

Azure Cloud Fundamentals

Ghid practic de laborator

Implementarea unei aplicații web multi-tier pe Azure

Web App + Azure SQL Database + Storage Account

Setare	Valoare
Modul	Web Apps, Azure SQL și Storage
Nivel	Fundamental (aliniat AZ-900)
Timp estimat	90 – 120 minute
Cost estimat	< 1 € dacă resursele sunt șterse în aceeași zi (SKU-uri Free + Serverless)
Regiune	West Europe
Metodă de implementare	Azure Portal (click-through)

1. Obiectivul laboratorului

În acest laborator practic veți implementa o aplicație web multi-tier completă pe Microsoft Azure, folosind Azure Portal. La sfârșitul laboratorului veți avea construite următoarele componente într-un singur grup de resurse:

- Un **Azure App Service Plan** bazat pe Linux (nivelul de calcul)
- Un **Azure App Service** (web app) care rulează .NET pe Linux
- Un **Azure SQL logical server** cu admin Microsoft Entra ID și reguli de firewall
- Un **Azure SQL Database** pe nivelul Serverless (Gen5)
- Un **Azure Storage Account** (StorageV2) cu un container blob pentru fișiere statice
- Un tabel de bază de date **populat cu date demo despre studenți**

Ce veți învăța

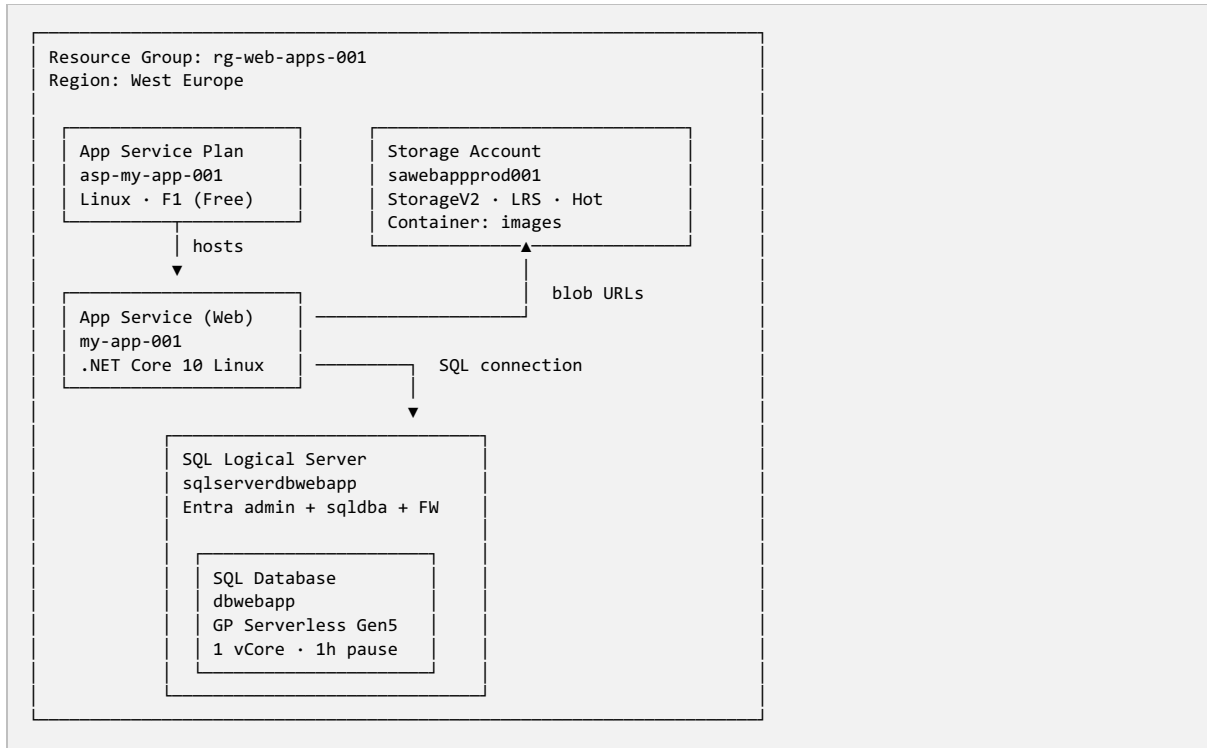
- Cum se planifică și se denumesc resursele Azure folosind Microsoft Cloud Adoption Framework (CAF)
- Cum se creează și se organizează resursele într-un Azure Resource Group
- Cum se configurează compute PaaS (App Service), data PaaS (Azure SQL) și storage PaaS
- Cum se securizează un Azure SQL logical server cu reguli de firewall și autentificare Entra ID
- Cum se rulează T-SQL împotriva unei baze de date Azure SQL folosind Query Editor încorporat
- Cum se identifică compromisurile uzuale de securitate și cost pe parcurs

NOTĂ

Acest laborator folosește nivelul Free (F1) pentru App Service și SQL Serverless cu auto-pauză de 1 oră, pentru a menține costurile aproape de zero. Acestea NU sunt niveluri de producție — diferențele sunt evidențiate în Secțiunea 8.

2. Prezentare generală a arhitecturii

Laboratorul implementează cinci resurse Azure într-un singur grup de resurse. Diagrama de mai jos arată cum sunt acestea legate logic.



Responsabilitățile componentelor

Componentă	Rolul în arhitectură
App Service Plan	Definește capacitatea de calcul (OS, regiune, SKU, număr de instanțe) pe care rulează una sau mai multe aplicații web.
App Service (Web)	Găzduiește codul aplicației web (.NET). Expune un endpoint HTTPS public pe *.azurewebsites.net.
SQL Logical Server	Limita de securitate și conexiune pentru una sau mai multe baze de date SQL. Conține regulile de firewall, administratorii și setările de auditing.
SQL Database	Baza de date relațională efectivă, care stochează datele aplicației (tabela Students în acest laborator).
Storage Account	Stochează date nestructurate (imagini, loguri, documente) în containere blob. Oferă endpoint-uri HTTP(S) publice sau private.

3. Cerințe prealabile

Înainte de a începe, asigurați-vă că aveți următoarele:

- **Un abonament Azure activ** cu cel puțin rolul **Contributor** la nivel de abonament sau pe un grup de resurse dedicat. Un abonament gratuit sau pay-as-you-go este suficient.
- **Un browser web modern** (Microsoft Edge, Chrome sau Firefox) pentru a accesa <https://portal.azure.com>
- **Adresa dvs. publică IPv4 curentă.** Veți avea nevoie de aceasta pentru a crea regula de firewall a serverului SQL. O puteți găsi la <https://whatismyipaddress.com> sau căutând „what is my IP” în orice motor de căutare.
- **(Opțional)** Un cont de utilizator Microsoft Entra ID în același tenant cu abonamentul, pentru a fi configurat ca admin SQL Entra.
- **(Opțional)** SQL Server Management Studio (SSMS) sau Azure Data Studio, dacă preferați să vă conectați dintr-un client desktop în locul Query Editor din portal.

⚠️ AVERTISMENT

NU folosiți un abonament corporativ sau de producție pentru acest laborator. Folosiți un abonament de învățare/sandbox. Veți crea resurse accesibile public și excepții de firewall care sunt acceptabile pentru training, dar nu pentru medii de producție.

4. Convenții de denumire Azure (CAF)

Înainte de a crea orice, este important să urmați o convenție de denumire consecventă. Microsoft publică Cloud Adoption Framework (CAF), care oferă abrevierile recomandate și pattern-urile de denumire pentru fiecare tip de resursă Azure.

Pattern-ul general CAF este:

```
<resource-abbreviation>-<workload/app>-<environment>-<region>-<instance>
```

În acest laborator folosim un pattern simplificat, potrivit pentru training:

```
<resource-abbreviation>-<workload>-<instance>
```

Abrevieri folosite în acest laborator

Tip de resursă	Abreviere CAF	Exemplu (acest laborator)
Resource Group	rg	rg-web-apps-001
App Service Plan	asp	asp-my-app-001
App Service (Web)	app	my-app-001
SQL Logical Server	sql	sqlserverdbwebapp
SQL Database	sqldb	dbwebapp
Storage Account	sa	sawebappprod001

NOTĂ

Două tipuri de resurse au constrângeri de denumire care suprascriu CAF: Storage account-urile trebuie să fie unice global, 3–24 caractere, doar litere mici și cifre (fără cratime). Numele de SQL logical server trebuie să fie de asemenea unice global și conforme DNS (litere mici, cratime permise, fără underscore-uri). De aceea sawebappprod001 nu are cratime.

SFAT

Dacă vă construiți propriul laborator, înlocuiți sufixul de instanță (-001) sau adăugați-vă inițialele pentru a evita conflictele de unicitate globală pentru numele de Storage și SQL server. De exemplu: sawebappprod001 sau sqlserverdbwebappim.

5. Sumar al resurselor laboratorului

Folosiți tabelul de mai jos ca referință rapidă pentru toate valorile pe care le veți introduce în timpul implementării. Toate resursele sunt create în regiunea West Europe.

Resursă	Nume	Configurare cheie
Resource Group	rg-web-apps-001	Regiune: West Europe
App Service Plan	asp-my-app-001	OS: Linux · SKU: F1 (Free) · Zone redundant: Disabled
App Service (Web)	my-app-001	Runtime: .NET Core 10.0 · OS: Linux · Publishing: Code
SQL Logical Server	sqlserverdbwebapp	Auth: SQL + Entra · Admin: sqlldbadmin · Public access: Selected networks
SQL Database	dbwebapp	Tier: General Purpose Serverless (Gen5) · 1 vCore · Auto-pause: 1h
Storage Account	sawebappprod001	StorageV2 · LRS · Hot · ADLS Gen2 (hierarchical NS) · TLS 1.2
Blob Container	images	Anonymous access level: Blob

6. Implementare pas cu pas

Parcurgeți secțiunile de mai jos în ordinea indicată. Resursele ulterioare depind de cele anterioare (de exemplu, Web App-ul are nevoie de App Service Plan, iar SQL Database are nevoie de SQL Server).

6.1 Crearea Resource Group-ului

Un resource group este un container logic pentru toate resursele din acest laborator. Ștergerea grupului de resurse la sfârșit va șterge fiecare resursă din interiorul său — o metodă curată și rapidă de cleanup.

1. Conectați-vă la Azure Portal la <https://portal.azure.com>.
2. În bara de căutare de sus, scrieți **Resource groups** și selectați-l din lista de servicii.
3. Apăsați **+ Create** în partea de sus.
4. Pe tab-ul **Basics**, introduceți:
 - **Subscription:** selectați abonamentul de laborator
 - **Resource group:** `rg-web-apps-001`
 - **Region:** (Europe) West Europe
5. Apăsați **Review + create**, apoi **Create**.
6. Așteptați finalizarea deployment-ului (de obicei 10–20 secunde) și apăsați **Go to resource group**.

SFAT

Regiunea aleasă pentru resource group este doar metadata — stochează informațiile RG-ului în sine. Fiecare resursă din interior poate fi tehnic într-o regiune diferită. Pentru acest laborator, păstrați fiecare resursă în West Europe, ca să fie co-localizate și să poată comunica cu cea mai mică latență și fără cost de egress.

6.2 Crearea App Service Plan-ului

App Service Plan-ul definește nivelul de calcul (CPU, memorie, OS) pe care vor rula aplicațiile dvs. web. Vom folosi nivelul Free F1, care este ideal pentru învățare, dar nu are SLA, are minute CPU limitate pe zi și nu are Always On.

1. În bara de căutare de sus, căutați **App Service plans** și selectați-l.
2. Apăsați **+ Create**.
3. Pe tab-ul **Basics**, introduceți:
 - **Subscription:** abonamentul dvs. de laborator
 - **Resource group:** `rg-web-apps-001`
 - **Name:** `asp-my-app-001`
 - **Operating System:** **Linux**
 - **Region:** West Europe
 - **Zone redundancy:** **Disabled** (nu este disponibil pe F1)
4. În secțiunea **Pricing plan**, apăsați **Explore pricing plans** (sau **Change size**).
5. Selectați tab-ul **Dev / Test** și alegeți **F1 — Free**. Apăsați **Select**.
6. Apăsați **Review + create**, apoi **Create**. Deployment-ul durează 30–60 secunde.

AVERTISMENT

Nivelul F1 are limitări stricte: 60 minute CPU pe zi, 1 GB memorie, fără domenii personalizate cu SSL, fără Always On, fără deployment slots, fără auto-scaling. Nu puteți schimba un plan existent din Free în Basic/Standard fără a crea un plan nou.

6.3 Crearea Web App-ului

Web App-ul este instanța aplicației propriu-zise. Trăiește în interiorul App Service Plan-ului și expune un endpoint HTTPS public sub *.azurewebsites.net.

1. În bara de căutare de sus, căutați **App Services** și selectați-l.
2. Apăsați **+ Create** → **Web App**.
3. Pe tab-ul **Basics**, introduceți:
 - **Subscription:** abonamentul dvs. de laborator
 - **Resource group:** `rg-web-apps-001`
 - **Name:** `my-app-001` (URL-ul rezultat va fi `https://my-app-001.azurewebsites.net`)
 - **Publish:** **Code**
 - **Runtime stack:** **.NET 10 (LTS)** sau runtime-ul .NET Core curent
 - **Operating System:** **Linux**
 - **Region:** West Europe
 - **Linux Plan (West Europe):** `asp-my-app-001` (selectați planul creat la pasul 6.2)
 - **Pricing plan:** Free F1 (moștenit din plan)
4. Pe tab-ul **Database**, lăsați **Create a new database** nebifat — vom crea SQL-ul separat în secțiunea următoare.
5. Pe tab-ul **Deployment**, lăsați **Continuous deployment: Disabled**.
6. Pe tab-ul **Networking**, acceptați valorile implicite (acces public activat, fără integrare VNet pentru F1).
7. Pe tab-ul **Monitoring**, setați **Enable Application Insights: No** pentru acest laborator, ca să-l păstrăm simplu (îl puteți activa mai târziu).
8. Apăsați **Review + create**, apoi **Create**. Deployment-ul durează 60–120 secunde.
9. După finalizare, apăsați **Go to resource**, apoi apăsați butonul **Browse**. Ar trebui să vedeți pagina de bun venit implicită App Service.

NOTĂ

Numele Web App-ului trebuie să fie unic global în întregul Azure, deoarece devine parte din numele DNS azurewebsites.net. Dacă „my-app-001” este deja luat, adăugați-vă inițialele sau un sufix random (de exemplu my-app-im-001).

6.4 Crearea Azure SQL Logical Server și Database

Azure SQL Database este o bază de date relațională PaaS complet gestionată. Constă din două obiecte: logical server-ul (limita de securitate, firewall, conturi de admin) și baza de date propriu-zisă (datele și calculul).

6.4.1 Crearea SQL logical server + database împreună

Cea mai rapidă cale este blade-ul combinat de creare „SQL Database”, care creează ambele resurse într-un singur wizard.

1. În bara de căutare de sus, căutați **SQL databases** și selectați-l.
2. Apăsați **+ Create**.
3. Pe tab-ul **Basics**, introduceți:
 - **Subscription:** abonamentul dvs. de laborator
 - **Resource group:** `rg-web-apps-001`
 - **Database name:** `dbwebapp`
 - **Server:** apăsați **Create new** (vezi pasul 4 de mai jos)
 - **Elastic pool:** **No**
 - **Workload environment:** **Development**
 - **Compute + storage:** apăsați **Configure database** (vezi pasul 5 de mai jos)
 - **Backup storage redundancy:** **Locally-redundant backup storage**
4. În panoul **Create SQL Database Server** (se deschide din „Create new”):
 - **Server name:** `sqlserverdbwebapp` (unic global, litere mici, fără underscore-uri)
 - **Location:** West Europe
 - **Authentication method:** **Use both SQL and Microsoft Entra authentication**
 - **Microsoft Entra admin:** apăsați **Set admin** și selectați utilizatorul Entra (de exemplu contul `iulian.mihai@yourtenant.onmicrosoft.com`)
 - **SQL authentication login:** `sqldba`
 - **Password:** o parolă puternică (salvați-o în password manager — veți avea nevoie de ea mai târziu)
 - Apăsați **OK** pentru a reveni la blade-ul bazei de date.
5. În panoul **Configure database** (se deschide din „Configure database”):
 - **Service tier:** **General Purpose** (Scalable compute and storage)
 - **Compute tier:** **Serverless**
 - **Hardware configuration:** **Standard-series (Gen5)**
 - **Max vCores:** **1** (minimul permis)
 - **Min vCores:** **0.5**
 - **Auto-pause delay:** **1 hour** (baza de date intră în pauză după 1 oră de inactivitate, pentru a economisi bani)
 - **Data max size:** **32 GB** (implicitul este în regulă)

- Apăsați **Apply**.
6. Pe tab-ul **Networking**:
 - **Connectivity method: Public endpoint**
 - **Allow Azure services and resources to access this server: Yes**
 - **Add current client IP address: Yes** (creează o regulă de firewall pentru IP-ul dvs. public)
 7. Pe tab-ul **Security**, lăsați valorile implicite (Defender for SQL este oprit pentru a evita costuri suplimentare în acest laborator).
 8. Pe tab-ul **Additional settings**:
 - **Use existing data: None** (o vom popula manual cu scriptul T-SQL mai târziu)
 - **Collation: SQL_Latin1_General_CP1_CI_AS** (implicit)
 9. Apăsați **Review + create**, apoi **Create**. Deployment-ul durează 2–5 minute.

AVERTISMENT

Setarea „Allow Azure services and resources to access this server” permite oricărei resurse care rulează în Azure (din ORICE tenant Azure, nu doar al dvs.) să încerce o conexiune. Este convenabilă pentru învățare, dar reprezintă o problemă de securitate în producție. În medii reale, preferați Private Endpoints sau intervalele de IP outbound explicite ale App Service-ului.

6.4.2 Verificarea regulilor de firewall

1. Navigați la SQL server-ul `sqlserverdbwebapp` (nu la baza de date).
2. În meniul din stânga, apăsați **Security** → **Networking**.
3. Confirmați că **Public network access** este setat la **Selected networks**.
4. Sub **Firewall rules**, confirmați că există o regulă cu un nume similar cu `ClientIPAddress_<timestamp>` având IP-ul dvs. public atât ca Start, cât și ca End IPv4 address.
5. Sub **Exceptions**, confirmați că **Allow Azure services and resources to access this server** este bifat.
6. Apăsați **Save** dacă ați făcut modificări.

SFAT

IP-ul dvs. public se poate schimba (în special acasă sau pe rețele mobile). Dacă nu vă puteți conecta la baza de date mai târziu, reveniți aici și apăsați din nou „Add your client IPv4 address” — se va adăuga o regulă nouă pentru IP-ul curent.

6.5 Crearea Storage Account-ului

Azure Storage este serviciul fundamental pentru date nestructurate. Acest laborator creează un cont StorageV2 (general-purpose v2) cu hierarchical namespace activat, care îl transformă într-un cont Azure Data Lake Storage Gen2 — util pentru workload-uri de analytics și pentru semantica de tip file-system.

1. În bara de căutare de sus, căutați **Storage accounts** și selectați-l.
2. Apăsați **+ Create**.
3. Pe tab-ul **Basics**, introduceți:
 - **Subscription:** abonamentul dvs. de laborator
 - **Resource group:** `rg-web-apps-001`
 - **Storage account name:** `sawebappprod001` (litere mici, 3–24 caractere, unic global — adăugați un sufix dacă este luat)
 - **Region:** West Europe
 - **Primary service:** **Azure Blob Storage** sau **Azure Data Lake Storage Gen2**
 - **Performance:** **Standard** (mai ieftin, suficient pentru laborator)
 - **Redundancy:** **Locally-redundant storage (LRS)** (cea mai ieftină — păstrează 3 copii într-un singur datacenter)
4. Pe tab-ul **Advanced**:
 - **Require secure transfer for REST API operations:** **Enabled**
 - **Allow enabling anonymous access on individual containers:** **Enabled** (ca să putem face containerul images public-readable)
 - **Enable storage account key access:** **Enabled**
 - **Minimum TLS version:** **Version 1.2**
 - **Enable hierarchical namespace:** **Enabled** (transformă contul în ADLS Gen2)
 - **Access tier:** **Hot**
5. Pe tab-ul **Networking**:
 - **Network access:** **Enable public access from all networks**
 - **Network routing:** **Microsoft network routing**
6. Pe tab-ul **Data protection**:
 - **Enable soft delete for blobs:** **Enabled** (retention: 7 zile)
 - **Enable soft delete for containers:** **Enabled** (retention: 7 zile)
 - **Enable soft delete for file shares:** **Enabled** (retention: 7 zile)
 - **Enable versioning for blobs:** **Disabled** (nu este necesar pentru laborator)
7. Pe tab-ul **Encryption**, lăsați valorile implicite (chei gestionate de Microsoft).
8. Apăsați **Review + create**, apoi **Create**. Deployment-ul durează 30–60 secunde.

NOTĂ

Activarea hierarchical namespace transformă acest cont într-un cont Data Lake Storage Gen2. Nu îl puteți dezactiva după creare. Este compatibil cu API-urile standard Blob, dar suportă și directoare și ACL-uri în stil POSIX.

⚠️ AVERTISMENT

În acest laborator activăm acces anonim la blob la nivel de cont și vom permite acces anonim la nivel de Blob pe containerul images. Acest lucru este acceptabil DOAR pentru active cu adevărat publice, precum imaginile de pe website. Nu activați niciodată accesul anonim pentru ceva sensibil — în producție, preferați token-uri SAS, Managed Identity sau Private Endpoints.

6.6 Crearea Blob Container-ului

Containerele sunt gruparea de top-level pentru blob-uri într-un storage account — echivalent cu directoarele rădăcină. Vom crea un container images pentru fișiere statice publice.

1. Navigați la storage account-ul `sawebapprod001`.
2. În meniul din stânga, apăsați **Data storage** → **Containers**.
3. Apăsați **+ Add container** în partea de sus.
4. În panoul **New container**, introduceți:
 - **Name:** `images`
 - **Anonymous access level:** **Blob** (acces de citire anonim doar pentru blob-uri)
5. Apăsați **Create**.
6. Acum ar trebui să vedeți două containere în listă: `$logs` (creat automat de sistem, Private) și `images` (containerul nou creat, Blob access).

💡 SFAT

Containerul \$logs este creat automat de Azure Storage analytics pentru logging-ul de diagnosticare. Este întotdeauna privat și nu poate fi șters.

6.7 Popularea bazei de date cu date demo

Acum că baza de date este creată, dar goală, vom folosi Query Editor încorporat în portal pentru a rula un script T-SQL care creează o tabelă `Students` și inserează 12 rânduri demo.

1. Navigați la baza de date SQL `dbwebapp` (sub `sqlserverdbwebapp`).
2. Dacă baza de date afișează **Status: Paused**, este normal. Orice query o va relua automat (poate dura 30–60 secunde la prima cerere).
3. În meniul din stânga, apăsați **Query editor (preview)**.
4. Autentificați-vă. Puteți folosi una dintre opțiuni:
 - **SQL server authentication** cu Login `sqldba` și parola setată la pasul 6.4.1
 - **Microsoft Entra authentication** cu utilizatorul configurat ca Entra admin
5. Dacă primiți o eroare de firewall de tipul „`Your client IP address does not have access`”, apăsați link-ul pentru a-l adăuga, apoi reîncercați.
6. Odată conectat, veți vedea Query editor-ul cu o zonă de text. Lipiți întregul script T-SQL din **Anexa B** de la sfârșitul acestui document.
7. Apăsați **Run**. Așteptați trei rezultate:
 - Comenzile `DROP TABLE / CREATE TABLE` se finalizează fără output.
 - Comanda `INSERT` raportează 12 rânduri afectate.
 - `SELECT`-ul final returnează 12 rânduri în grila de rezultate.
8. Verificați rezultatul: ar trebui să vedeți coloanele `StudentId`, `FirstName`, `LastName`, `Email`, `CourseName`, `GPA` etc. pentru 12 studenți.

NOTĂ

Primul query după o pauză va fi lent (30–60 secunde) deoarece baza de date trebuie să se reia din starea de `paused`. Query-urile ulterioare în următoarea oră vor fi rapide. Acest comportament de `pause/resume` este unic pentru nivelul `Serverless`.

7. Verificare

Confirmați că totul este implementat și accesibil.

7.1 Verificarea resource group-ului

1. Navigați la resource group-ul `rg-web-apps-001`.
2. Ar trebui să vedeți exact 5 resurse:
 - `asp-my-app-001` — App Service plan
 - `dbwebapp (sqlserverdbwebapp/dbwebapp)` — SQL database
 - `my-app-001` — App Service
 - `sawebappprod001` — Storage account
 - `sqlserverdbwebapp` — SQL server

7.2 Verificarea Web App-ului

1. Deschideți `https://my-app-001.azurewebsites.net` într-un tab nou de browser.
2. Ar trebui să vedeți pagina implicită App Service („Your App Service app is up and running”).
3. Reveniți la App Service-ul `my-app-001` în portal și confirmați:
 - **Status: Running**
 - **App Service Plan: asp-my-app-001 (F1: 1)**
 - **Operating System: Linux**
 - **Runtime Stack: Dotnetcore - 10.0**

7.3 Verificarea datelor din SQL Database

1. Deschideți din nou **Query editor** pe `dbwebapp`.
2. Rulați următorul query:

```
SELECT COUNT(*) AS TotalStudents FROM dbo.Students;
```

3. Confirmați că rezultatul este **12**.
4. Rulați un al doilea query pentru a testa filtrarea:

```
SELECT FirstName, LastName, CourseName, GPA  
FROM dbo.Students  
WHERE IsActive = 1  
ORDER BY GPA DESC;
```

5. Ar trebui să vedeți 11 studenți activi ordonați descrescător după GPA (Mia Ruiz pe primul loc, cu 3.95).

7.4 Verificarea Storage Account-ului

1. Navigați la storage account-ul `sawebappprod001` → **Data storage** → **Containers**.
2. Apăsați containerul `images`.
3. Apăsați **Upload**, selectați un fișier mic de imagine (PNG/JPG) și încărcați-l.
4. Apăsați blob-ul încărcat, apoi copiați URL-ul afișat în proprietăți.
5. Lipiți URL-ul într-o fereastră nouă de browser în mod incognito/private.
6. Imaginea ar trebui să se încarce fără niciun prompt de autentificare — confirmând că accesul de citire anonim public funcționează la nivel de Blob.

8. Puncte de învățare: Laborator vs. Producție

Acest laborator prioritizează simplitatea și costul. Multe dintre alegerile făcute NU ar fi potrivite într-un workload de producție. Tabelul de mai jos compară ceea ce ați construit cu ceea ce ar arăta o versiune de nivel producție.

Aspect	Acest laborator	Recomandare pentru producție
SKU App Service	F1 Free (fără SLA, fără Always On)	P1v3 sau mai mare, cu Always On, deployment slots și autoscaling
Networking App Service	Doar endpoint public	VNet integration + Private Endpoints + Front Door/WAF
Auth admin SQL	SQL login (sqldba) + Entra admin	Doar autentificare Microsoft Entra (dezactivare SQL auth)
Acces de rețea SQL	Endpoint public + reguli de firewall	Private Endpoint într-un VNet, acces public dezactivat
SQL „Allow Azure services”	Activat (orice tenant Azure poate încerca să se conecteze)	Dezactivat — doar reguli de firewall explicite sau Private Endpoints
Compute tier SQL	General Purpose Serverless 1 vCore cu auto-pauză	Provisioned Business Critical sau Hyperscale, zone-redundant
Backup-uri SQL	Locally-redundant (LRS)	Geo-redundant (GRS) sau zone-redundant (ZRS)
Acces anonim Storage	Activat la nivel de cont + Blob pe images	Dezactivat — servire prin SAS tokens, CDN sau Managed Identity
Redundanță Storage	LRS (un singur datacenter)	ZRS sau GZRS pentru date business-critical
Acces de rețea Storage	Public din toate rețelele	Private Endpoint + allow-list de firewall pentru câteva IP-uri de încredere
Storage TLS	1.2 (bun)	1.2 minim; considerați 1.3 când e suportat de clienți
Gestiunea secretelor	Parola copiată-lipită, connection string DB în portal	Azure Key Vault + Managed Identity; nicio credențial hardcodată
Monitorizare	Niciuna	Application Insights + workspace Log Analytics + alerte

Aspect	Acest laborator	Recomandare pentru producție
IaC	Click-through în portal	Bicep / Terraform în Git, deployed via Azure DevOps sau GitHub Actions

Concluzii cheie de securitate

- **Responsabilitate partajată.** Azure securizează infrastructura fizică și platforma; DVS. sunteți responsabil pentru identitate, date, reguli de rețea și codul aplicației.
- **Implicit privat.** Fiecare endpoint public creat este o suprafață potențială de atac. În producție, Private Endpoints + VNet integration ar trebui să fie alegerea implicită.
- **Preferăți Entra în loc de SQL auth.** Autentificarea bazată pe Entra oferă conditional access, MFA, PIM și revocare centralizată. SQL login-urile nu au nimic din toate acestea.
- **Nu expuneți blob-urile anonim decât dacă sunt cu adevărat publice.** Întrebați-vă: este acest fișier OK să fie descărcat de oricine din lume? Dacă nu, folosiți SAS sau Managed Identity.
- **Etichetați-vă întotdeauna resursele.** În acest laborator am sărit peste tag-uri pentru a fi concisi, dar în viața reală etichetați fiecare resursă cu owner, environment, cost-center și workload.

9. Cleanup

Când ați terminat laboratorul, ștergeți întregul resource group pentru a opri toată facturarea.

1. Navigați la resource group-ul **rg-web-apps-001**.
2. În partea de sus a blade-ului Overview, apăsați **Delete resource group**.
3. În panoul de confirmare, scrieți numele resource group-ului (exact **rg-web-apps-001**) pentru a confirma.
4. Apăsați **Delete**. Ștergerea este asincronă și se finalizează de obicei în 2–5 minute.
5. După câteva minute, reîmprospătați lista de Resource Groups și confirmați că **rg-web-apps-001** a dispărut.

⚠️ AVERTISMENT

Ștergerea este PERMANENTĂ. Toate resursele, datele și backup-urile din resource group vor fi distruse. Dacă doriți să păstrați datele studenților, exportați-le (SELECT INTO sau export BACPAC) înainte de ștergere.

Anexa A — Considerații de cost

Costuri aproximative pentru rularea acestui laborator timp de 24 de ore în West Europe. Prețurile sunt orientative și se pot schimba; verificați întotdeauna calculatorul de prețuri Azure pentru regiunea dvs.

Resursă	SKU / nivel	Cost aproximativ
App Service Plan	F1 (Free)	0,00 € pe lună
App Service	—	Inclus în plan
SQL Server (logical)	—	0,00 € (fără cost pentru obiectul server în sine)
SQL Database	GP Serverless Gen5, 1 vCore, auto-pauză 1h	0 € când e în pauză; ≈ 0,50 € / oră activă; storage ≈ 0,10 € / GB / lună
Storage Account	StorageV2, LRS, Hot, ADLS Gen2	≈ 0,02 € / GB / lună + costuri tranzacționale (neglijabile pentru laborator)

SFAT

Folosiți blade-ul Cost Management + Billing pentru a seta o alertă de buget pe abonamentul de laborator. Setarea unui buget de 5 € cu alerte pe email la 50%, 80% și 100% este o practică bună. Studenții care lasă din neatenție resurse pornite vor primi un avertisment timpuriu.

Anexa B — Script T-SQL: Popularea tabelii Students

Copiați scriptul de mai jos și lipiți-l în Query editor (preview) al bazei de date dbwebapp. Rulați-l o singură dată. Șterge tabela dacă există deja (idempotent pentru re-rulări la clasă), o creează, inserează 12 rânduri demo și returnează un SELECT cu toate rândurile pentru verificare.

```
-- Folosiți baza de date corectă
-- În Azure SQL, asigurați-vă că sunteți conectat la dbwebapp înainte de a rula acest script

-- Ștergeți tabela întâi dacă doriți să o recreați în timpul testării
IF OBJECT_ID('dbo.Students', 'U') IS NOT NULL
BEGIN
    DROP TABLE dbo.Students;
END
GO

-- Crearea tabelii Students
CREATE TABLE dbo.Students
(
    StudentId          INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
    FirstName          NVARCHAR(50)     NOT NULL,
    LastName           NVARCHAR(50)     NOT NULL,
    DateOfBirth        DATE              NOT NULL,
    Email              NVARCHAR(100)    NOT NULL UNIQUE,
    PhoneNumber        NVARCHAR(20)     NULL,
    EnrollmentDate     DATE              NOT NULL,
    CourseName         NVARCHAR(100)    NOT NULL,
    YearLevel          INT               NOT NULL,
    GPA                DECIMAL(3,2)     NULL,
    IsActive           BIT               NOT NULL DEFAULT 1,
    CreatedAt          DATETIME2        NOT NULL DEFAULT SYSUTCDATETIME()
);
GO

-- Inserarea datelor demo despre studenți
INSERT INTO dbo.Students
(
    FirstName, LastName, DateOfBirth, Email, PhoneNumber,
    EnrollmentDate, CourseName, YearLevel, GPA, IsActive
)
VALUES
('Emma',      'Johnson',  '2003-04-15', 'emma.johnson@schooldemo.com',  '+34 600 111 001', '2023-09-01',
'Computer Science', 2, 3.75, 1),
('Liam',     'Martinez',  '2002-11-22', 'liam.martinez@schooldemo.com', '+34 600 111 002', '2022-09-01',
'Business Administration',3, 3.42, 1),
('Olivia',   'Garcia',    '2004-02-10', 'olivia.garcia@schooldemo.com',  '+34 600 111 003', '2024-09-01',
'Graphic Design',   1, 3.90, 1),
('Noah',     'Fernandez', '2001-07-30', 'noah.fernandez@schooldemo.com', '+34 600 111 004', '2021-09-01',
'Mechanical Engineering', 4, 3.21, 1),
('Sophia',   'Lopez',     '2003-09-05', 'sophia.lopez@schooldemo.com',  '+34 600 111 005', '2023-09-01',
'Psychology',       2, 3.88, 1),
('James',    'Sanchez',   '2002-05-18', 'james.sanchez@schooldemo.com',  '+34 600 111 006', '2022-09-01',
'Law',              3, 3.33, 1),
('Isabella', 'Perez',     '2004-01-27', 'isabella.perez@schooldemo.com',  '+34 600 111 007', '2024-09-01',
'Architecture',     1, 3.67, 1),
('Benjamin', 'Gomez',     '2001-12-03', 'benjamin.gomez@schooldemo.com',  '+34 600 111 008', '2021-09-01',
'Civil Engineering', 4, 3.11, 1),
('Mia',      'Ruiz',     '2003-06-12', 'mia.ruiz@schooldemo.com',       '+34 600 111 009', '2023-09-01',
'Data Science',     2, 3.95, 1),
('Lucas',    'Diaz',      '2002-03-09', 'lucas.diaz@schooldemo.com',      '+34 600 111 010', '2022-09-01',
'Marketing',        3, 3.48, 1),
('Charlotte', 'Hernandez', '2004-08-21', 'charlotte.hernandez@schooldemo.com', '+34 600 111 011', '2024-09-01',
'Medicine',         1, 3.80, 1),
('Henry',    'Moreno',   '2001-10-14', 'henry.moreno@schooldemo.com',    '+34 600 111 012', '2021-09-01',
'Finance',          4, 3.27, 0);
GO

-- Query de test
SELECT *
FROM dbo.Students
```

```
ORDER BY StudentId;  
GO
```

Rezultate așteptate

- Tabela este ștearsă dacă există, apoi este recreată.
- Mesajul **12 rows affected** pentru INSERT.
- SELECT-ul final returnează 12 rânduri în grila de rezultate, ordonate după **StudentId** de la 1 la 12.

Felicitări!

Ați implementat, configurat, verificat și securizat cu succes un stack complet de aplicație web Azure folosind doar Azure Portal. Acum aveți experiență practică cu:

- Organizarea resurselor prin resource groups și denumire CAF
- Azure App Service și App Service Plans
- Azure SQL Database (logical server + database) cu autentificare Entra și reguli de firewall
- Azure Storage Accounts cu ADLS Gen2 și containere blob
- Rularea T-SQL împotriva Azure SQL din Query editor în browser
- Identificarea compromisurilor de securitate și cost între un mediu de training și un workload de producție

Module recomandate în continuare

- Implementarea aceleiași stack folosind Bicep sau Terraform (Infrastructure as Code)
- Adăugarea Application Insights și Log Analytics pentru monitorizare
- Securizarea stack-ului cu Private Endpoints și VNet integration
- Automatizarea deployment-ului cu Azure DevOps Pipelines sau GitHub Actions