Azure Fundamentals — Sesiunea 8

Unde Trăiesc Datele Tale în Cloud?

Bine ați revenit! Astăzi descoperim un capitol esențial al oricărei arhitecturi cloud serioase: **Azure Storage**. Vom înțelege de ce serverul nu este locul potrivit pentru date, cum funcționează un Storage Account și cum protejăm ceea ce contează cu adevărat.

SESIUNEA 8 AZURE STORAGE FUNDAMENTALS

Agenda Sesiunii de Astăzi

Iată prin ce vom trece împreună în această sesiune. Fiecare parte construiește peste cea anterioară, de la concept la practică.

01

Serverul vs. Storage-ul

De ce nu ținem datele pe VM și care este principiul fundamental al separării compute de storage.

02

Ce este un Storage Account?

Blob Storage, File Share, Queue Storage — cele trei tipuri principale și analogiile lor din lumea reală.

03

Practică: Creăm Storage Account

Configurare pas cu pas în Azure Portal, cu discuție despre redundanță (LRS, ZRS, GRS).

04

Securitate: Public vs. Private

Una dintre cele mai frecvente cauze de scurgeri de date în cloud — și cum o evităm.

05

File Share și Queue Storage

Scenarii practice de utilizare și cum se integrează în arhitecturi moderne.

06

Cleanup și Tema pentru Acasă

Curățarea resurselor și cele trei întrebări de consolidare pentru acasă.

Întrebarea Esențială Pe Care Nu Am Pus-O Încă

Până acum am construit compute — mașini virtuale, web apps. Am configurat rețele și reguli de securitate. Dar am ignorat o întrebare fundamentală:

> Ce se întâmplă cu datele când serverul se oprește?

Dacă șterg o mașină virtuală pe care am lucrat săptămâni întregi — fișierele, configurațiile, documentele, imaginile salvate direct pe VM — **dispar odată cu mașina?**

Răspunsul, în majoritatea cazurilor, este **da**. Dacă datele sunt stocate doar pe discul mașinii virtuale, atunci când mașina dispare, dispar și ele. Aceasta este una dintre cele mai periculoase greșeli în arhitectura cloud: să confunzi *locul unde procesezi* cu *locul unde păstrezi*.

Principiul Fundamental: Serverul Procesează, Storage-ul Păstrează

🏢 VM-ul este Biroul

Biroul este locul unde angajații lucrează, fac calcule, scriu rapoarte, iau decizii. Este spațiul activ, dinamic, plin de energie — dar volatil. Dacă biroul arde, este o tragedie. Activitatea se oprește.

VM-ul funcționează exact la fel: procesează cereri, rulează cod, execută instrucțiuni. Dar când se oprește sau este șters, tot ce era stocat *local* pe el dispare.

🗄️ Storage-ul este Arhiva

Arhiva este locul unde documentele importante sunt stocate în siguranță — contracte, facturi, istoricul întregii activități. **Dacă și arhiva arde, firma pierde totul.**

Storage-ul în cloud este construit exact pentru durabilitate: este separat de compute, redundant, persistent. Datele rămân acolo indiferent ce se întâmplă cu serverele care le procesează.

Le construim separat. Le gestionăm separat. Le protejăm separat.

Regula de aur a arhitecturii cloud: Niciodată nu stoca date importante exclusiv pe discul local al unui VM. Folosește un serviciu de stocare dedicat, independent de ciclul de viață al serverului.

Ce Este un Storage Account în Azure?

Un **Storage Account** în Azure este un container mare — un fel de depozit central — care ține datele tale în cloud. Nu este un server. Nu procesează nimic. Pur și simplu *stochează, organizează și pune la dispoziție* datele ori de câte ori este nevoie, de oriunde din lume.

🗂️ Blob Storage

Fișiere nestructurate de orice tip: imagini, video, backup-uri, arhive. Cel mai folosit serviciu din familia Azure Storage.

📁 File Share

Echivalentul unui folder partajat în rețea, montat ca drive pe Windows sau Linux. Ideal pentru configurații și fișiere comune.

📬 Queue Storage

Mesagerie asincronă între componente ale unei aplicații. Decuplează serviciile și le permite să lucreze independent.

🔲 Table Storage

Stocare NoSQL pentru date semi-structurate. Îl menționăm astăzi, dar nu îl detaliem — va reveni în sesiunile despre baze de date.

Blob Storage — Depozitul Universal

Ce este Blob Storage?

Blob vine de la *Binary Large Object*. Este locul unde stochezi fișiere neorganizate, obiecte de orice fel — indiferent de format sau dimensiune. Gândiți-vă la un depozit mare cu cutii: nu contează ce este în cutie, nu contează ordinea. Pui cutia, primești o adresă unică (un URL), și o poți accesa oricând, de oriunde.

Ce se stochează în Blob?

- Avataruri și fotografii de profil ale utilizatorilor
- Documente uploadate de clienți (PDF, Word, Excel)
- Imagini de produs pentru magazine online
- Fișiere video și audio pentru platforme media
- Backup-uri automate ale bazelor de date
- Log-uri ale aplicațiilor și sisteme de monitorizare
- Artefacte de build și pachete de deployment

Blob Storage este **serviciul cel mai folosit din toată familia Azure Storage**. Aproape orice aplicație modernă stochează ceva în Blob.

Acces prin URL: Fiecare fișier (blob) are un URL unic, accesibil prin HTTP/HTTPS. Aplicațiile pot citi și scrie direct prin acest URL.

File Share — Folderul Partajat în Cloud

Conceptul din Spatele File Share

File Share este echivalentul unui **drive de rețea** pe care toți angajații unei firme îl văd și îl accesează — același concept, dar în cloud, accesibil de oriunde, de pe orice dispozitiv.

Poate fi montat ca o unitate de rețea pe **Windows** (drive G:, H: etc., vizibil în File Explorer) sau ca un director în **filesystem-ul Linux**. Aplicațiile pot citi și scrie fișiere în el *ca și cum ar fi un folder local* — fără să știe că de fapt vorbesc cu Azure Storage.

Când Folosim File Share?

Un caz de utilizare tipic și relevant: o companie are **zece servere** care trebuie să partajeze fișiere de configurare. În loc să copieze fișierele pe fiecare server separat (cu riscul de a le desincroniza), montează același Azure File Share pe toate serverele.

O singură sursă de adevăr, accesibilă de oriunde. O modificare în File Share se reflectă imediat pe toate serverele care îl montează.

- Fișiere de configurare partajate între servere
- Share-uri de fișiere pentru aplicații legacy care nu pot fi migrate la Blob
- Medii de dezvoltare cu fișiere comune între developeri

Queue Storage — Comunicare Asincronă

Analogia Cutiei Poștale

Imaginați-vă o cutie poștală. **Aplicația A** pune un mesaj în cutie. **Aplicația B** vine mai târziu, găsește mesajul și îl procesează. Cele două aplicații nu trebuie să funcționeze simultan. Nu trebuie să comunice direct una cu cealaltă.

Povestea Magazinului Online

Când un client plasează o comandă, se întâmplă mai multe lucruri: se trimite un email de confirmare, se scade produsul din stoc, se generează o factură, se notifică depozitul. Dacă toate s-ar întâmpla simultan, sistemul ar fi fragil — un email care durează prea mult ar bloca toată comanda.

Cu Queue Storage, aplicația principală pune pur și simplu un mesaj în coadă: *"a venit o comandă nouă"*. Fiecare serviciu vine la propria lui viteză, citește mesajul și procesează. **Aplicația principală nu așteaptă pe nimeni.**

Conceptul cheie: Queue Storage *decuplează* componentele unei aplicații și le permite să comunice fără să depindă una de cealaltă în timp real. Sistemul devine rezistent și scalabil.

⏸️ Pauză de 10 Minute

Pauză

Luăm o pauză de zece minute. La întoarcere, trecem la practică și creăm primul nostru Storage Account în Azure Portal.

📌 Recap rapid

VM = procesare temporară. Storage = persistență permanentă.

🗂️ Blob

Fișiere orice tip, acces prin URL.

📁 File Share

Drive de rețea partajat în cloud.

📬 Queue

Mesaje asincrone între servicii.

Practică: Creăm Storage Account — Pas cu Pas

Bine ați revenit! Acum trecem la practică. Urmați pașii de mai jos în Azure Portal.

1 Creați un Resource Group Nou

Mergeți la **Resource Groups → Create**. Numiți-l `rg-s08-[prenumele_vostru]`. Selectați regiunea **West Europe**. Dați Review and Create, apoi Create.

2 Creați Storage Account

În Azure Portal, mergeți la **Create a Resource** și căutați *Storage Account*. Completați câmpurile:

- **Name:** `s08storage[prenumele]` — fără cratime, fără spații, doar litere mici și cifre. Trebuie să fie unic la nivel global!
- **Region:** West Europe
- **Performance:** Standard
- **Redundancy:** LRS (Locally Redundant Storage)

3 Review and Create

Dați **Review and Create**, verificați configurația în pagina de sumar, și dați **Create**. Deployment-ul durează câteva secunde. Așteptați mesajul *"Your deployment is complete"*.

Redundanță în Azure Storage — Cât de Sigure Sunt Datele?

La crearea Storage Account-ului, ați ales opțiunea **Redundancy**. Aceasta determină câte copii ale datelor voastre păstrează Azure și unde sunt stocate. Cu cât redundanța este mai mare, cu atât datele sunt mai protejate — și cu atât costul este mai mare.

LRS — Locally Redundant Storage

3 copii ale datelor în *același datacenter*. Dacă un disc fizic se defectează, datele sunt recuperate automat din copia de rezervă, fără nicio intervenție din partea voastră. **Cel mai ieftin**. Potrivit pentru date necritice sau scenarii de dezvoltare/testare.

ZRS — Zone Redundant Storage

3 copii ale datelor în *trei zone de disponibilitate diferite* din aceeași regiune. Dacă un datacenter întreg cade, datele rămân disponibile din celelalte zone. Recomandat pentru aplicații de producție care necesită **High Availability** în cadrul aceleiași regiuni.

GRS — Geo Redundant Storage

6 copii ale datelor — 3 în regiunea primară și 3 în *o regiune geografică secundară* (de exemplu, West Europe + North Europe). Dacă o catastrofă naturală afectează o regiune întreagă, datele sunt recuperabile. **Cel mai scump**, dar oferă protecție maximă pentru date critice de business.

Pentru exercițiul de astăzi: LRS este mai mult decât suficient. Într-un mediu de producție real, alegerea redundanței depinde de cât de critice sunt datele și care este RTO/RPO acceptat de business (Recovery Time Objective / Recovery Point Objective).

Practică: Blob Container și Primul Upload

Acum că avem Storage Account-ul creat, hai să creăm primul container Blob și să uploadăm un fișier.

1

Navigați la Containers

În Storage Account, în meniul din stânga, găsiți secțiunea **Containers** și dați click pe ea.

2

Creați Container Nou

Dați + **New Container**. La Name: `s08-container-[prenumele]`. La Public access level: **Private**. Confirmați.

3

Upload Fișier

Intrați în container. Selectați **Upload**, alegeți un fișier text sau o imagine de pe calculatorul vostru, și confirmați uploadul.

4

Verificați Rezultatul

Fișierul vostru este acum **în cloud**. Stocat în Azure, redundant în trei copii, accesibil de oriunde printr-un URL unic.

Observați că fișierul uploadat are un URL complet, de forma: `https://[storageaccount].blob.core.windows.net/[container]/[fisier]`. Vom explora imediat ce se întâmplă cu accesul la acest URL.

⚠️ Public vs. Private — Una Dintre Cele Mai Periculoase Setări

Aceasta este una dintre cele mai importante discuții despre securitate în storage. Urmați pașii de mai jos și observați cu atenție ce se întâmplă.

Pasul 1: Schimbați în Public

Navigați la containerul vostru. Găsiți opțiunea de a schimba nivelul de acces. Schimbați temporar din **Private** în **Blob** (permite acces public la fișierele individuale).

Pasul 2: Copiați URL-ul

Selectați fișierul uploadat și copiați **URL-ul** său. Deschideți un tab nou în browser și lipiți acel URL.

Pasul 3: Observați

Oricine cu acel link poate vedea fișierul vostru. **Fără autentificare. Fără parolă. Fără niciun fel de verificare.** Pur și simplu accesând URL-ul.

Pasul 4: Reveniți la Private

Schimbați înapoi la **Private**. Verificați că URL-ul nu mai funcționează pentru acces anonim. Aceasta este setarea corectă și sigură.

Scurgerile de Date din Cloud: Un Risc Real

De Ce Contează Această Setare?

Dacă un Storage Account cu date sensibile este setat accidental la **Public**, oricine care găsește sau *ghicește* URL-ul poate descărca acele date — fără nicio autentificare, fără nicio urmă în log-uri de alertă.

Aceasta nu este o vulnerabilitate teoretică. Este una dintre **cele mai frecvente cauze de scurgeri de date în cloud**. Companii mari au expus accidental milioane de înregistrări din cauza unui singur container Blob setat la Public.

URL-urile de Blob Storage urmează un pattern predictibil. Un atacator care știe numele companiei poate enumera și testa containere. Un container Public înseamnă acces imediat.

Regula de Bază în Producție

Containerele Blob sunt **Private by default** și rămân Private dacă nu aveți un motiv explicit și *documentat* să le faceți publice.

Scenarii legitime pentru acces public:

- Imagini de produs pentru un site de e-commerce (date publice prin natura lor)
- Fișiere statice pentru un website (CSS, JS, fonturi)
- Documente publice destinate descărcării libere

Dacă faceți un container public, **asigurați-vă că știți exact ce date sunt în el**. Nicio dată personală, niciun fișier de configurare, niciun secret sau credential.

Alternativa sigură pentru acces temporar: **SAS Tokens** (Shared Access Signatures) — URL-uri cu expirare automată și permisiuni granulare.

File Share și Queue Storage

Explorăm **File Share** și **Queue Storage** — cele două servicii rămase din familia Azure Storage.

✅ Am acoperit

Storage Account, Blob Container, Upload, Public vs. Private, Redundanță (LRS/ZRS/GRS).

⏭️ Urmează

File Share (drive de rețea în cloud) și Queue Storage (mesagerie asincronă între servicii).

Practică: Creăm un File Share

Bine ați revenit! Explorăm acum **File Shares** în Azure Storage.

Pași de Creare

1 Navigați la File Shares

În Storage Account-ul vostru, în meniul din stânga, găsiți **File Shares** și dați click pe el.

2 Creați File Share Nou

Selectați + **New File Share**. La Name, scrieți `s08-fileshare-[prenumele]`. Confirmați crearea.

3 Explorați Opțiunile

Observați opțiunile de **Connect** — Azure vă oferă automat scriptul de PowerShell sau bash pentru montarea ca drive de rețea.

Diferența față de Blob

Blob Storage este pentru fișiere pe care le accesezi prin *HTTP, prin URL, printr-o aplicație*. Este optimizat pentru acces web și volume mari de date nestructurate.

File Share este pentru situații în care vrei să montezi un folder ca un *drive de rețea* — vizibil în File Explorer pe Windows sau în filesystem pe Linux. Aplicațiile pot scrie și citi fișiere în el fără să știe că de fapt vorbesc cu Azure Storage.

Protocolul utilizat: **SMB (Server Message Block)** — același protocol folosit de Windows pentru share-uri de rețea tradiționale. Aplicațiile legacy care folosesc deja SMB pot fi migrate în cloud *fără modificări de cod*.

Practică: Creăm un Queue Storage

Pași de Creare

În Storage Account-ul vostru, găsiți **Queues** în meniu. Dați + **Queue** și introduceți numele: `s08-queue-[prenumele]`. Confirmați.

Ce Vedeți în Portal

Queue-ul creat este inițial gol. În mod normal, mesajele sunt adăugate de aplicații prin codul lor, nu manual. Puteți adăuga manual un mesaj de test prin butonul **Add Message** pentru a vizualiza cum funcționează.

Structura unui Mesaj

Un mesaj în Queue Storage este simplu: conține text (de obicei JSON) cu informații despre evenimentul care s-a produs — de exemplu:

```
{"orderId": "12345", "customerId": "C789", "total": 249.99, "timestamp": "2024-01-15T14:30:00Z"}
```

Fiecare serviciu downstream (email, stoc, facturare) citește mesajul și procesează independent, la propria viteză.

Retenție mesaje: Mesajele din Queue Storage expiră după maxim 7 zile dacă nu sunt procesate. Serviciul garantează că fiecare mesaj este livrat *cel puțin o dată* (at-least-once delivery).

Cleanup: Ștergem Resursele

Ultima operație a sesiunii de astăzi: curățarea resurselor pentru a evita costuri inutile.

01

Navigați la Resource Group

Mergeți la **Resource Groups** și găsiți grupul `rg-s08-[prenumele_vostru]`.

02

Inițiați Ștergerea

Selectați **Delete Resource Group** din meniul de sus. Azure va cere confirmarea — introduceți exact numele resource group-ului în câmpul de confirmare.

03

Confirmați și Așteptați

Dați **Delete**. Procesul poate dura câteva minute. Toate resursele din grup — Storage Account, containere, date — vor fi șterse.

Important: Azure aplică o perioadă scurtă de retenție înainte ca datele să fie definitiv șterse. Nu vă faceți griji dacă resursa dispare mai lent decât VM-urile — este comportament normal și intenționat. Această protecție împiedică ștergerile accidentale ireparabile.

Tema pentru Acasă — Sesiunea 8

Trei întrebări de consolidare. Răspunsurile trebuie să demonstreze că ați înțeles conceptele, nu doar că le-ați memorat. Folosiți **analogii proprii**, diferite de cele discutate astăzi.

1

Blob, File Share și Queue

Explicați diferența dintre cele trei tipuri de stocare. Folosiți câte o **analogie proprie** pentru fiecare — diferită de depozit cu cutii, drive de rețea și cutie poștală. Gândiți-vă la situații din viața de zi cu zi sau din experiența voastră profesională.

2

De Ce Nu Pe Discul VM-ului?

De ce nu ținem datele importante exclusiv pe discul local al unei mașini virtuale? Descrieți **cel puțin două scenarii concrete** în care această decizie greșită ar duce la pierdere de date. Ce principiu arhitectural violăm?

3

Ce Este LRS și De Ce Contează?

Definiți **LRS** (Locally Redundant Storage) și explicați mecanismul din spatele lui. De ce este important pentru o companie care nu vrea să piardă date? Când ar trebui o companie să aleagă ZRS sau GRS în locul LRS?

Recapitulare: Ce Am Construit Împreună

Priviți ce ați acumulat în aceste opt sesiuni. Nu sunt servicii izolate — sunt componente ale unui **puzzle coerent**. O aplicație reală are nevoie de toate piesele.

1

🌐 Rețea (S4–S5)

VNet, Subnet, NSG, porturi, reguli de securitate, izolare și conectivitate controlată.

2

💻 Compute (S6–S7)

Mașini virtuale (IaaS), Web Apps (PaaS), scalare, disponibilitate, alegerea modelului potrivit.

3

🗄️ Storage (S8 — Astăzi)

Blob, File Share, Queue, redundanță (LRS/ZRS/GRS), securitate (Public vs. Private), separarea compute de stocare.

Piesa lipsă: Datele structurate. Bazele de date. Acesta este subiectul sesiunii 9 — și va completa imaginea de ansamblu a unei arhitecturi cloud complete.

Privire spre Sesiunea 9 — Baze de Date în Cloud

Ați completat astăzi un alt capitol important. Dar mai lipsește ceva esențial: **datele structurate**. Bazele de date.

Ce Urmează în Sesiunea 9?

În sesiunea nouă facem un **click mental important**: diferența dintre o bază de date instalată pe o mașină virtuală — pe care tu o administrezi complet (instalare, patch-uri, backup-uri, disponibilitate) — și o bază de date **managed**, în care Azure gestionează toată infrastructura pentru tine.

Este o decizie de arhitectură cu implicații majore asupra costurilor, timpului de management și riscurilor operaționale. Și acum știți deja cum să gândiți astfel de decizii — folosind același framework IaaS vs. PaaS pe care l-am explorat la compute.

Întrebări Cheie pentru S9

- Azure SQL Database vs. SQL pe VM — care este diferența reală?
- Ce înseamnă un serviciu *fully managed* și la ce renunți în schimb?
- Cum se conectează o aplicație la o bază de date în cloud?
- Backup automat, geo-replication, scaling — cum funcționează în managed databases?

Veți vedea cum **toate piesele — rețea, compute, storage, baze de date** — se asamblează într-o arhitectură completă, production-ready.

Vă mulțumesc pentru participare și pentru implicare! Ne vedem la **Sesiunea 9**. 🚀