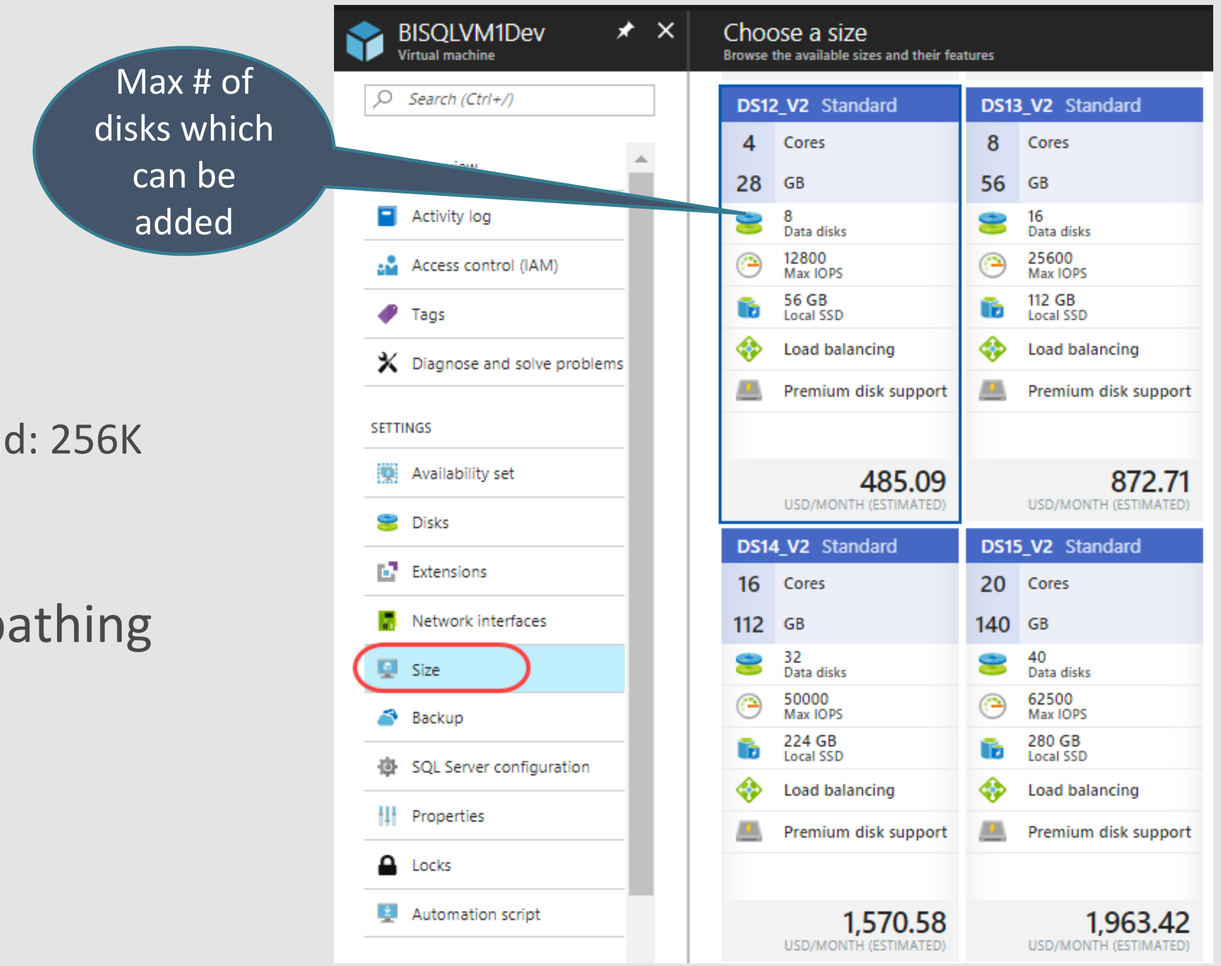
Step 3: Add storage pool with disk striping

Stripe size (virtual disk interleave) for DW workload: 256K

Volume allocation unit size: 64K

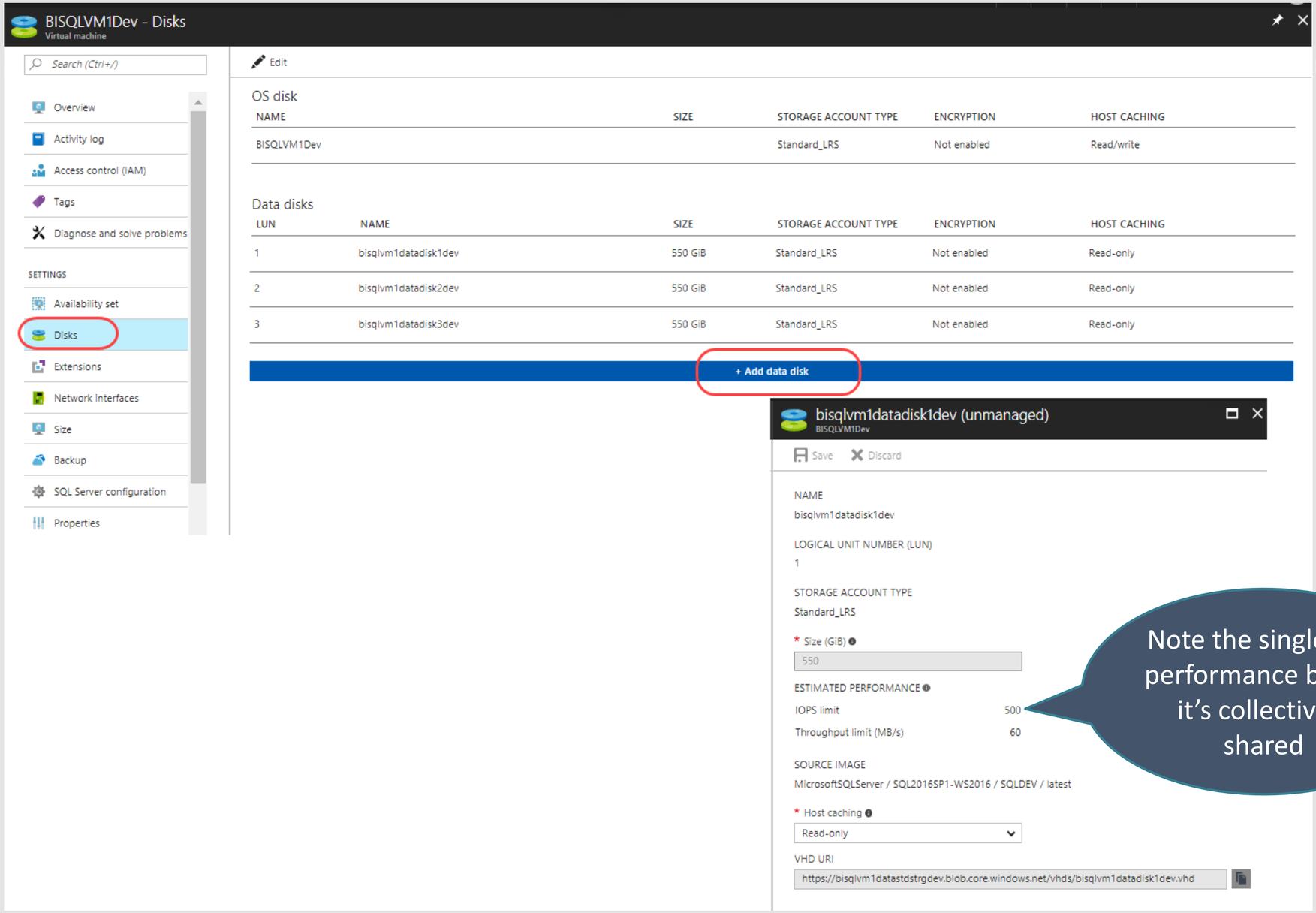
Step 4: Create folders + reassign SQL Server pathing to your preference

Step 1



VM Disks (4/8)

Step 2: Add disks to VM (Azure portal or PowerShell)

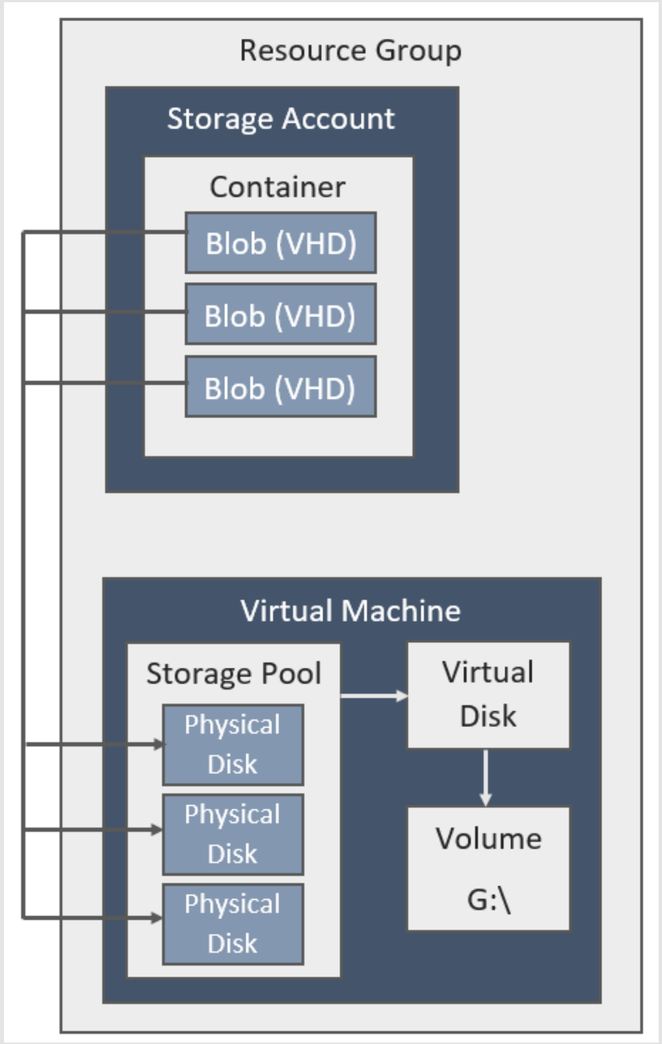


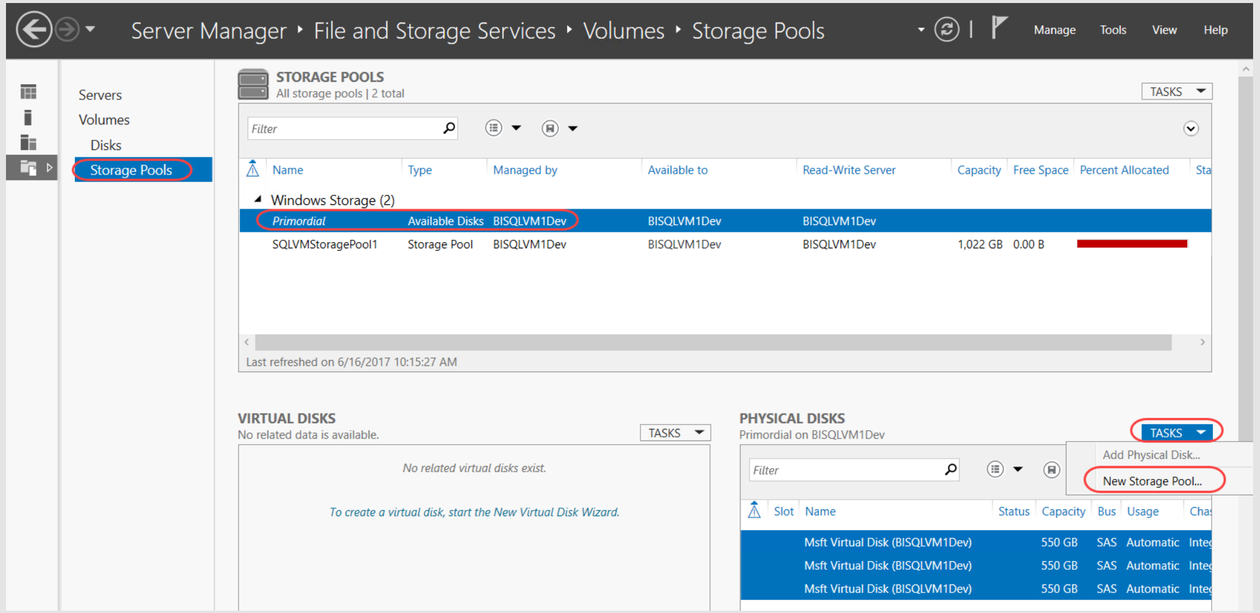
VM Disks

Step 3: Add storage pool + virtual disk + volume in the VM

Step 4: Create folders + reassign SQL Server pathing

- G:\MSSQL\Data
- G:\MSSQL\Backup
- G:\OLAP\Data
- G:\OLAP\Backup
- L:\MSSQL\ErrorLog
- L:\MSSQL\Log
- L:\MSSQL\Dump
- L:\OLAP\Log
- L:\OLAP\Dump
- T:\MSSQL\Data
- T:\OLAP\Temp



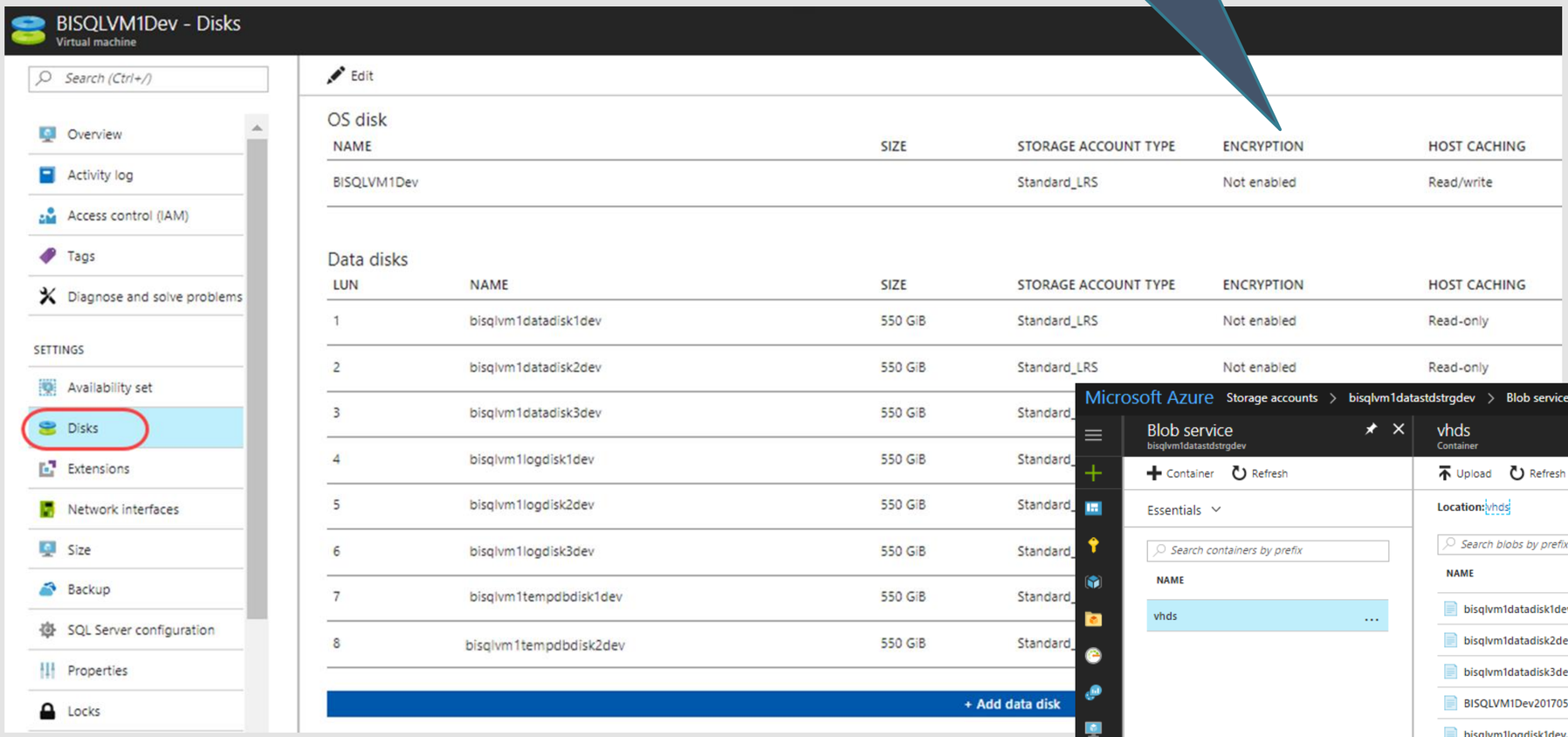


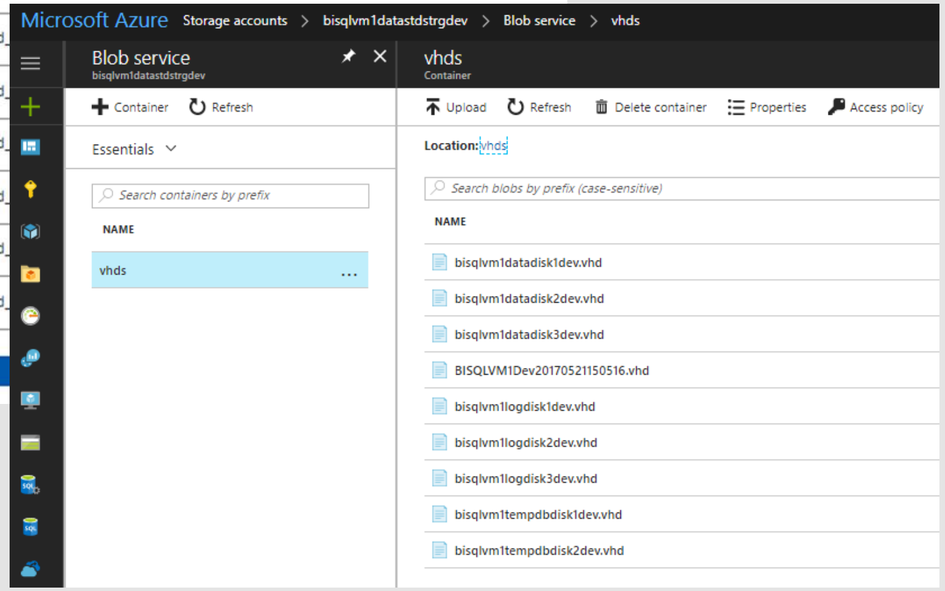
VM Disks

(6/8)

We're going to handle encryption next

End Result: 8 data disks attached to the Azure VM



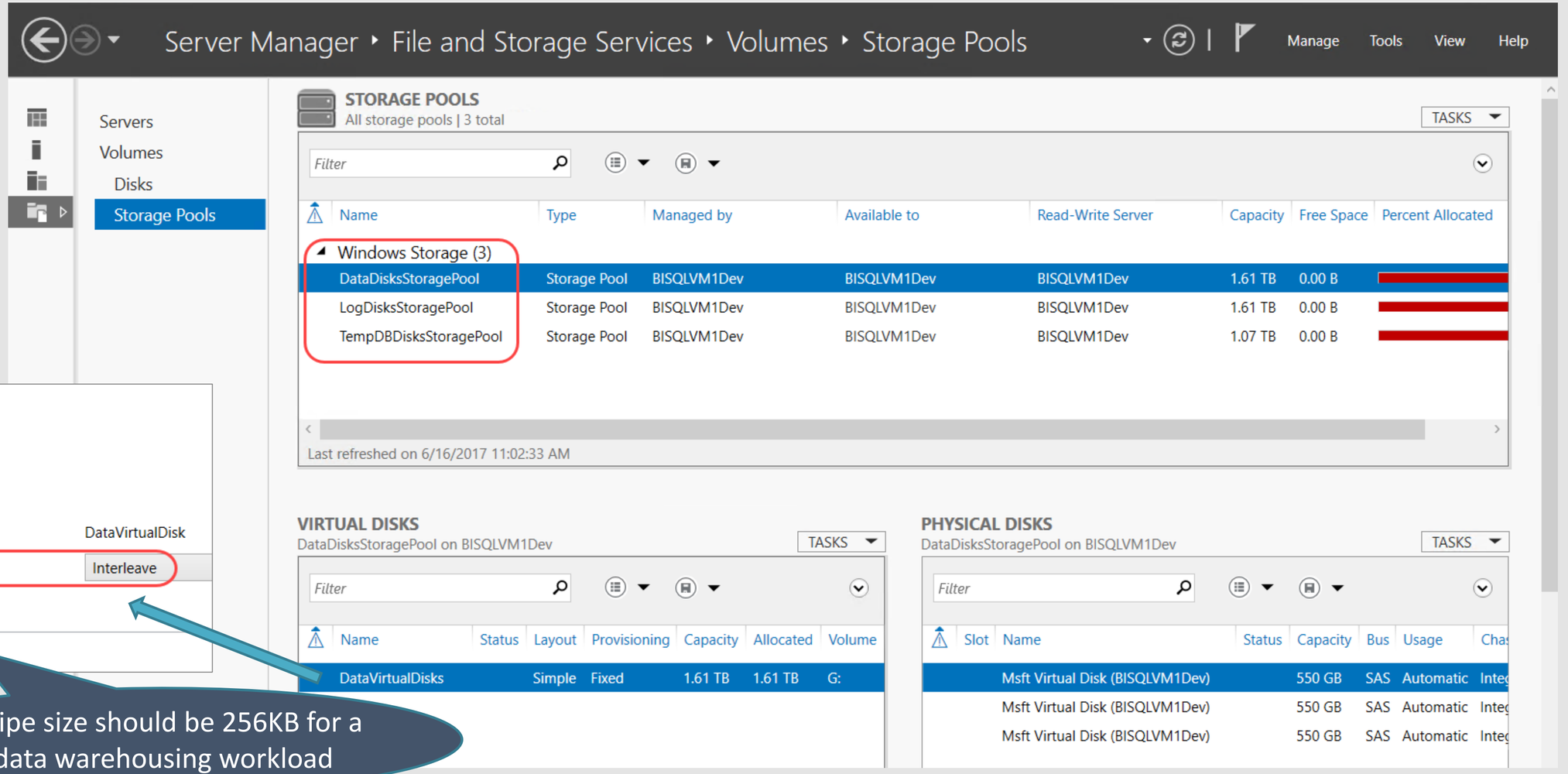


Disks attached to the VM

Blob storage

VM Disks

End Result:
3 storage pools

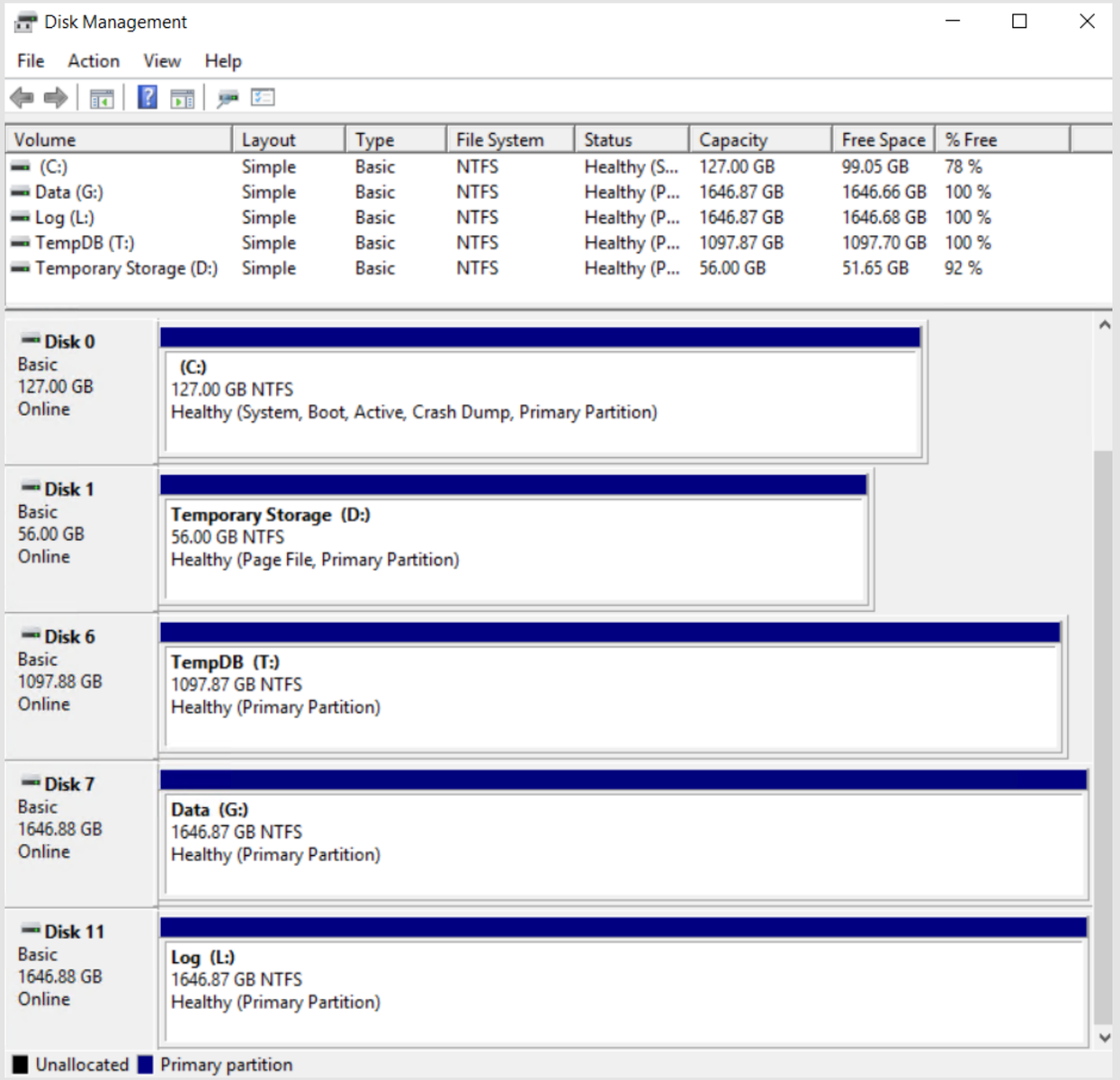


VM Disks

End Result:

3 volumes

- G:\ - Data
- L:\ - Log
- T:\ - TempDB



Encryption – Data at Rest

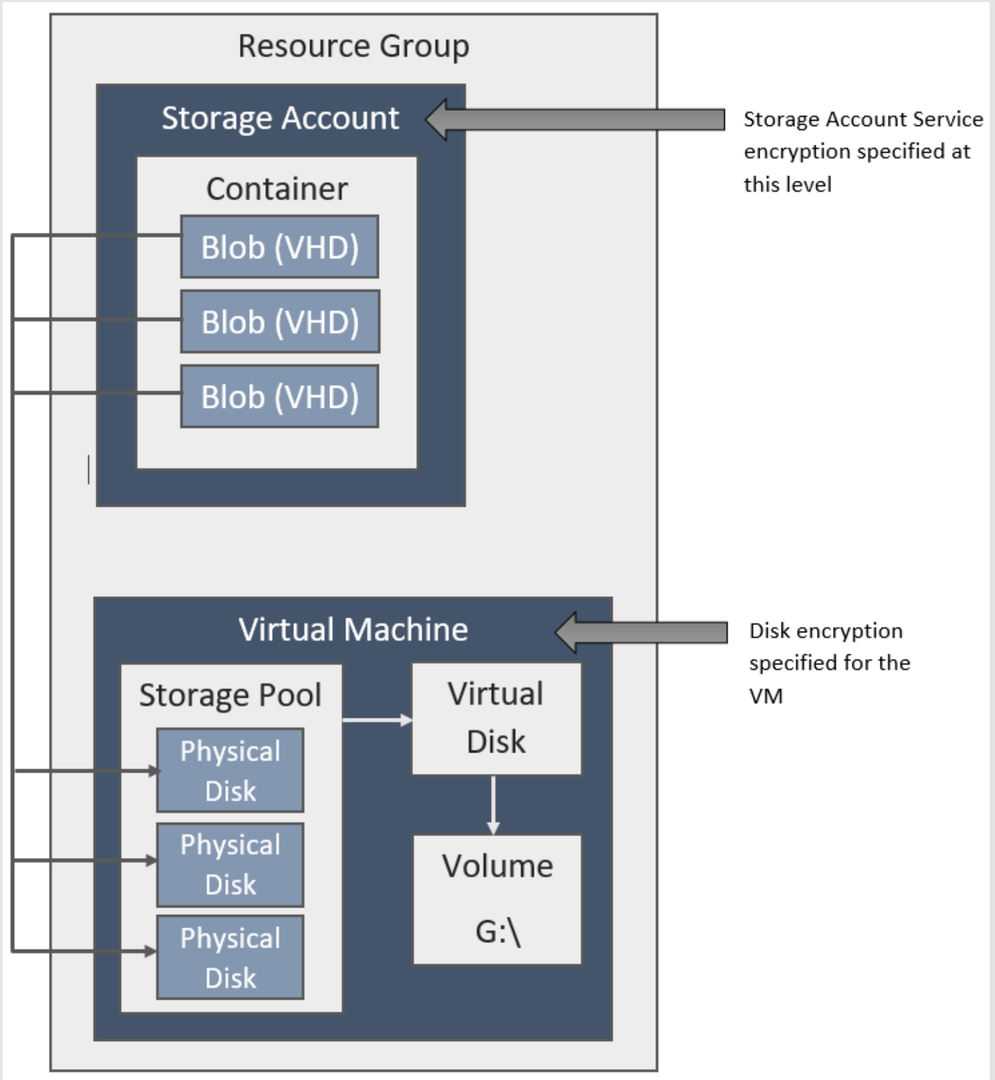
Storage Service Encryption (SSE) ← associated with the storage account

Disk Encryption (BitLocker) ← inside the VM

More info:
http://www.sqlchick.com/entries/2017/2/5/deciding-on-encryption-at-rest-for-an-azure-virtual-machine

http://www.sqlchick.com/entries/2017/2/9/setting-up-azure-disk-encryption-for-a-virtual-machine-with-powershell

Other options to consider for encrypting the database:
- TDE (Transparent Data Encryption)
- AE (Always Encrypted)



Disk Encryption (1/2)

Step 1: Create a Service Principal

Step 2: Provision Azure Key Vault

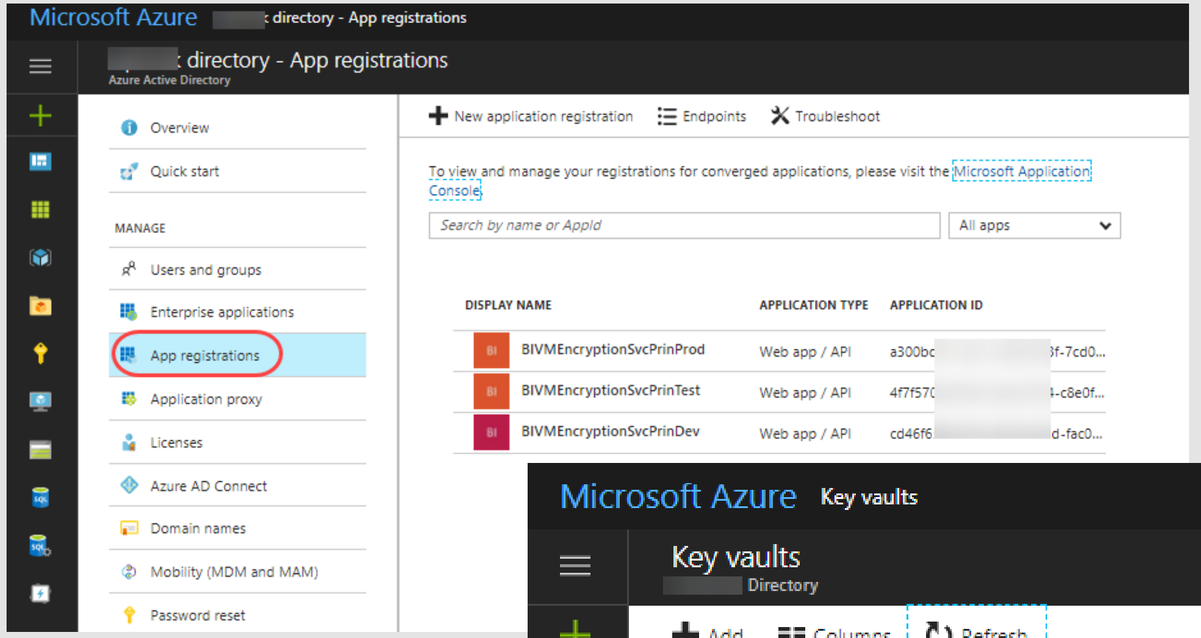
Step 3: Enable the Service Principal to communicate with Key Vault

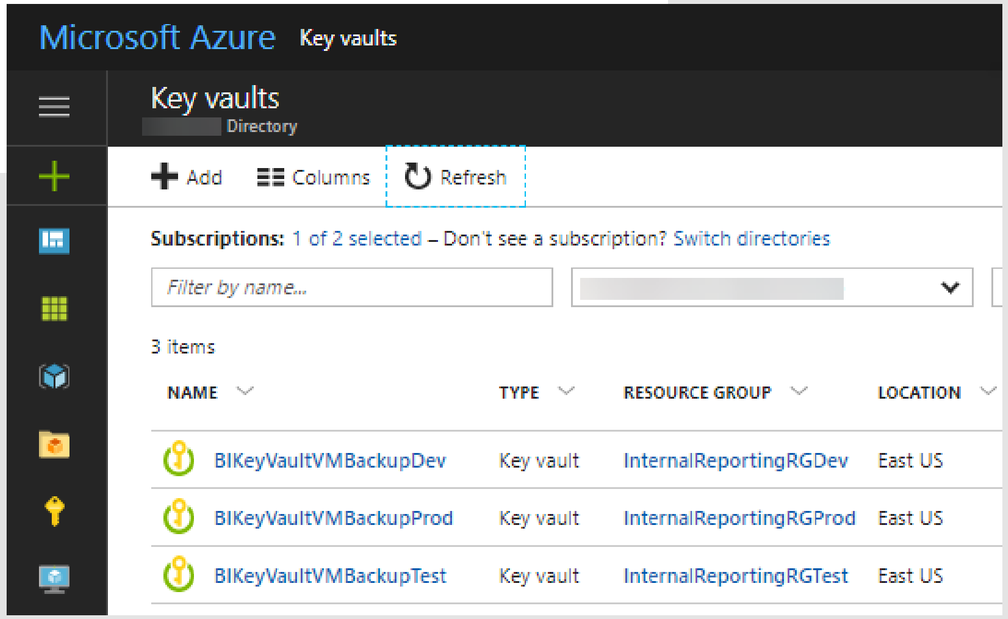
Step 4: Enable Disk Encryption

PowerShell scripts here:

http://www.sqlchick.com/entries/2017/2/5/deciding-on-encryption-at-rest-for-an-azure-virtual-machine

http://www.sqlchick.com/entries/2017/2/9/setting-up-azure-disk-encryption-for-a-virtual-machine-with-powershell

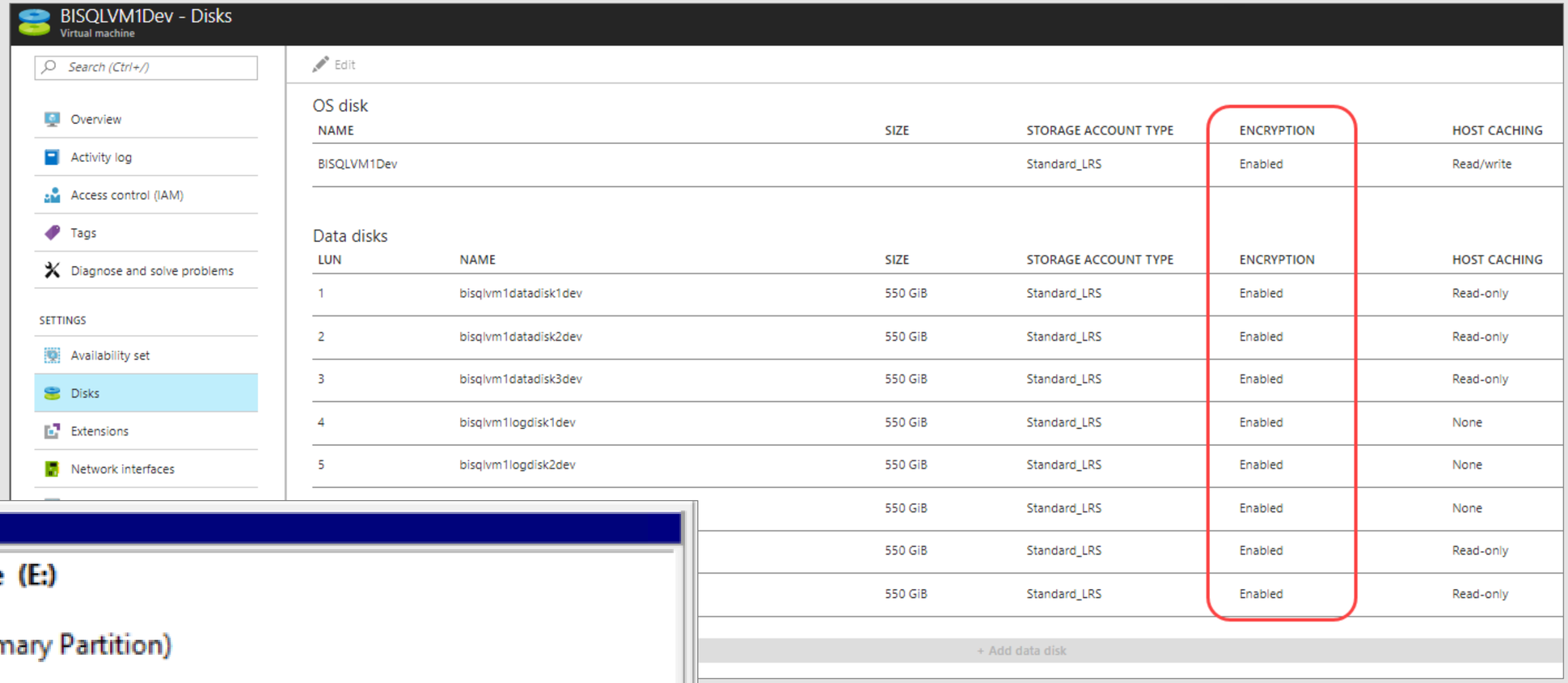


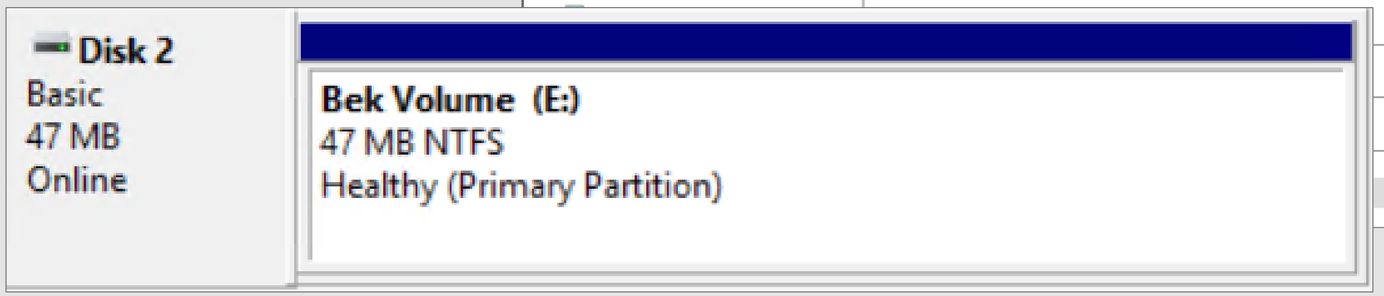


Disk Encryption (2/2)

End Result: All drives are encrypted.

A new Bek volume is created which is where the encryption key is read from:





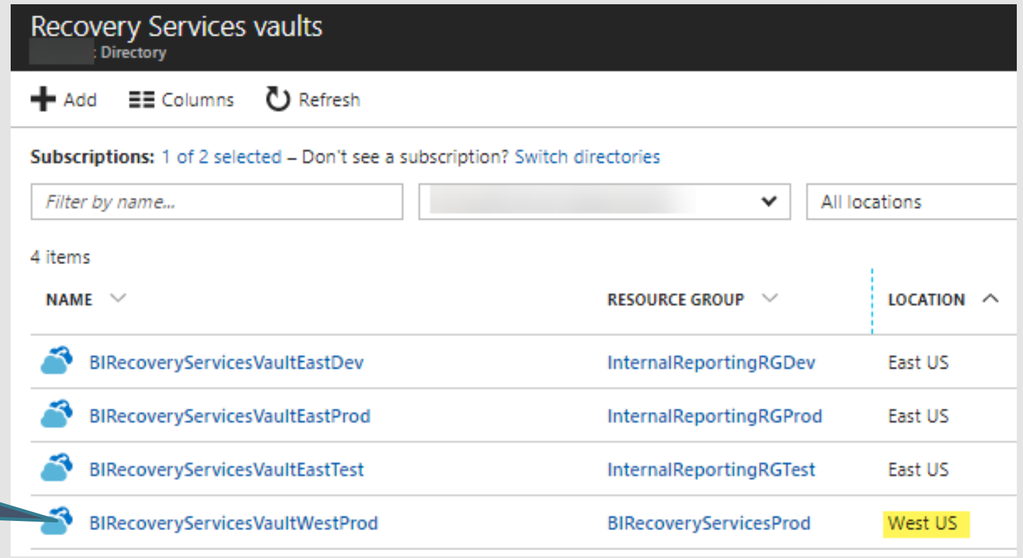
More info: https://blogs.msdn.microsoft.com/mast/2016/11/27/azure-disk-encryption-how-to-recover-bek-file-from-azure-key-vault/

Disaster Recovery

We (the Analytics Team) don't want to incur the cost or complexity of supporting an availability set. Therefore, we are doing the following for DR purposes:

1. Documented steps for re-provisioning & redeploying
2. Use of Premium storage in Production – this gets us a 99.9% uptime SLA for a single instance Azure VM https://azure.microsoft.com/en-us/support/legal/sla/virtual-machines/
3. For Prod: we are considering use of Azure Site Recovery (still in preview) https://docs.microsoft.com/en-us/azure/site-recovery/site-recovery-azure-to-azure

Important to create your Recovery Services vault in a different region (the RG and the vault itself) than the source objects



Automation & Scheduling

Daily & Weekly Schedule

Analytics Environment Scheduling
All times EST

Production:

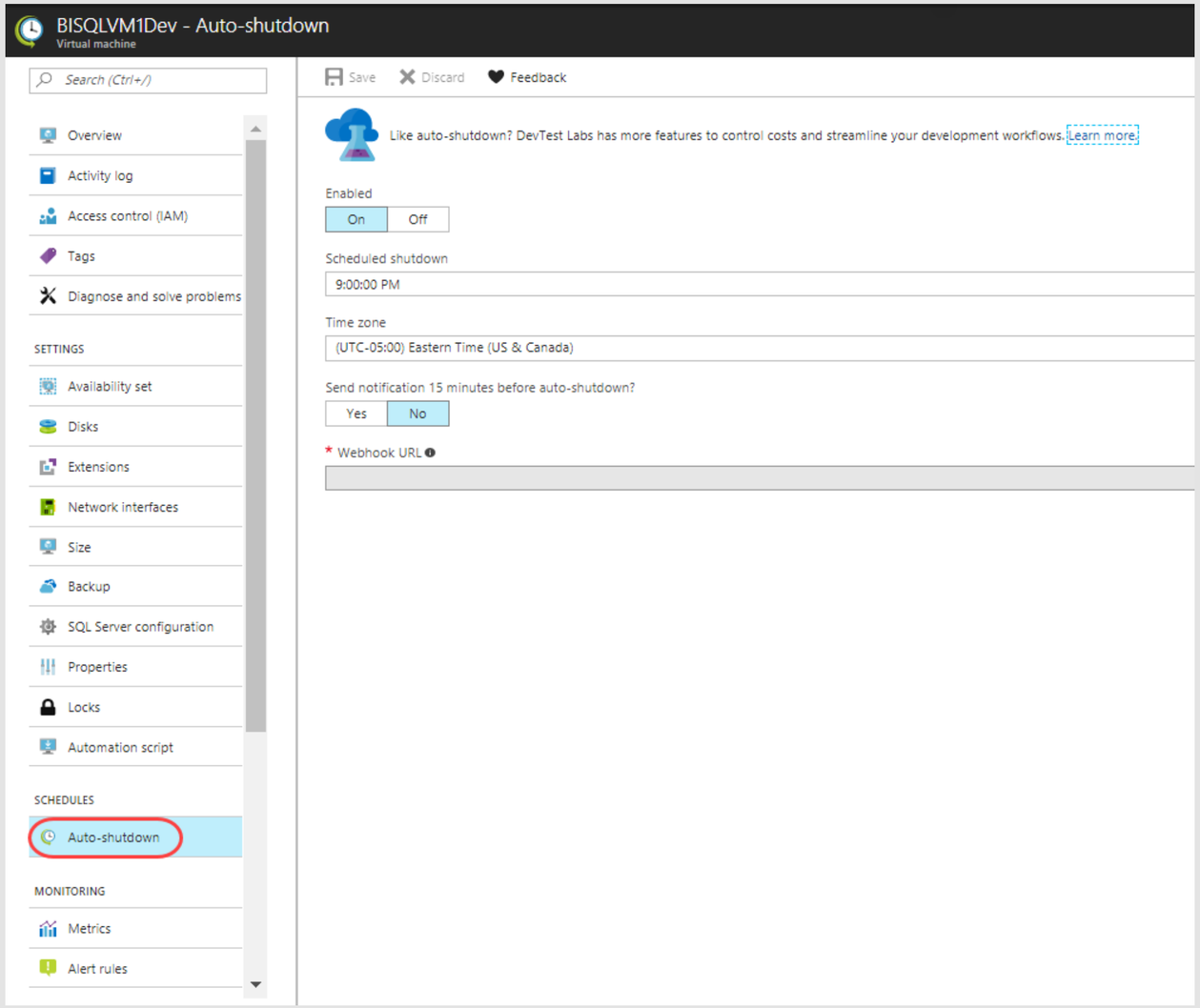
12am	1am	2am	3am	4am	5am	6am	7am	8am	9am	10am	11am	12pm	1pm	2pm	3pm	4pm	5pm	6pm	7pm	8pm	9pm	10pm	11pm
Engine+SSIS VM shut down					Auto start																Engine+SSIS VM shut down		
								EST standard business hours															
			UK standard business hours																				
ADF job						Daily ETL		Hourly ETL & SSAS processing										Auto patching (Fri)	SSIS maint (Daily)	SQL backup (Daily)	VM bck to vault (Daily)		

Dev and Test:

12am	1am	2am	3am	4am	5am	6am	7am	8am	9am	10am	11am	12pm	1pm	2pm	3pm	4pm	5pm	6pm	7pm	8pm	9pm	10pm	11pm
Both Dev & QA machines shut down								Manual start (when in use)													Both Dev & QA machines shut down		
								Manual (on-demand) ETL & SSAS processing										Auto patching (Fri)	SSIS maint (Daily)	SQL backup (Daily)	VM bck to vault (Fri)		

Auto patching: requires VM to be on.
VM backup to vault: not not require VM to be on.

Automatic Shutdown



Dev & Test

We are using the auto-shutdown feature within the VM properties (Dev/Test Labs).

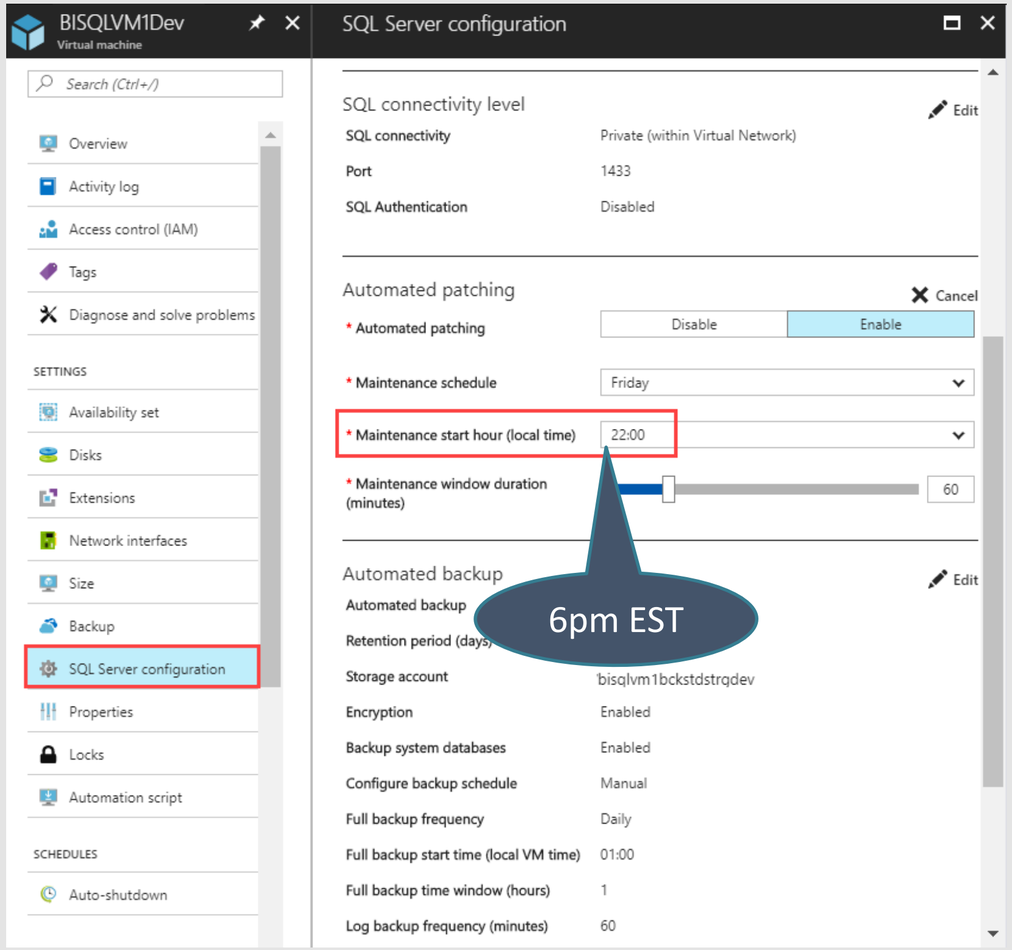
Production

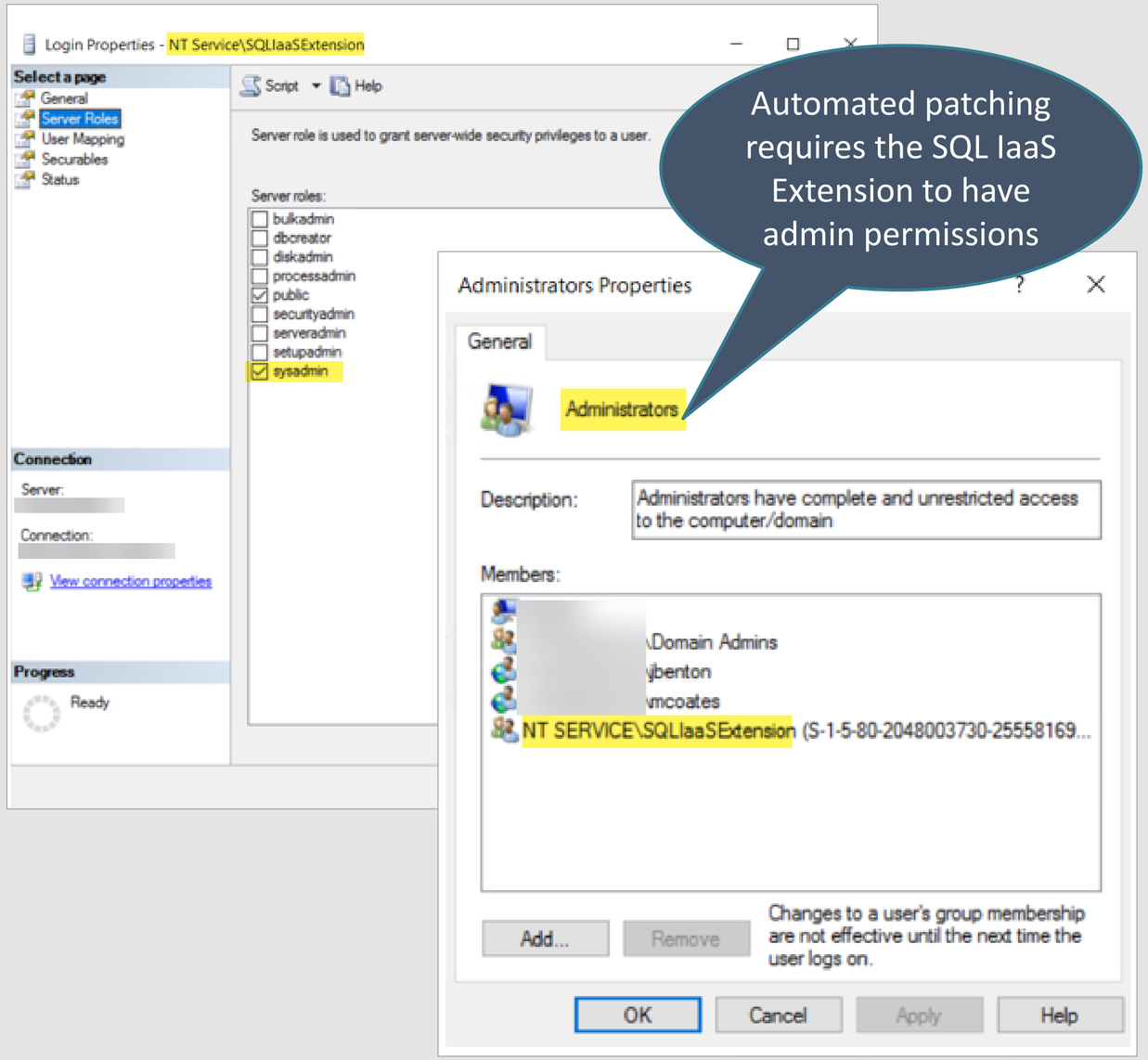
We are using Azure Automation to shut down, and restart, the 1st VM (DB engine/SSIS) overnight since our ETL runs 1x/hour during business hours only. This saves ~33% of compute cost.

The 2nd VM (SSAS) stays on 24/7.

Automatic Patching

We are using the Windows and SQL auto-patching feature. The VM “Manage Updates” feature is in private preview (July 2017).



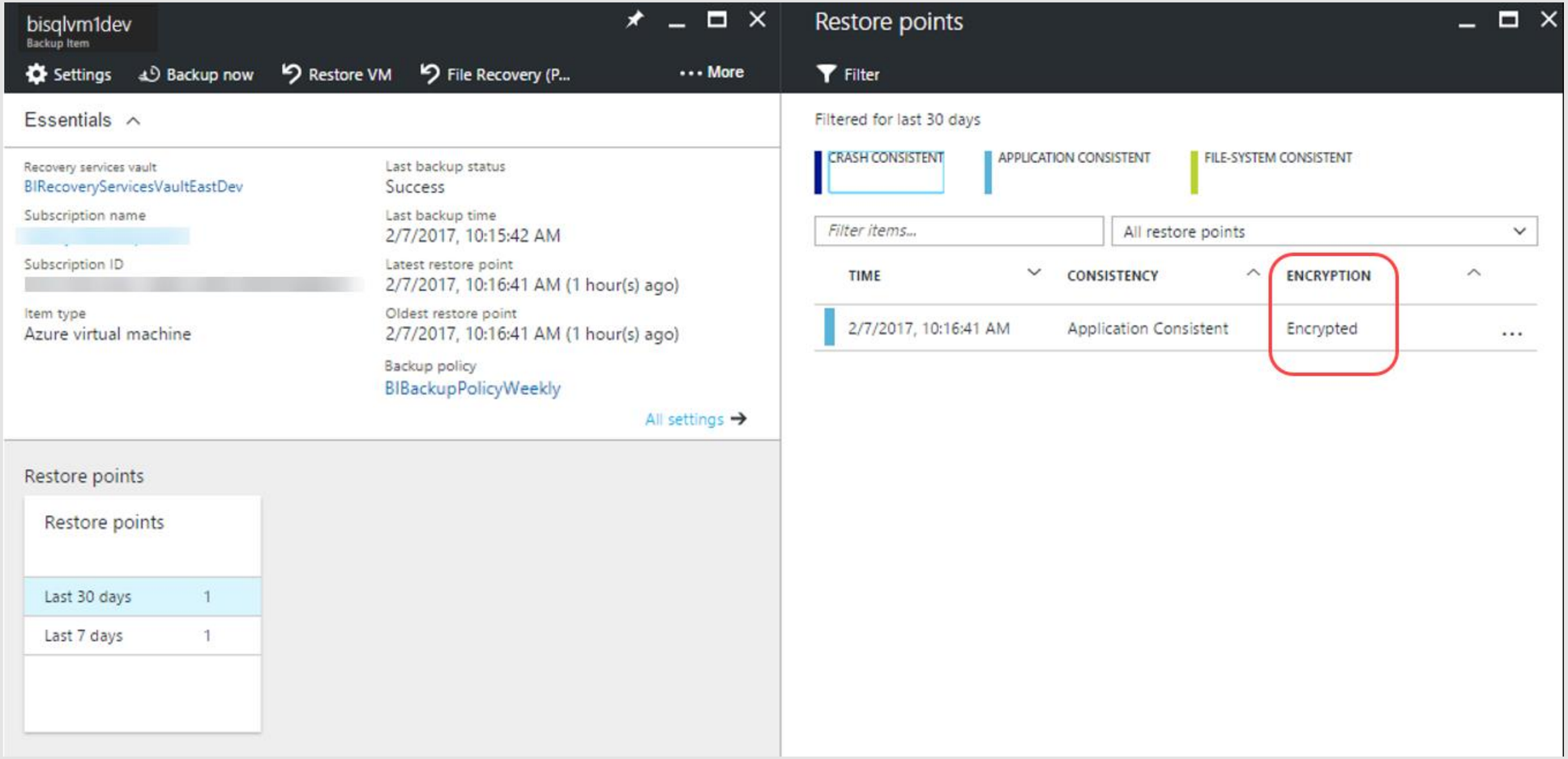


More info: https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sql/virtual-machines-windows-sql-server-agent-extension

VM Backups

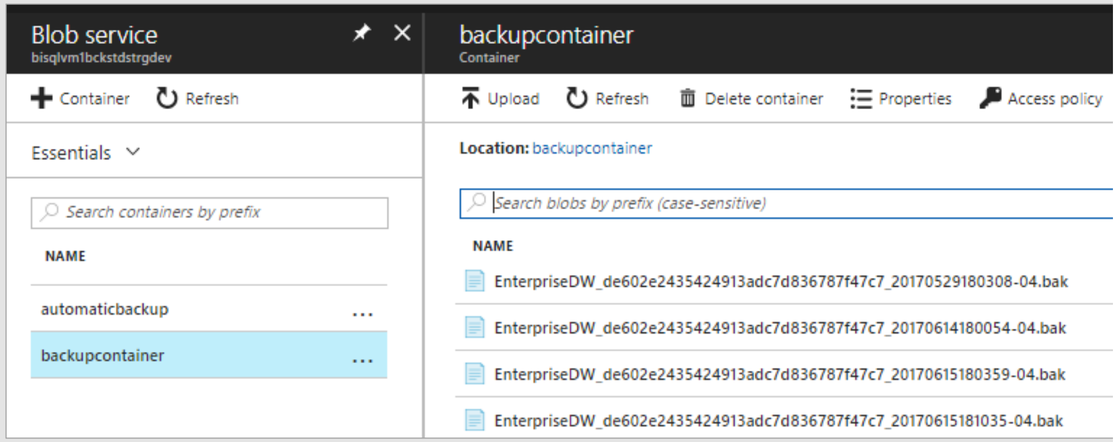
The VM backups are managed in the Recovery Services Vault.

For the VM backups to be encrypted, disk encryption is required (discussed in the previous section).

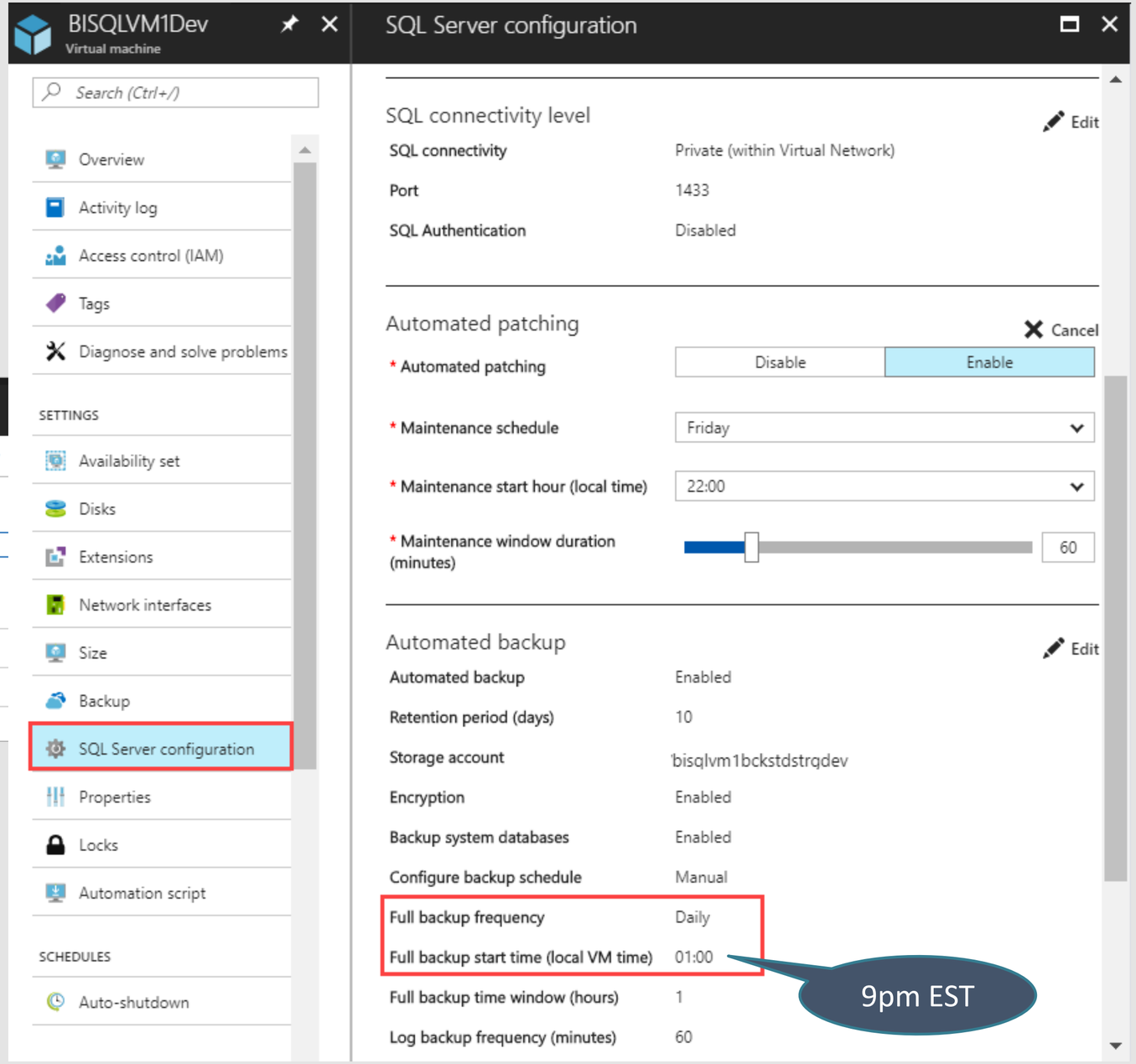


SQL Backups: From VM

We are using the scheduled SQL Server backups feature.



Be sure that this storage account is set to be encrypted – the “automaticbackup” container contains certificates & keys.



SQL Backups: From SQLDB

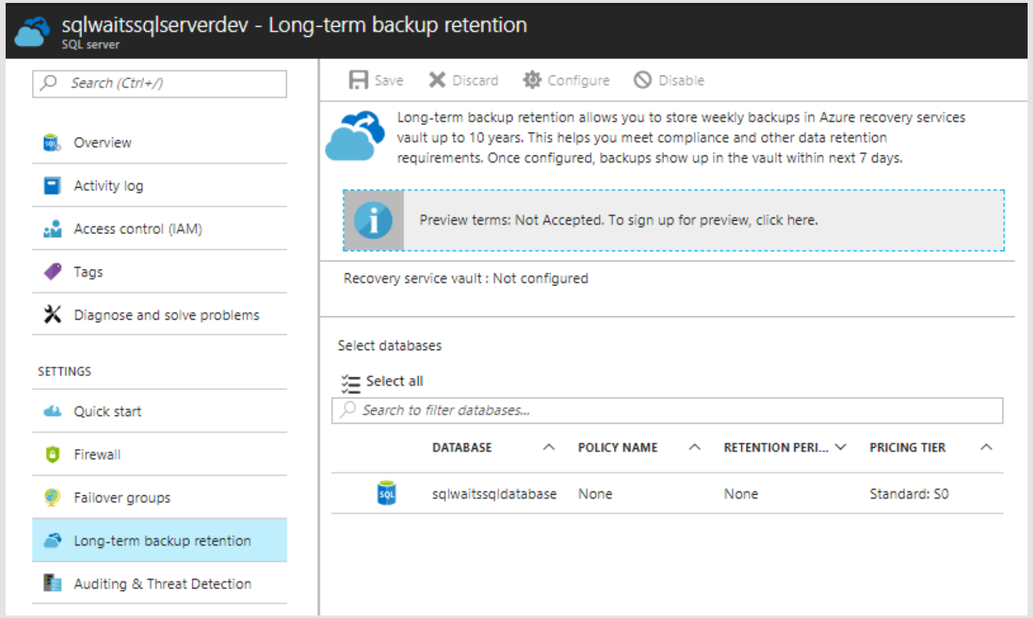
Backups for SQLDB depend on the pricing tier selected:

Full backups: weekly
Differential: every few hours
T-log: every 5-10 minutes

Basic service: retained 7 days
Standard service: retained 35 days
Premium service: retained 35 days

Longer-term backup retention:

- Set up a process to back up the .bacpac file (schema + data), or
- Recovery Services vault (still in Preview):



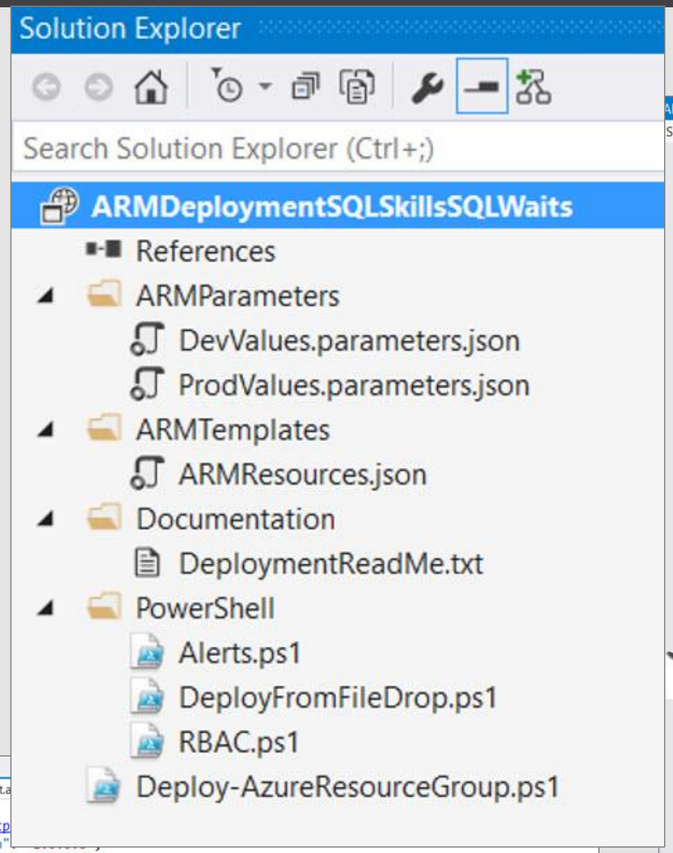
More info: https://blogs.msdn.microsoft.com/mast/2013/03/03/different-ways-to-backup-your-windows-azure-sql-database/

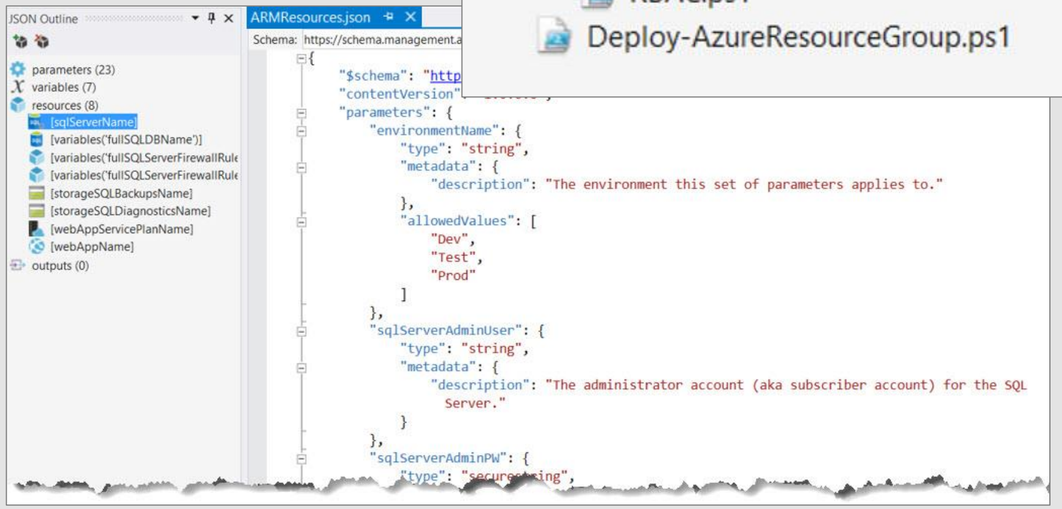
ARM Templates

Goal is for deployments to be modular & repeatable.

We have invested time in learning how to do ARM templates so we can make a good decision when to use them.

We have our "infrastructure as code" checked into source control within a Visual Studio project.





Blog post on ARM deployments: https://blogs.sentryone.com/melissacoates/getting-started-with-automated-arm-deployments-in-azure/

PowerShell

We are being selective about automation.
Balance of: wanting infrastructure in source code vs. little need for deployment repeatability.

We have some PowerShell scripts for deployment purposes, such as:

- Assign tags
- Enable disk encryption
- Provision key vault
- Deploy ARM template

```
#Input Area
$subscriptionName = 'InsertSubscriptionName'
$resourceGroupName = 'SQLSkillsWaitsLibraryRGDev'
$deploymentName = 'SQLSkillsSQLWaitsDeployment'
$templateFilePath = 'O:\Deployments\SQLWaits\2017-06-07\ARMResources.json'
$parameterFilePath = 'O:\Deployments\SQLWaits\2017-06-07\DevValues.parameters.json'

#ARM template deployment
New-AzureRmResourceGroupDeployment `
  -Name $deploymentName `
  -ResourceGroupName $resourceGroupName `
  -TemplateFile $templateFilePath `
  -TemplateParameterFile $parameterFilePath
```

Blog post on ARM deployments:
https://blogs.sentryone.com/melissacoates/getting-started-with-automated-arm-deployments-in-azure/

Recommendations for ARM and PowerShell

Use ARM for:

- Deployment of resources to Test and Prod
  - Recognition of dependencies
  - Parallel deployment of resources (faster than PowerShell or CLI)

Use PowerShell for:

- Execution of ARM template
- Management of resources (ex: start/stop a VM - in conjunction with Azure Automation)
- Selective items you would prefer in PowerShell vs. ARM (ex: Tags or Metrics Alerts)

Monitoring the Solution

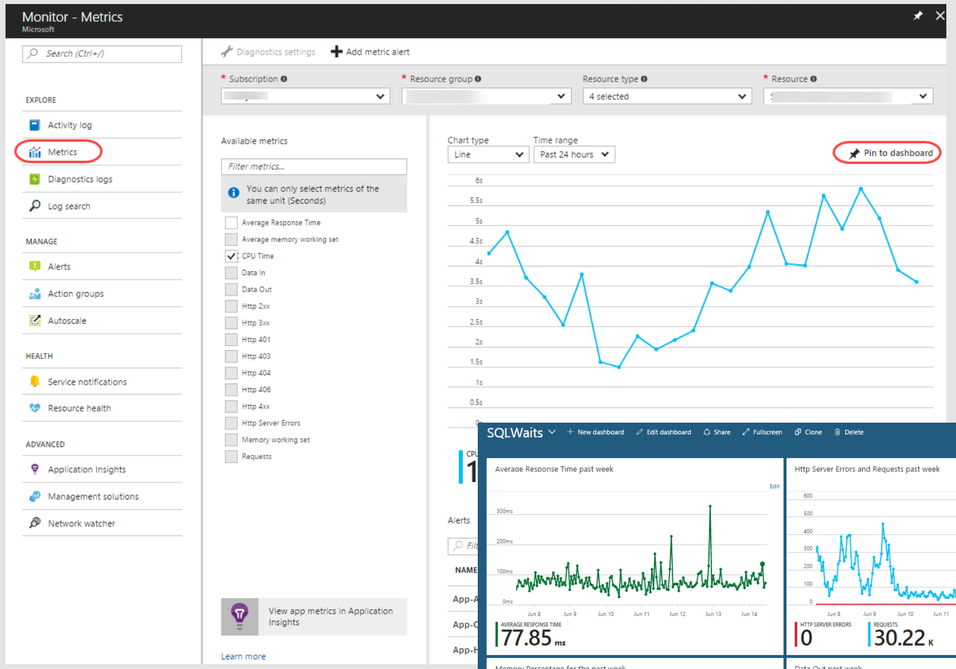
Monitoring

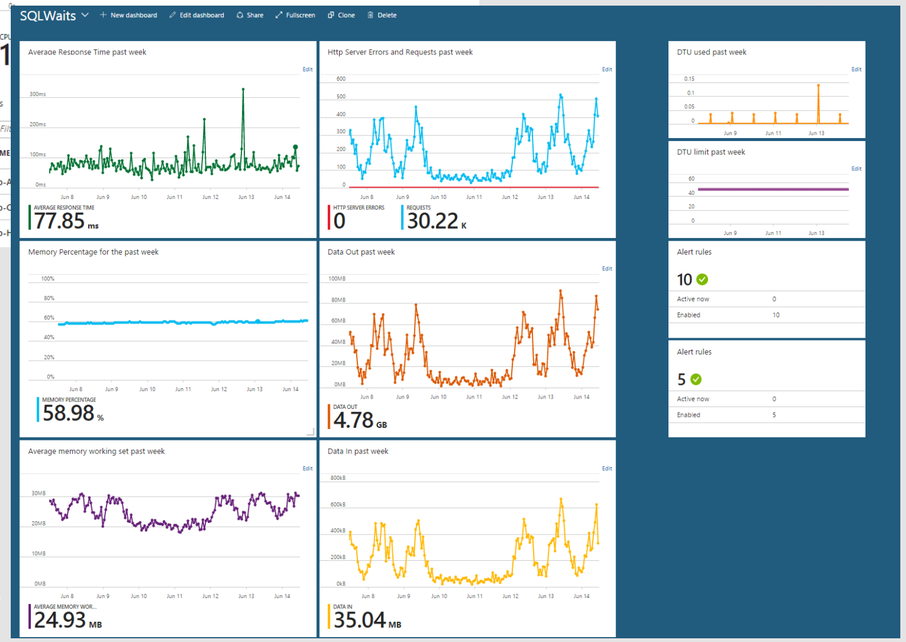
We are currently using a combination of:

- Azure Monitor (Metrics, Activity Log, Diagnostic Log)
- Azure Monitor Alerts
- Azure Blob Storage (Metrics & Activity Log Data)
- Azure Security Center
- SentryOne software

We have not needed Microsoft OMS (Operations Management Suite) as of yet, but we may grow into it.

Azure Monitor



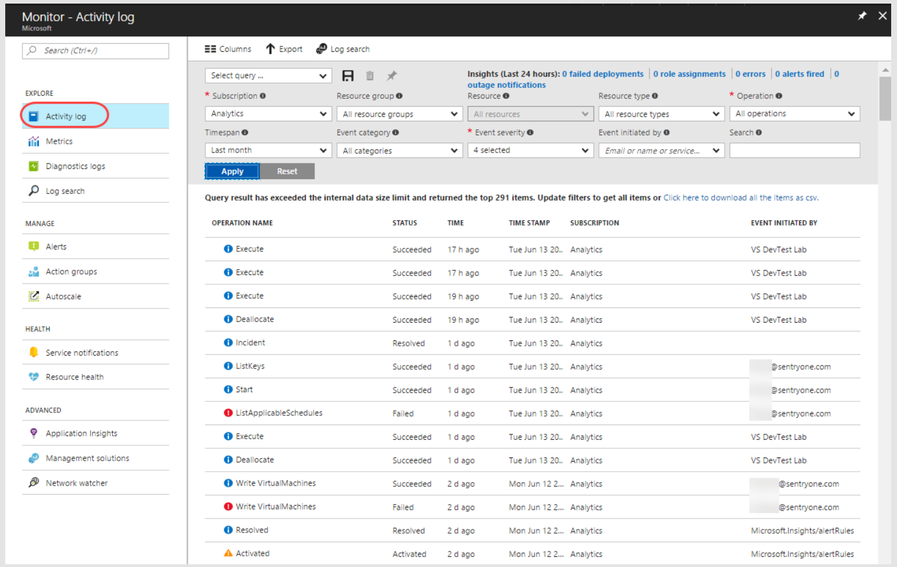


1 Metrics

- ✓ Performance counters in 1-minute frequency

2 Activity Log

- ✓ Captures create, update & delete for resources
- ✓ Still called ‘Operational Logs’ when sent to blob storage



3 Diagnostic Log

- ✓Emitted by each resource

Azure Monitor: Alerts

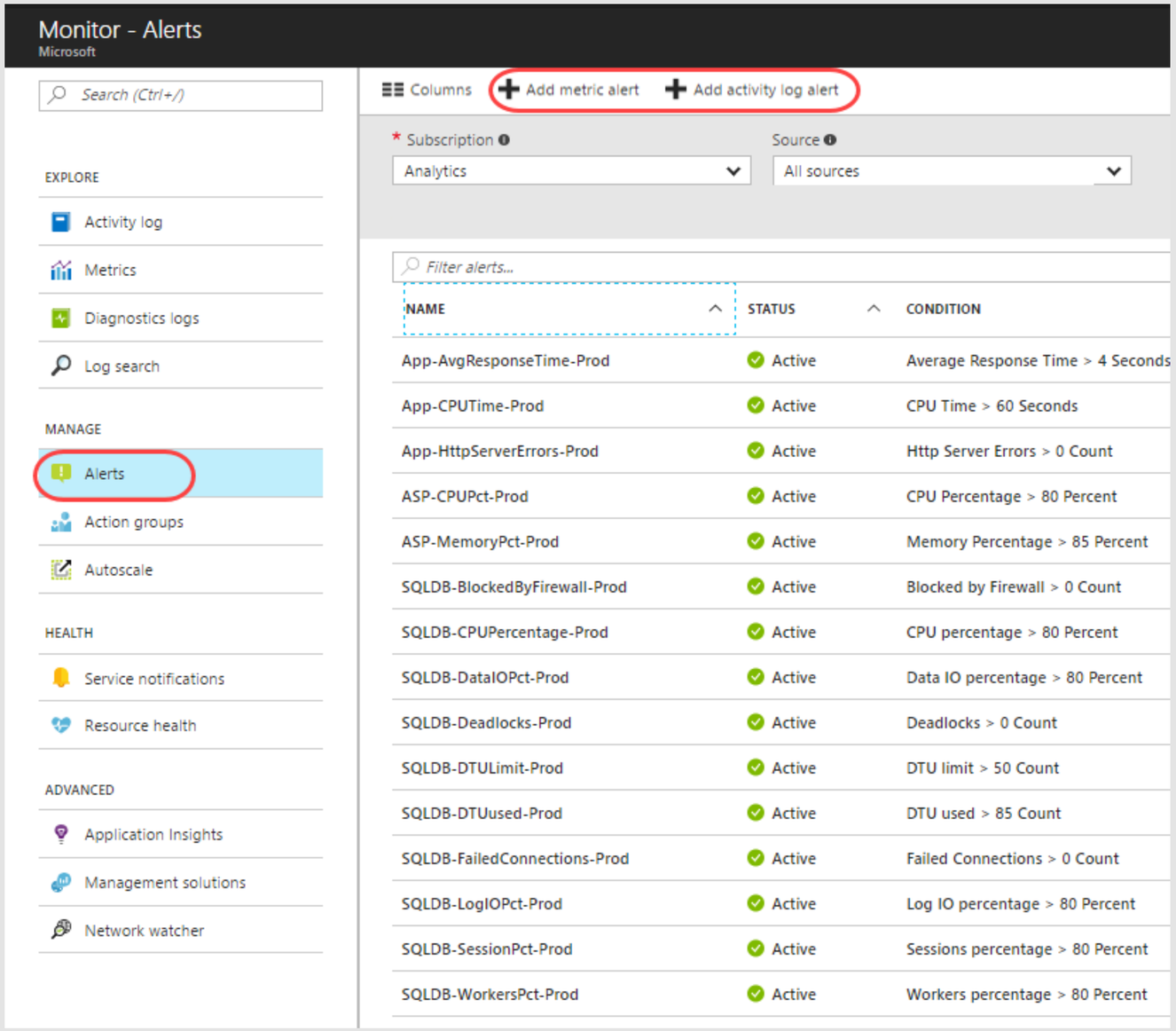
Metric Alert

- Email or webhook specified within the alert
- Always applies to one single resource

Activity Log Alert

- Can use an Action Group for SMS, email, or webhook
- Is a resource itself assigned to a RG
- Can span resource groups & resources

There are currently no alerts for the diagnostics logs.

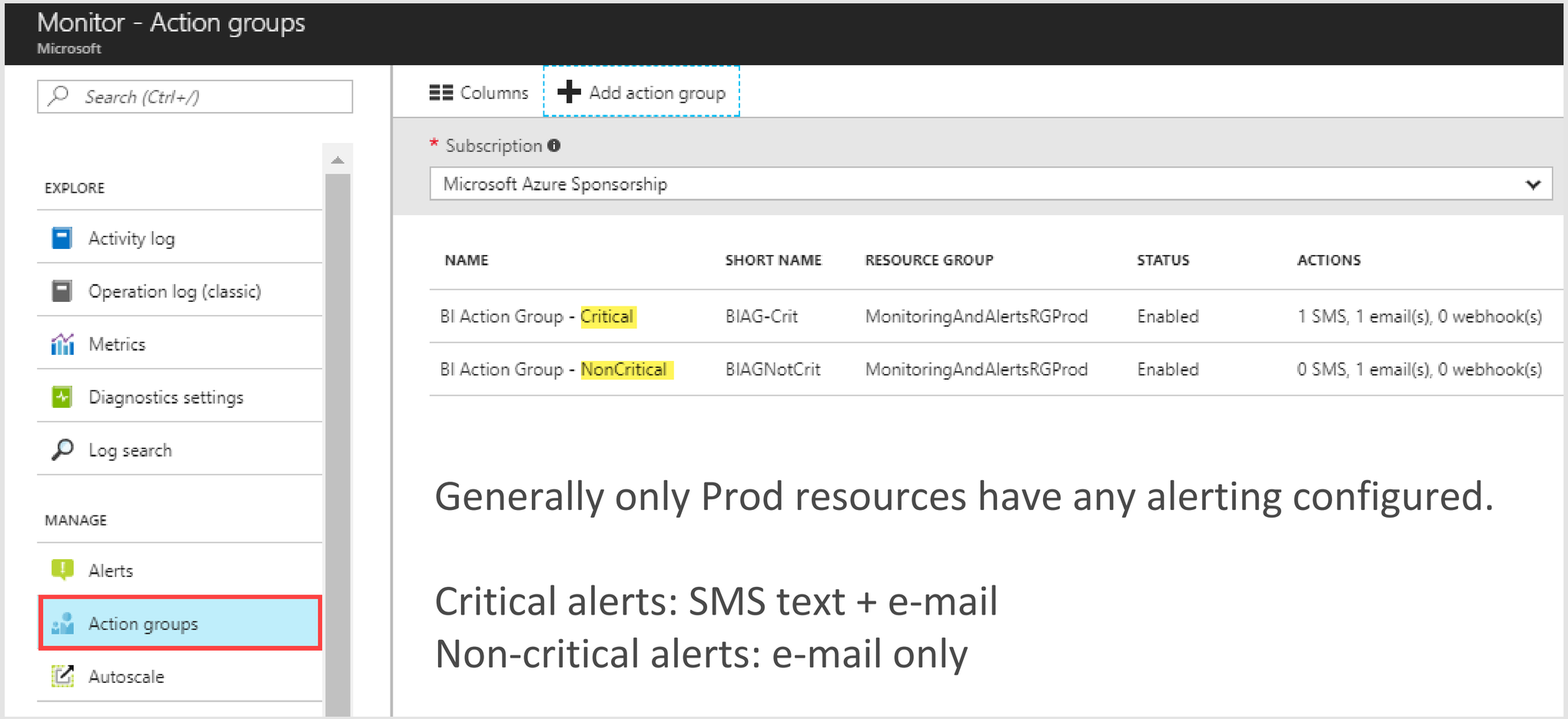


Azure Monitor: Action Groups

Used with Activity Log Alerts only right now (not supported with Metric Alerts).

Action types:

- SMS
- Email
- Webhook



Generally only Prod resources have any alerting configured.

Critical alerts: SMS text + e-mail
Non-critical alerts: e-mail only

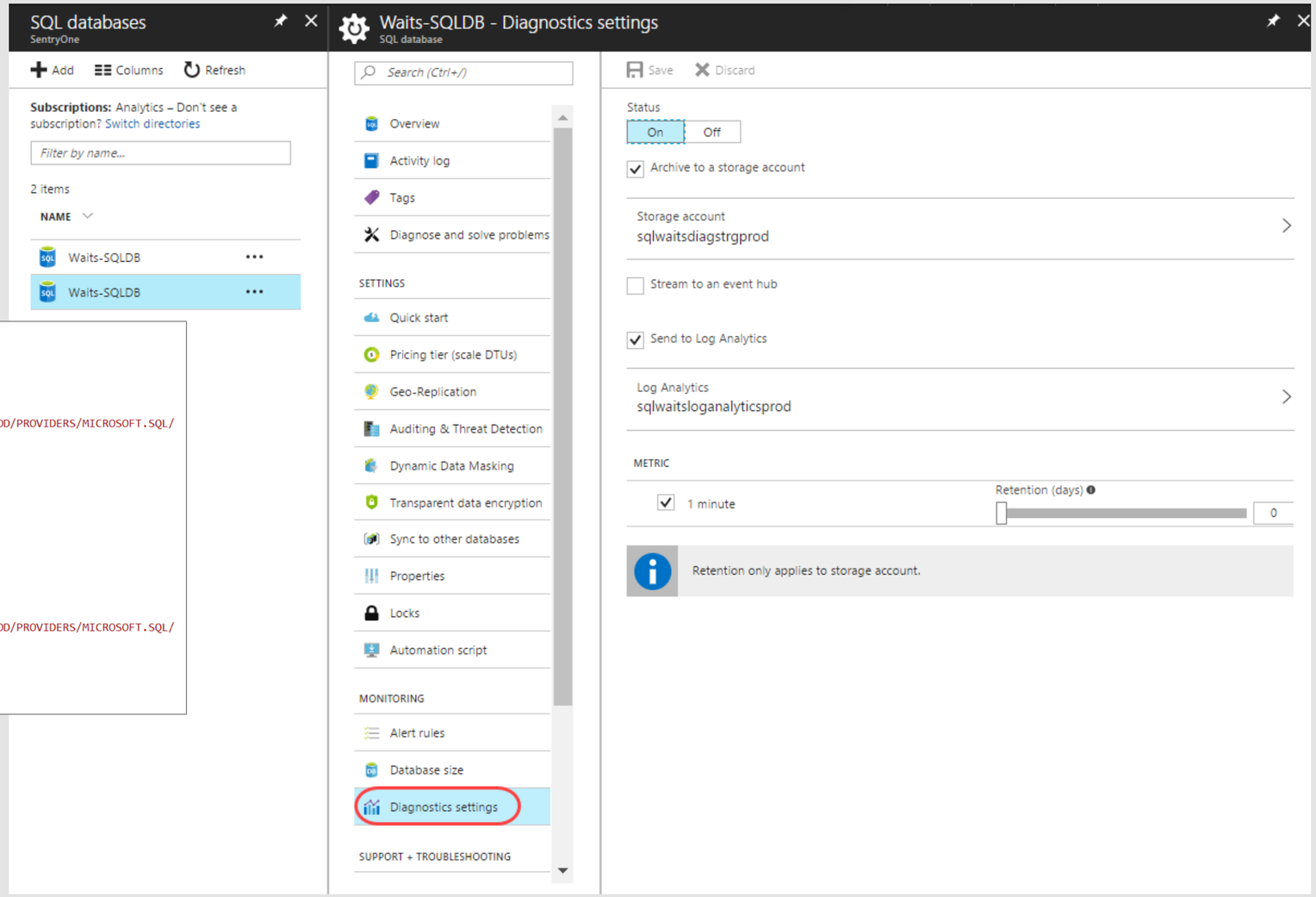
Azure Diagnostics: SQL Database (PaaS)

We are sending diagnostics to:

- Storage Account (JSON data)
- Log Analytics


```
{
    "count": 1,
    "total": 24247296,
    "minimum": 24247296,
    "maximum": 24247296,
    "average": 24247296,
    "resourceId": "/SUBSCRIPTIONS/[redacted]/RESOURCEGROUPS/SQLWAITSRGPROD/PROVIDERS/MICROSOFT.SQL/
      SERVERS/SQLWAITSSERVERPROD/DATABASES/WAITS-SQLDB",
    "time": "2017-05-10T14:17:00.0000000Z",
    "metricName": "storage",
    "timeGrain": "PT1M"
}
,
{
    "count": 1,
    "total": 1,
    "minimum": 1,
    "maximum": 1,
    "average": 1,
    "resourceId": "/SUBSCRIPTIONS/[redacted]/RESOURCEGROUPS/SQLWAITSRGPROD/PROVIDERS/MICROSOFT.SQL/
      SERVERS/SQLWAITSSERVERPROD/DATABASES/WAITS-SQLDB",
    "time": "2017-05-10T14:17:00.0000000Z",
    "metricName": "connection_successful",
    "timeGrain": "PT1M"
}
```

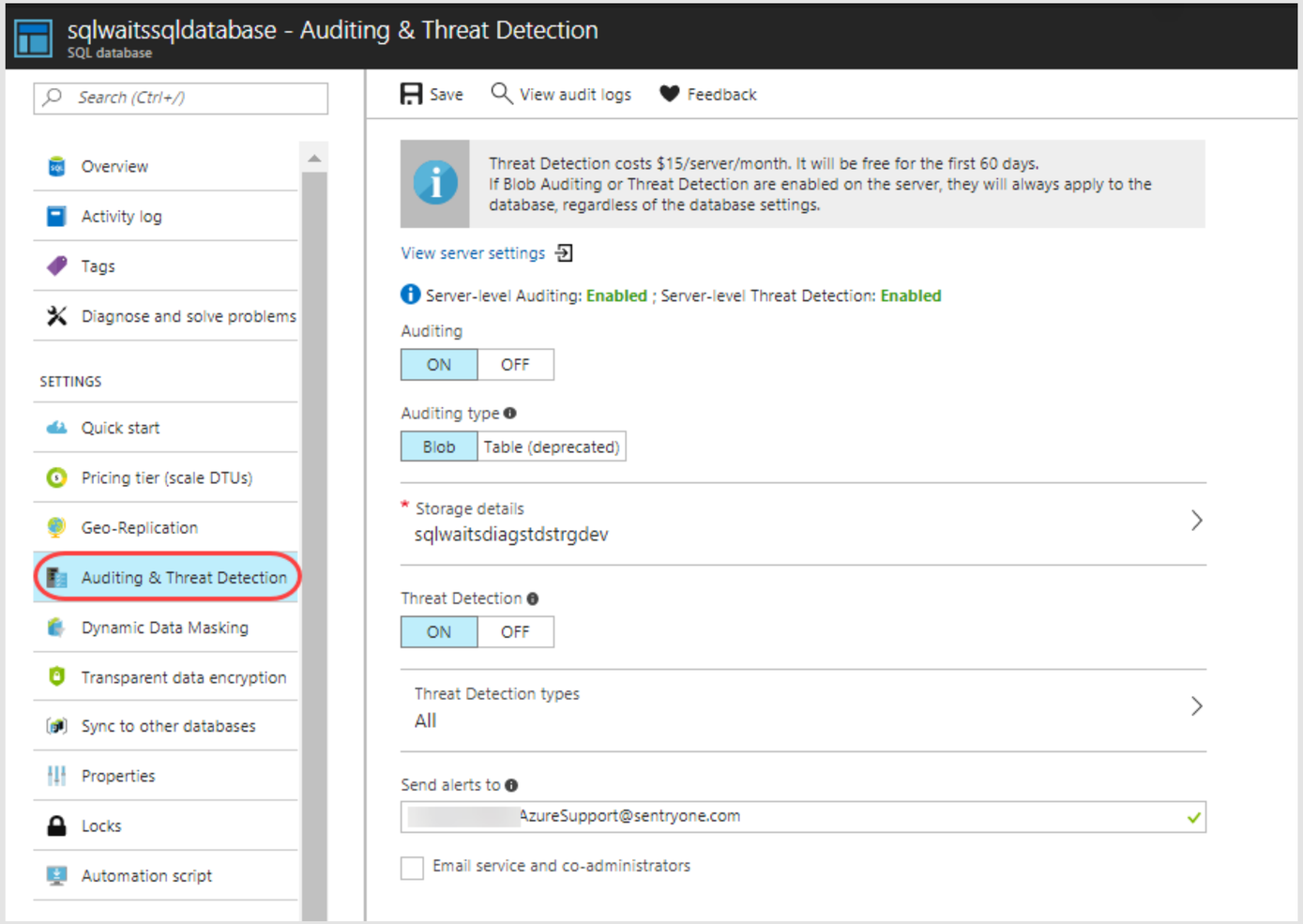

Partitioned in Storage as low as:
Resource Type > Year > Month > Day > Hour > Minute



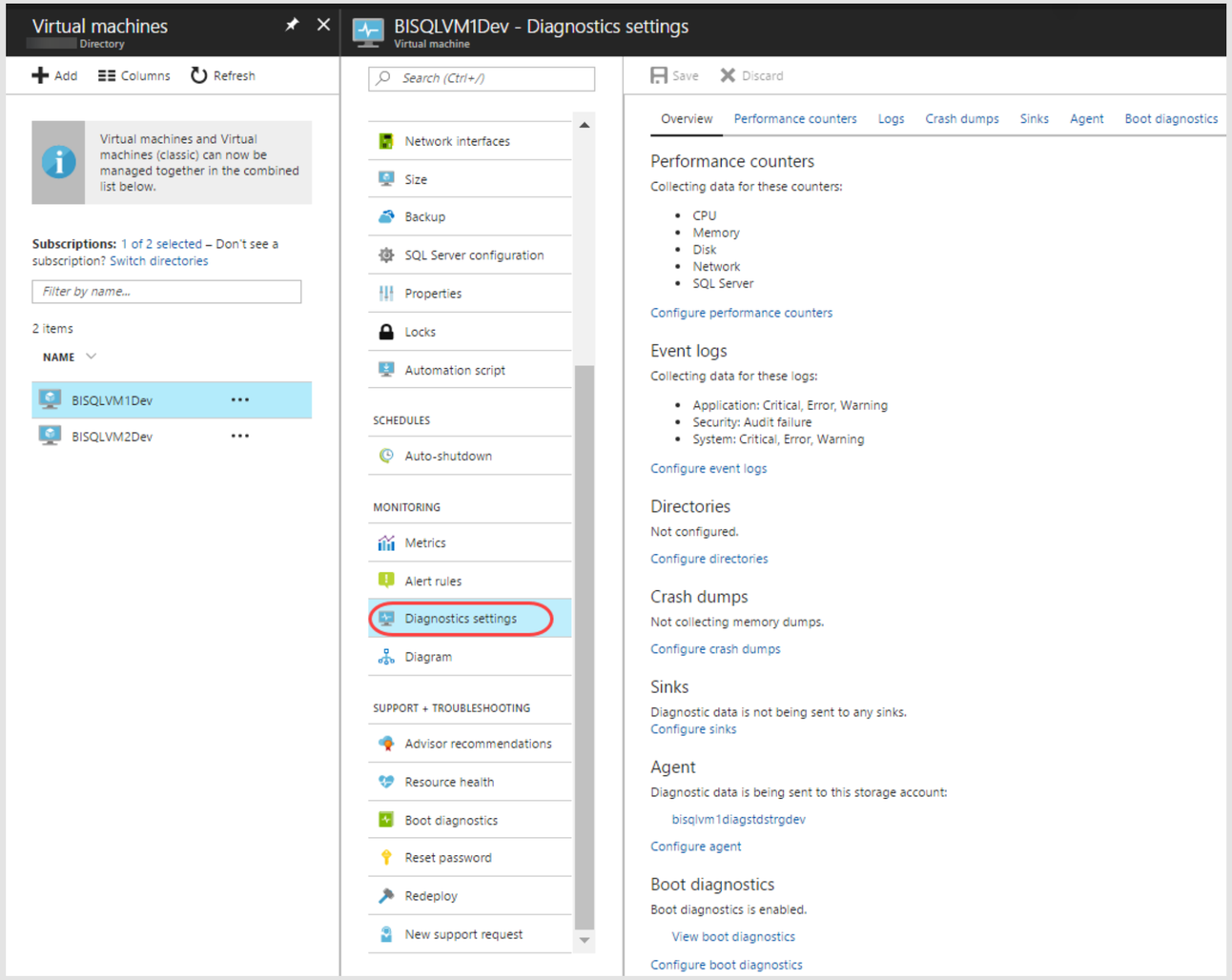
Azure Auditing: SQL Database (PaaS)

Auditing is set at both the Server and the Database level for SQLDB.

Send to: the ‘diagnostics’ blob storage account associated with this particular project.



Azure Diagnostics: Virtual Machine (IaaS)



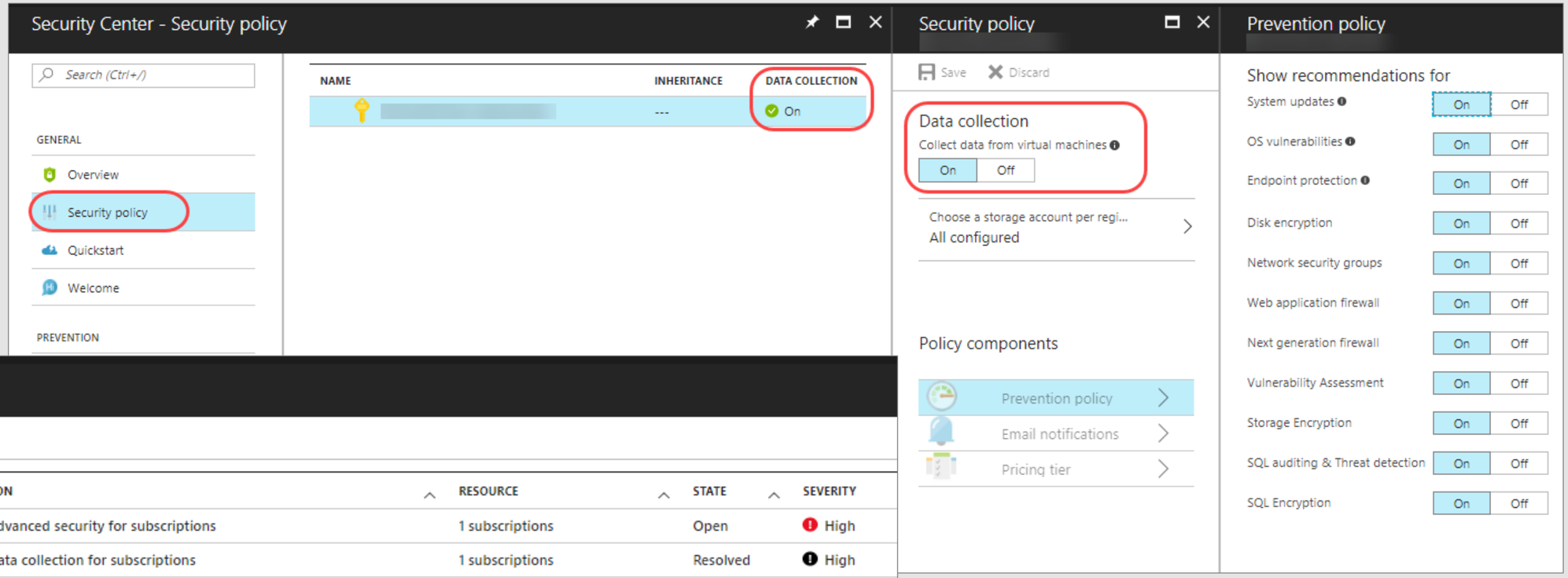
Performance Counters

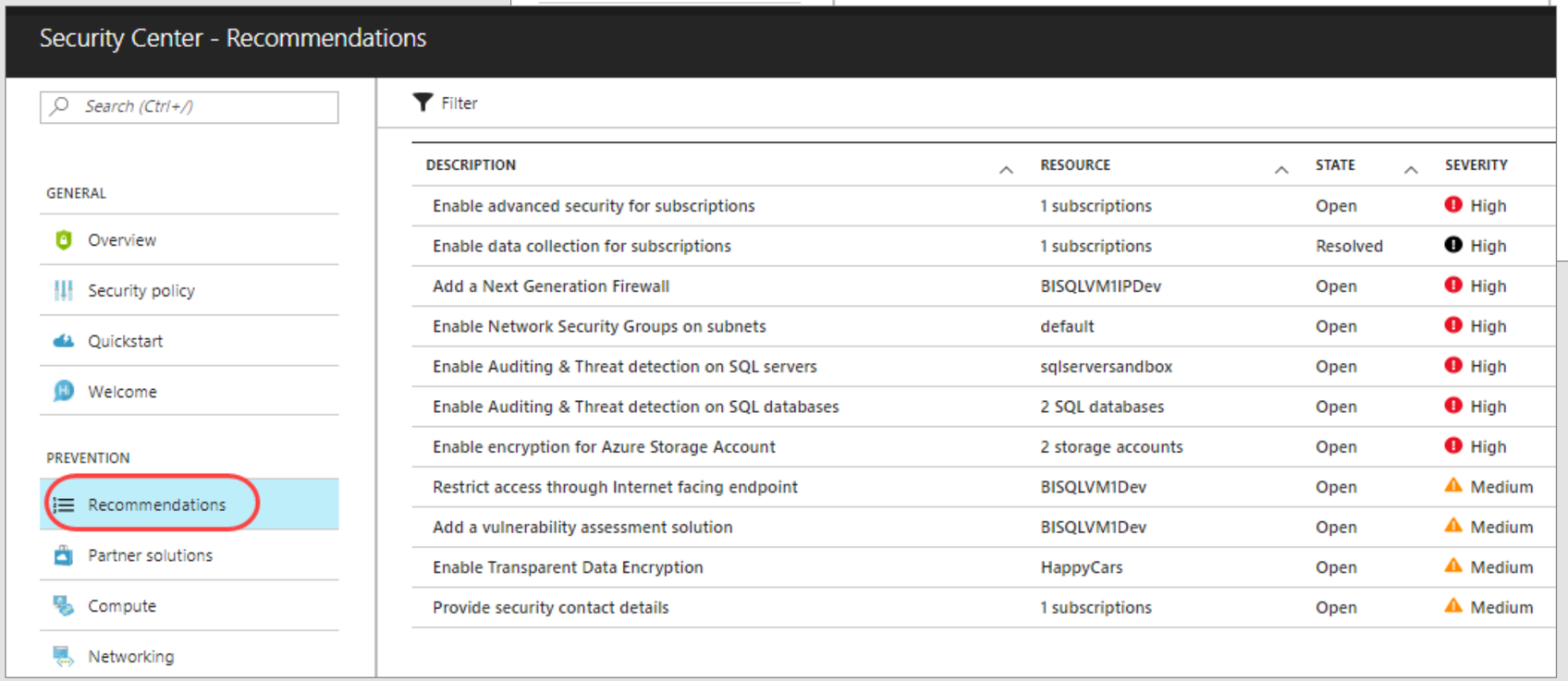
SQL Server counters are not gathered by default – need to explicitly select them.

Agent

The Azure Diagnostics Agent doesn't prompt for storage account – will need to change (i.e., if you have a naming convention for diagnostics storage), and then delete the auto-created storage account.

Azure Security Center





More info:
https://docs.microsoft.com/en-us/azure/security-center/security-center-recommendations

SentryOne Monitoring

Support for monitoring:

- ✓ SQL Server
  (on-prem, or in Azure or a cloud VM)
- ✓ Azure SQL Database
- ✓ Azure SQL Data Warehouse
- ✓ SQL Server Analysis Services
  (on-prem, or in Azure or a cloud VM)
- ✓ Windows Computer
- ✓ Tintri VMstore
- ✓ VMware Host

Key Takeaways
and
Q&A

Key Takeaways

The pace of change in Azure is frequent—plan to keep up with announcements. New features can change a previous decision really fast.

Look at using PaaS unless you really need an IaaS solution. Deploy the simplest solution that gets the job done.

Don’t shortcut the initial planning phase – do a POC if you can to learn. Don’t assume a feature is definitely available or works as you expect it to.

For unfamiliar features or services, consider creating a sandbox area to test & learn on first.

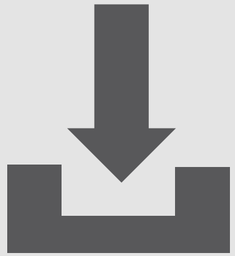
Plan to invest some time learning ARM templates & PowerShell, if you haven’t already.

Pay attention to efficiency and opportunities for cost savings. Developers & Administrators have a huge impact on cost.

Thank You!

To download a copy of this presentation:

SQLChick.com “Presentations & Downloads” page



Melissa Coates

Lead Analytics Architect,
SentryOne

Blog: sqlchick.com

Twitter: @sqlchick

Creative Commons License:
Attribution-NonCommercial-NoDerivative Works 3.0