Microsoft MSFarsi Community

GROUP # 1 SCHOLARSHIP AZ–104

NOV 30th ,2024

AMIR FARAHANCHI

Mahdi Muhaddesi

Sasan Arham

Ghazal Farajirad

LEARNING ORIENTED

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT OVERVIEW

Scenario Overview:
Global Media Corp needs a *private .NET application* hosted in Canada for internal use and a *virtual machine* in Germany to upload ~1,000 photos daily to Canada. The solution must be *cost-effective*, *secure*, and capable of *handling high-demand* requests while *automating* the photo upload process. The design should prioritize *cost management* and *security* while excluding Azure *WAF* and *Azure Firewall* in the initial phase.

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT KEY ELEMENTS

1. Data, Compute and Cost:

1. Deploy the .NET application in Canada region, minimizing costs for hosting and scaling
2. A storage account in Canada and transfer of ~1,000 daily photos from Germany
3. VM in Germany region

2. Networking and Security:

1. Limit access to the .NET application to ensure privacy and regional access
2. Secure data transfer between the Germany VM and Canada storage without WAF or Azure Firewall

3. Automation:

1. Implement a daily automated upload process for photos from the Germany VM to the Canada storage account.

4. Monitoring, Logs and alerts:

1. Track application performance and log analytics for resources

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT Dissection into 4 Major Areas

1. Data, Compute and Cost

Deployment:
Host the .NET application in Azure Canada using Azure App Service or a VM with necessary configurations for high-demand requests.

Photo Storage:
Use Azure Blob Storage in Canada for scalable and cost-efficient storage of photos.

Cost Optimization:

- Leverage Azure Reserved Instances or Spot VMs for hosting.
- Use Azure Storage tiers (e.g., Cool or Cold) for the non-critical photos.

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT Dissection into 4 Major Areas

2. Networking and Security

Private Access:
Use **Private Endpoint** or **Service Endpoint** to restrict the .NET application's accessibility within the Germany region.

Using NSG on both v-net

Data Transfer Security:
Enable **Azure Storage Secure Transfer** (HTTPS only).
Authenticate using SAS Token or Azure Managed Identity (utilizing Entra) to secure photo uploads.

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT Dissection into 4 Major Areas

3. Automation

Photo Upload Automation:
Set up a PowerShell script to run on the Germany VM for daily uploads at a set time.
Use Azure Automation solutions to run the script automatically.

Process efficiency:
Compare available options for the implementation of the automation process from multiple aspects.

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT Dissection into 4 Major Areas

4. Monitoring and Logs

Application and resources Monitoring:
Use **Azure Monitor** and **Application Insights** to track the .NET application’s performance.

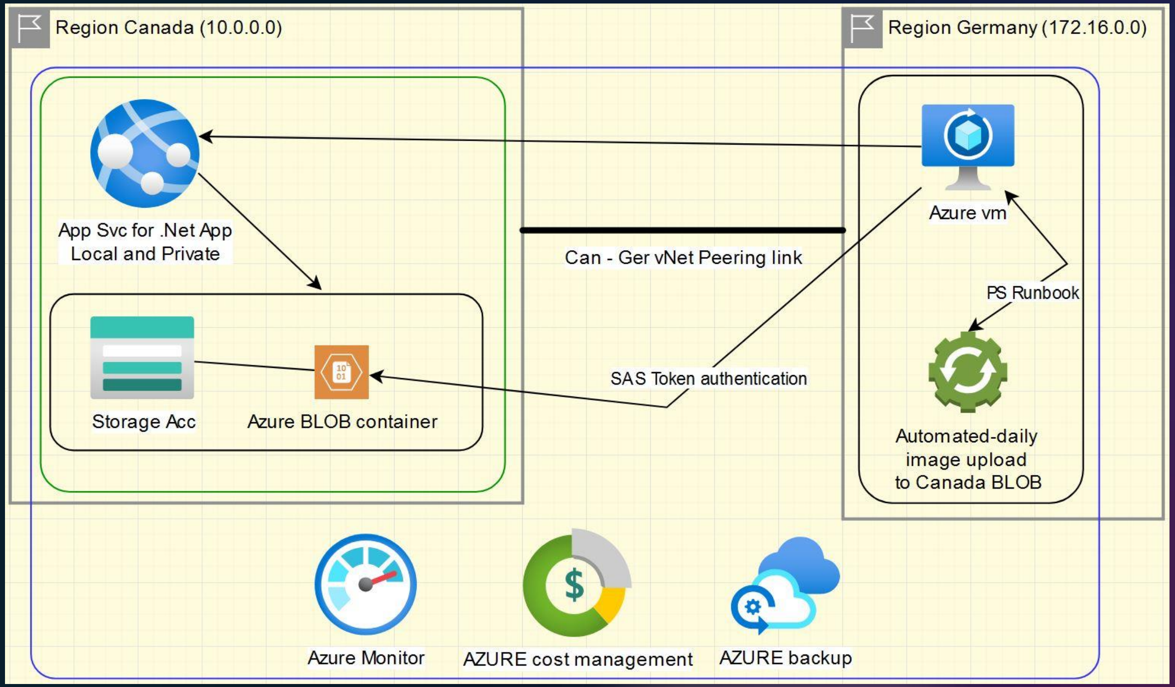
Configure **Log Analytics Workspace** to capture logs from the other resources and their activities.

Requirement: Enable diagnostics settings for all resources to be able to monitor activities.

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT SOLOUTION ARCHITECTURE AT A GLANCE



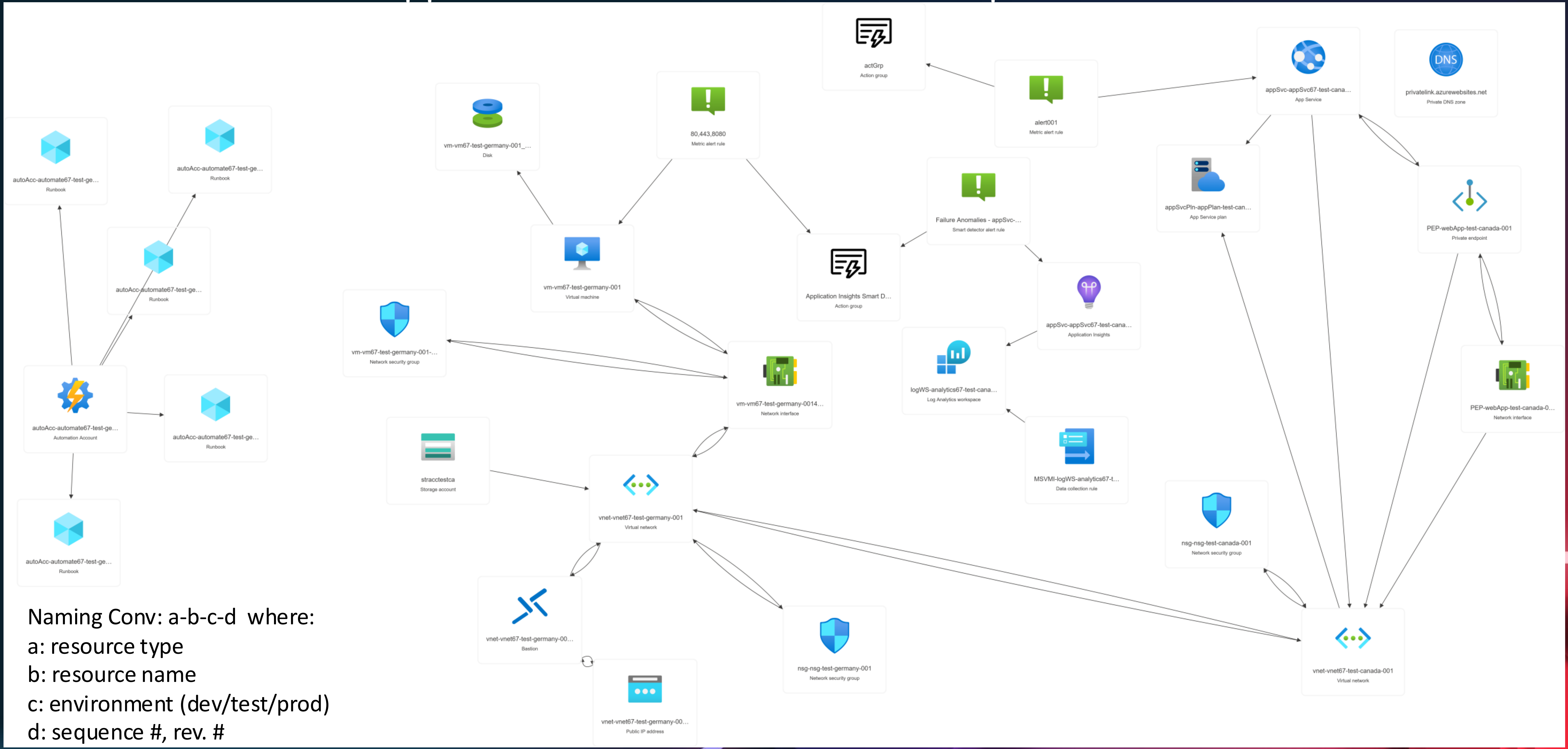
GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario


actGrp
Action group
alert001
Metric alert rule
appSvc-appSvc67-test-cana...
App Service
privatelink.azurewebsites.net
Private DNS zone
80,443,8080
Metric alert rule
vm-vm67-test-germany-001_...
Disk
autoAcc-automate67-test-ge...
Runbook
autoAcc-automate67-test-ge...
Runbook
autoAcc-automate67-test-ge...
Runbook
Failure Anomalies - appSvc-...
Smart detector alert rule
appSvcPln-appPlan-test-can...
App Service plan
PEP-webApp-test-canada-001
Private endpoint
vm-vm67-test-germany-001
Virtual machine
Application Insights Smart D...
Action group
appSvc-appSvc67-test-cana...
Application Insights
vm-vm67-test-germany-001-...
Network security group
vm-vm67-test-germany-0014...
Network interface
logWS-analytics67-test-cana...
Log Analytics workspace
PEP-webApp-test-canada-0...
Network interface
autoAcc-automate67-test-ge...
Automation Account
autoAcc-automate67-test-ge...
Runbook
stracctestca
Storage account
vnet-vnet67-test-germany-001
Virtual network
MSVMI-logWS-analytics67-t...
Data collection rule
nsg-nsg-test-canada-001
Network security group
autoAcc-automate67-test-ge...
Runbook
vnet-vnet67-test-germany-00...
Bastion
nsg-nsg-test-germany-001
Network security group
vnet-vnet67-test-canada-001
Virtual network
vnet-vnet67-test-germany-00...
Public IP address

Naming Conv: a-b-c-d where:
a: resource type
b: resource name
c: environment (dev/test/prod)
d: sequence #, rev. #

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost** (VM and App Service type selection)

We've designed this solution to keep it cost-efficient while maintaining performance during periods of higher workload. The solution will be deployed using low-cost and small compute resources, and after monitoring your real-world workload, it will adjust to meet requirements if needed. Below are the cost predictions for three different possible scenarios:

- Virtual Machine:

VM Name	B1ms(C:1,R:2 GiB)	D2as v6(C:2,R:8 GiB)	F16s v2(C:16,R:32 GiB, TS: 128 GiB)	E4d v5(C:4,R:32 GiB)
Monthly Cost	On demand:$17.52 Spot: N/A	On demand:$80.3 Spot: $20.07	On demand:$566.48 Spot: $79.3072	On demand: $252.58 Spot: $35.3612

Note: After monitoring the nature of your software and its resource consumption, we will determine which VM family is suitable for your use case: the D family for applications that require balanced resources, the F family, which is powered by high-frequency CPUs for applications that consume more CPU power, and the E family for applications that primarily consume memory.

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost** (Low Cost VM type selection)

Why we suggest Spot VM for your project?

Azure Spot Virtual Machines (VMs) have some great cost-saving advantages. One of the main benefits is that Spot VMs can be up to 90% cheaper than regular pricing. This makes them perfect for non-critical tasks that can handle interruptions, like your case in this project.

On the downside, there are some important disadvantages to consider. The biggest issue is that there are no service level agreements (SLAs) with Spot VMs, and they can be evicted with very little notice 30 seconds) if Azure needs the resources back.
This unpredictability can be a problem for workloads that need high availability or consistent performance which is not your case.

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost** (App Service type selection)

- App Service:

App Service Name	B1(C:1,R:1.75 GiB, S:10GB)	B3(C:4,R:7 GiB, S:10GB)	P3v3(C:8,R:32 GiB, S:250 GiB)	I3v2(C:8,R:32 GiB, S:1TB)
Monthly Cost	$60.59/month	$240.90/month	$957.76/month	$1,635.20/month

B: App Service Basic plan - designed to run workloads that have low traffic requirements, and do not require advanced auto-scale and traffic management features.
P: App Service Premium plan - A high performance, high reliability, and highly scalable service that offers the latest Azure platform capabilities and innovations as a multi-tenant managed service.
I: App Service Environment plan - An App Service Environment is an Azure App Service feature that provides a fully isolated and dedicated environment for running App Service apps securely at high scale.

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

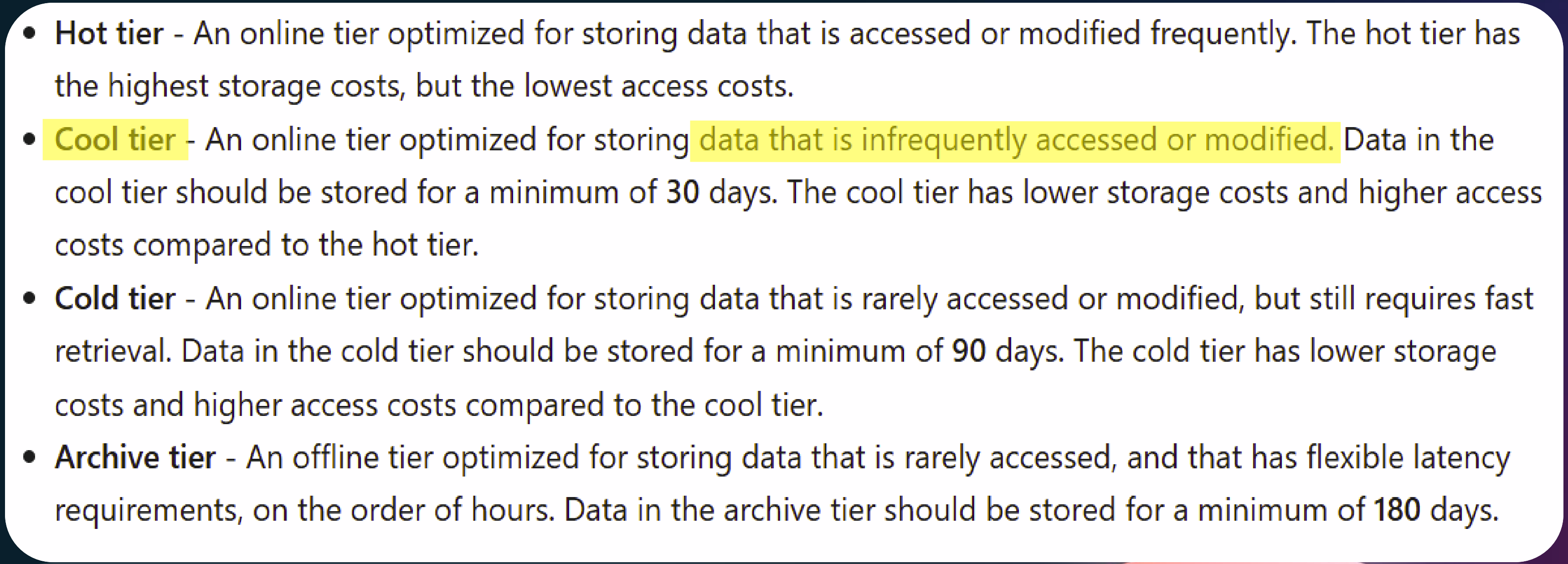
NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost** (Storage tier selection)

Storage account BLOB tier identification based on project requirements:



- **Hot tier** - An online tier optimized for storing data that is accessed or modified frequently. The hot tier has the highest storage costs, but the lowest access costs.
- **Cool tier** - An online tier optimized for storing data that is infrequently accessed or modified. Data in the cool tier should be stored for a minimum of **30** days. The cool tier has lower storage costs and higher access costs compared to the hot tier.
- **Cold tier** - An online tier optimized for storing data that is rarely accessed or modified, but still requires fast retrieval. Data in the cold tier should be stored for a minimum of **90** days. The cold tier has lower storage costs and higher access costs compared to the cool tier.
- **Archive tier** - An offline tier optimized for storing data that is rarely accessed, and that has flexible latency requirements, on the order of hours. Data in the archive tier should be stored for a minimum of **180** days.

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost** (Storage tier selection)

Storage account BLOB tier identification based on project requirements:

Scenario/Tier	Hot	Cool	Cold	Archive
If 1 MB per file				
Your First Month Storage Cost Bill (30 GB):	$0.6	$0.33	$0.135	$0.054
Your Third Month Storage Cost Bill (30 + 60 + 90 GB):	$3.6	$1.98	$0.81	$0.0.324
Data Retrieval Per 180GB	$0	$1.8	$5.4	$3.96 / High Priority: $21.6
If 5 MB per file				
Your First Month Storage Cost Bil(150 GB):	$3	$1.65	$0.675	$0.27
Your Third Month Storage Cost Bill (150 + 300 + 450 GB):	$18	$9.9	$4.05	$1.62
Data Retrieval Per 900GB	$0	$9	$27	$19.8 / High Priority: $108
If 10 MB per file				
Your First Month Storage Cost Bil(300 GB):	$6	$3.3	$1.35	$0.54
Your Third Month Storage Cost Bill (300 + 600 + 900 GB):	$36	$19.8	$8.1	$3.24
Data Retrieval Per 5000GB	$0	$50	$150	$110 / High Priority: $600

Scenario Description: For simplicity, this table expects 30,000 files to be uploaded on the first day of each month instead of 1,000 files daily.
Region: Canada Central, Redundancy: LRS

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost (App service deployment type)**

Why we suggest "CODE" App service deployment type vs. "CONTAINER" for your project?

Feature	Code	Container
Ease of Deployment	Simplified; no Docker knowledge	Requires Docker setup and expertise
Customization	Limited	Full control over runtime & libraries
Portability	Tied to Azure App Service	Portable across platforms
Scaling	Azure manages automatically	Azure scales containers, but you manage dependencies
Use Case	Standard web apps	Complex or custom requirements

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost (App service scalability)**

Why we suggest "Scalable Web App" vs. other scalability solutions (handling high load efficiently with reasonable cost)

	Feature	Description	Advantages	Disadvantages
	Scalable Web App (Autoscaling)	Automatically increases the number of Web App instances during high demand.	- Smart management of high traffic. - Cost-effective and fully automated.	- Costs may rise significantly with high instance scaling.
	Load balancer	Distributes requests among multiple servers or services in the Backend Pool.	- Manages load and balances traffic. - Increases availability of resources.	- Requires multiple resources in the Backend Pool. - Adds cost without significant improvement in this case.
optional	**CDN (Content Delivery)**	Stores and delivers static files (images, CSS, JS) through a CDN network to users.	- Reduces load on the Web App. - Improves static content load times for users.	- Only applicable for static files. - Extra cost for CDN traffic.
optional	**Optimize Code & Queries**	Improves application logic, uses connection pooling, and database indexing for better efficiency.	- Zero cost. - Enhances overall Web App and database performance.	- Requires time and expertise for analysis and optimization. - Limited impact for very high loads.

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost (App service auto-scaling options)**

Why we suggest "**Scale Out**" vs. **Scale Up** (handling high load efficiently with reasonable cost)

Feature/Aspect	**Scale Up (Vertical)**	**Scale Out (Horizontal)**
Approach	Increases resources per instance	Increases the number of instances
Used For	Resource-intensive workloads	☺ High traffic and concurrent requests
Cost Impact	Higher tier costs more	Each additional instance adds to the cost
Performance Impact	Improves individual instance performance	☺ Improves scalability by distributing the load (auto)
Maximum Limit	Limited to the highest tier available	Limited by the App Service Plan's instance cap
Downtime	Possible during tier change	☺ No downtime during scaling

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost (App service database – SQL based)**

Why we suggest "Azure SQL Database" for App service in your project? (Not included in DEMO)

DATABASE TYPES	Azure SQL Database	Azure SQL Managed Instance	SQL Server on Azure Virtual Machines
Deployment Model	Platform-as-a-Service (PaaS)	Platform-as-a-Service (PaaS) with full SQL Server engine	Infrastructure-as-a-Service (IaaS) with full SQL Server engine
Management	Fully managed by Azure, automated backups, patching, and updates	Fully managed, but provides more control over configuration and features	Managed by the user, needs manual patching and backups
Customization	Limited, optimized for basic use cases	High level of customization and feature support for legacy applications	Full control over OS and SQL Server version and configurations
Scalability	Automatic scaling, but within certain limits	Can scale within limits defined by instance size and configuration	Requires manual scaling, limited by VM size and capacity
Cost	Pay-as-you-go, based on DTUs or vCores	Pay-as-you-go, more expensive than Azure SQL Database, based on vCores	Pay-as-you-go, based on VM size, storage, and licensing
Use Case	Ideal for cloud-native applications with moderate resource requirements	Ideal for migrating legacy SQL Server applications to Azure, or hybrid environments	Best for applications requiring full control over the SQL Server setup, legacy applications, or specific configurations
High Availability and Disaster Recovery	Built-in with automated failover, zone-redundant options available	Built-in high availability with failover cluster support	You must configure your own high availability solutions (e.g., Always On Availability Groups)
Security	Built-in security features (e.g., firewall, encryption, threat detection)	Advanced security features, similar to on-prem SQL Server	Must configure your own security (e.g., firewalls, encryption)
Backup and Restore	Automated backups with point-in-time restore capabilities	Full backup and restore capabilities with more flexibility	You must manage backups manually, but full flexibility is available

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 1: **Data, Compute and Cost**

Considerable Business continuity, Backup and Disaster recovery options for this project (OPTIONAL - Partially included in DEMO)

Business continuity	Azure Site Recovery, Private Endpoint, Application Auto-Scaling, Load Balancer, Azure SQL Database Geo-Replication, Zone-Redundant Storage (ZRS), Geo-Redundant Storage (GRS).
Backup	Azure Backup, Snapshots (Azure Storage), Managed Disk Backups, Azure SQL Automatic Backups, Azure Backup Vault, Veeam Backup for Azure, Acronis Backup.
Disaster recovery	Azure Site Recovery (ASR), Geo-Redundant Storage (GRS), SQL Database Geo-Replication, Veeam Disaster Recovery for Azure, Azure Traffic Manager for Region Failover.

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 2: **Networking and Security**

NETWORKING & SECURITY OPTIONS USED IN THIS DEMO

1. Virtual Network (VNet) with different ranges (10.0.0.0/24 and 172.16.0.0/24) and limit total host IPs by subnets

2. Peering (***Inbound and outbound traffic is charged at both ends of the peered networks.)*** between vnets

3. Network Security Group (NSG) to restrict access to private v-nets of regions

4. Total Internal Access (NO PUBLIC IP)

5. Private Endpoint usage

6. Azure Bastion (for NON-RDP remote connection to VM – for test usage only, cost excluded)

7. SAS token usage for BLOB storage access (instead of managed identity by Entra ID)

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 3: **Process Automation**

Sub-project Overview

The intended sub-project is related to the implementation of an *automatic image upload system*, on a daily basis from a *virtual machine* located in Germany to a *BLOB* in Canada, which uploads about 1000 photos daily.

Different methods of automation

To implement the automation system, several different methods have been investigated. These methods include using tools such as *Azure Logic Apps*, *Azure Functions*, and *Azure Automation Account*.

Suggested method and reasons for choosing

After a thorough review of the methods, it was decided to use *Azure Automation* using Scheduled *Runbook* with *Hybrid Worker*. This method seemed to be the most suitable option considering the *cost*, *security* , simplicity and *project requirements*. Since the Germany *VM* is configured as a *Hybrid Worker*, it can easily run the required *PowerShell scripts* and upload images to the desired destination. (more detail follows)

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 3: **Process Automation**

Sub-project automation process selection >> Azure Automation vs. Azure Logic Apps vs. Azure Functions

Criteria	Azure Automation	Azure Functions	Azure Logic Apps
Cost	Cost-effective for long-running tasks.	Higher cost for high-frequency executions.	Higher cost due to multiple workflow actions.
Ease of Setup	Easy to automate with minimal configuration.	Requires developer expertise for optimization.	Complex setup for long-running tasks.
Scalability	Highly scalable for daily upload tasks.	Performance may degrade with high volume.	Not optimized for large, repetitive tasks.
Integration with Azure	Seamless integration with Azure services (Blob Storage, VM).	Requires manual setup for Azure services integration.	Excellent for integrating workflows, but not suited for bulk uploads.
Security	Secure with Managed Identity and Entra or SAS Token for storage connection.	Secure, but may require additional setup.	Secure, with built-in connectors for Azure services.
Error Handling & Retries	NO Built-in retry and error handling.	Must be manually configured for retries.	Built-in, but may not be ideal for batch uploads.
Suitability for Task	Most suitable for daily photo uploads.	Not optimized for bulk, repetitive tasks.	Good for workflows but not efficient for bulk uploads.
Side service requirement	None, direct upload from VM to BLOB	The need for an intermediary (File Share or FTP)	Azure Functions as an intermediary

GROUP # 1
SCHOLARSHIP
AZ-104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 3: **Process Automation**

Sub-project security simplification overview >> SAS Token vs. RBAC

Feature	SAS Token	Managed Identity (RBAC)
Validity Duration	Limited to a specific time period	Permanent (as long as the VM is active)
Management Complexity	Requires periodic renewal	Initial setup requires IAM configuration
Security	Temporary access with restricted permissions	Full and centralized control via Azure IAM
Best for this Scenario	Yes (for simplicity and time-limited access)	Only if long-term, managed access is needed

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 4: **Monitoring, Log Analytics and Alerting**

Azure Monitor is a comprehensive service for collecting, analyzing, and monitoring resources in Azure and hybrid environments. It gathers data from various resources (such as VMs, apps, storage, etc.) and provides you with insights for monitoring, reporting, and analysis.

A **Log Analytics Workspace** in Azure is a storage space where log and metric data is collected. It serves as a centralized hub for managing and analyzing monitoring data across your resources.

To monitor resources in a specific region, you first need to create a **Log Analytics Workspace**. So in the resource page, select Diagnostics Settings, and enable *Send to Log Analytics* and Choose the workspace you created.

To get notified about important events or issues in your resources, you can set up Alerts in **Azure Monitor**. Alerts can notify you about high resource usage, errors, or abnormal conditions.

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

PROJECT sec # 4: **Monitoring, Log Analytics and Alerting**

Monitoring Tool	Key Features	Advantages	Disadvantages
Azure Monitor	Centralized monitoring service for Azure resources.	Provides insights for performance and issues	Costs may increase with large-scale data collection
Log Analytics Workspace	Data collection hub for log and metric data from resources.	Centralized log storage and analysis	Costs related to data retention and storage
Azure Application Insights	Application performance monitoring and diagnostics.	Deep insights for applications	Can be complex for users unfamiliar with logs
Alerts (rules/actions)	Configuration of automated alerts based on conditions like high resource usage or errors.	Immediate notification of issues	Can generate excessive alerts if not configured properly

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

Cost estimate:

#	Resource	Purpose	Region	Min. Cost (monthly-USD)
1	VM	Germany access to webApp and photo upload	Germany	$25 – $70
2	Log analytics w/s	Gather and monitor diagnostics, log and metrics	Canada	$2 – $5
3	Network PEERing	Data transfer between regions	Germany / Canada	$0.50 – $1
4	Storage Acc.	Store files	Canada	$0.20 – $0.50
5	Automation Acc.	Automate processes	Germany / Canada	$0 – $10
6	App service	Hosting, Deployment and Management of web app	Canada	$70 – $120
			Min.Total	~ $100

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

Additional cost management by resource life-cycle considerations: (Preview only)

Resource	***Life Cycle Stages***	***Management Recommendations***
Virtual Machine (VM)	- Creation: Configure size, disk, and network. - Operation: Start, stop, restart, or update. - Deletion: Associated disks may persist and need manual removal.	- Enable Auto-Shutdown to reduce costs. - Regularly monitor resource usage. - Use snapshots for backup.
Storage Account	- Creation: Choose redundancy (LRS, GRS, ZRS) and access tiers. - Operation: Manage Blobs and access policies. - Deletion: Deletes all data unless backed up or replicated.	- Apply Retention Policies for versioning and recovery. - Use GRS for regional data recovery. --Manage data tiers as required
App Service	- Creation: Configure service plan, network settings, and deployment slots. - Operation: Scale or stop app services. - Deletion: Deletes app and configurations.	- Configure Auto-Scaling to handle high demand. - Use Deployment Slots for staging and production.
Log Analytics Workspace	- Creation: Associate workspace with VM or App Service. - Operation: Log ingestion and analysis. - Deletion: Removes all log data permanently.	- Define a Retention Period to manage data storage. - Ensure workspace is linked to active resources.
Automation Account	- Creation: Setup Runbooks and Hybrid Workers. - Operation: Monitor Job executions and script updates. - Deletion: Deletes all jobs and Runbooks.	- Regularly test and update Runbooks. - Use version control for Runbooks. - Monitor job history for errors.

Life Cycle Management exemptions: VNet, NSG, Private Endpoint

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

Complimentary research: **Deployment Automation using Bicep and Visual studio code**

Bicep is a declarative Infrastructure as Code (IaC) language for deploying Azure resources. It simplifies the deployment and management of Azure resources through clean and reusable code.

- Streamlined Resource Management
- Cost-Effective and Time-Saving
- Flexibility and Scalability
- Readability and Reusability

Why *Visual Studio Code*?

Visual Studio Code (*VS Code*) is an excellent IDE for writing and deploying Bicep files due to its robust support for extensions, debugging, and Azure integration.

- Bicep Extension for VS Code
- Azure CLI/PowerShell Integration
- Resource Template Management

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

Complimentary research: **Deployment Automation using Bicep and Visual studio code**

Steps to Use Bicep with Visual Studio Code:

- Install VS Code
- Install the Bicep Extension
- Set Up Azure CLI
- Write and Validate Bicep Files
- Deploy from VS Code Terminal

Why Use VS Code for Bicep in This Project?

- **Ease of Use:** Simplifies writing and deploying complex resource definitions.
- **Error Reduction:** Real-time validation reduces manual errors.
- **Seamless Azure Integration:** Directly deploy and test configurations without leaving the IDE.

LETS HAVE A QUICK VIEW

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024

PROJECT TITLE

Private and Local .NET Application with Automated Photo Upload Scenario

Thank you for supporting this presentation by your presence

Q&A

GROUP # 1
SCHOLARSHIP
AZ–104

Microsoft MSFarsi Community

NOV 30
2024