Azure Cloud Fundamentals

Lab 4

Security Mindset în Practică

Nu despre servicii. ***Despre mentalitate.***

RBAC · MFA · NSG hygiene · Public vs Private · Backups · Defender for Cloud · GDPR · Assume Breach

Parametru	Valoare
Modul	Security Mindset și Best Practices
Nivel	Fundamentals (aliniat AZ-900; mapează pe S11)
Timp estimat	150 – 180 minute
Cost estimat	€0,00 — toate exercițiile folosesc free tier și configurații simulate
Regiune	West Europe
Prerechizite	Lab 1–3 finalizate (rg-web-apps-001 cu Web App, SQL, Storage, monitoring)
Abordare	Exerciții hands-on + reflecție scrisă
Versiune	1.0 · Aprilie 2026

Cuprins

1. Obiectiv și mentalitate

Cele trei lab-uri anterioare v-au învățat cum să construiți, să deployați și să monitorizați o aplicație funcțională pe Azure. Acest lab este diferit. Nu este despre adăugarea de noi servicii — este despre a schimba felul în care gândești serviciile pe care deja le cunoști.

> 🧠 **REFLECȚIE**
> Dacă lab-urile anterioare au fost despre **capabilitate** (ce poți construi), acest lab este despre **responsabilitate** (cum protejezi ce ai construit și oamenii ale căror date trăiesc în acele sisteme).

La final, vei fi:

- **Experimentat gradientul** de roluri RBAC Reader → Contributor → Owner și simțit pe propria piele ce înseamnă least privilege
- **Creat și corectat** o regulă NSG clasică nesigură (port 3389 deschis la Any) și văzut-o semnalată de Defender for Cloud
- **Făcut** un container de storage public, descărcat date anonim, apoi recompus accesul corect cu SAS token
- **Verificat** MFA pe propriul cont Entra — și activat-o dacă nu era deja
- **Simulat** commit-ul unor credențiale în Git și învățat cum să le găsești și cum să le rotești
- **Auditat** postura de backup și recovery pentru SQL Database și Storage Account
- **Activat Defender for Cloud free tier** și revizuit Secure Score-ul și recomandările
- **Reflectat în scris** asupra celor cinci reguli personale de securitate pe care le vei duce în carieră

Ce face acest lab diferit

Lab-urile anterioare aveau o singură stare "**corectă**" la final — fie aplicația servea date, fie nu. Acest lab conține exerciții în care starea intenționat nesigură este parte din învățare. Vei crea pe loc configurații proaste, vei observa consecințele, apoi le vei repara. Scopul este ca data viitoare când vezi aceeași configurație proastă în producție, să știi instinctiv ce e greșit și de ce.

> 📘 **NOTĂ**
> Acest lab NU te învață să fii security engineer. Aceea e o specializare care cere ani de zile. Te învață minimumul de literație de securitate pe care orice cloud engineer, architect sau developer trebuie să o aibă din prima zi de muncă.

2. Reflecție de început: postura ta de securitate

Înainte să ne atingem de Azure, câteva întrebări despre tine. **Ia-ți cinci minute**. Scrie răspunsurile în spațiul de mai jos — nu doar în gând.

> ⚠️ **ATENȚIE**
> Este privat. Nimic din ce scrii aici nu ajunge la instructor sau altundeva. Scopul este să facem concret ceea ce deja știi abstract.

Î1 — Câte conturi online ai?

Include totul: email, social media, shopping, streaming, bancă, muncă, școală, gaming, servicii cloud, abonamente la newsletter...

_____

Î2 — La câte dintre ele folosești aceeași parolă?

Fii sincer. Nimeni nu vede.

_____

Î3 — La câte ai Multi-Factor Authentication (MFA) activat?

_____

Î4 — Când ai revizuit ultima oară ce aplicații au acces la contul tău Google / Microsoft / Apple?

_____

Î5 — Dacă principalul tău cont de email ar fi compromis astăzi, enumeră trei lucruri concrete, rele, care s-ar putea întâmpla în următoarele 24 de ore.

_____

_____

_____

> 🧠 **REFLECȚIE**
> Majoritatea oamenilor care fac acest exercițiu sincer descoperă că igiena lor **personală** de securitate este mult mai proastă decât practicile **profesionale** pe care se așteaptă să fie plătiți să le implementeze. E normal — și este exact spațiul pe care acest lab încearcă să-l închidă. Obiceiurile cu care vii la muncă sunt formate acasă, mai întâi. Corectează-ți conturile proprii săptămâna asta și judecata ta profesională se va îmbunătăți odată cu ele.

3. Pre rechizite și reguli de bază

3.1 Din lab-urile anterioare

- Resource group `rg-web-apps-001` cu toate cele cinci resurse (Web App, SQL Server, SQL DB, Storage Account, App Service Plan)
- Web app funcțional `<your-webapp>` din Lab 2
- Monitorizarea din Lab 3 (Log Analytics, App Insights, alerte) — recomandat, nu obligatoriu
- Rol de Owner sau Contributor pe rg-web-apps-001

3.2 Nou pentru acest lab

- Un al doilea browser sau fereastră incognito, ca să testezi accesul ca utilizator neautentificat
- (Opțional) Azure CLI instalat local — unele exerciții includ și echivalentul CLI
- (Opțional) Git instalat local, pentru Exercițiul 5

3.3 Reguli de bază

Spre deosebire de Lab-urile 1–3, acest lab conține exerciții în care creezi intenționat configurații nesigure. Trei reguli te țin în siguranță:

- **Folosește doar subscripția de lab.** Niciodată nu face aceste exerciții pe o subscripție de producție sau corporate. Dacă singurul tău acces Azure este printr-un tenant al companiei, oprește-te și cere instructorului un sandbox.
- **Revocă fiecare schimbare nesigură în 15 minute.** Fiecare exercițiu se încheie cu un pas de curățenie. Fă-l. Porturi nesigure, containere publice și roluri excesiv de permisive lăsate active sunt scanate de atacatori în mai puțin de o oră — nu e exagerare.
- **Nu încărca date reale.** Când lab-ul îți cere să urci un "export de clienți", folosește CSV-ul fals din exercițiu. Niciodată date personale reale, nici măcar ale tale.

> ⚠ **ATENȚIE**
>
> Exercițiile din acest lab creează temporar resurse marcate drept "risc ridicat" de orice scanner automat de securitate. E intenționat și educațional. Este și ceea ce ți-ar atrage o acțiune disciplinară într-un mediu corporate real. Nu face aceste exerciții pe un tenant de muncă.

4. Exercițiul 1 — Least Privilege cu RBAC

Azure Role-Based Access Control (RBAC) răspunde la o singură întrebare la fiecare operație pe resurse: "are această identitate voie să facă asta?". Cele trei roluri built-in pe care le vei întâlni constant sunt Reader, Contributor și Owner.

Rol	Poate face	Nu poate face
Reader	Vede toate resursele și configurațiile.	Modifică ceva. Nu poate vedea valori secrete.
Contributor	Creează, modifică, șterge resurse.	Acordă acces altor identități (nu poate asigna roluri).
Owner	Tot ce face Contributor + poate asigna roluri altora.	Nimic (în cadrul scope-ului). Control total.

Principiul: fiecărei identități i se dă exact rolul de care are nevoie pentru treaba ei. Nimic în plus. Un engineer care doar citește dashboard-uri primește Reader. O aplicație care scrie într-un singur storage account primește Storage Blob Data Contributor pe acel storage account — nu Owner pe toată subscripția.

4.1 Testul "nivelurilor de rol"

Îți vei retrograda rolul la Reader, vei încerca să creezi o resursă, vei eșua, vei face upgrade la Contributor, vei încerca din nou și vei simți diferența. Apoi curățăm.

> ⚠ **ATENȚIE**
> **Îți trebuie un al doilea utilizator sau un coleg ca ajutor.** Azure nu îți va permite să îți elimini propriul rol de Owner dacă ești singurul Owner la nivelul respectiv — te-ar bloca. Soluții: (a) cere unui coleg să îți creeze asignarea de rol, (b) folosește un al doilea cont Entra pe care îl deții, sau (c) citește exercițiul și înlocuiește numele colegului în comenzile CLI pentru a rula mai târziu cu un partener. **Conceptul** este ce contează.

Pas cu pas (portal)

1. Navighează la resource group-ul `rg-web-apps-001` → **Access control (IAM)**.
2. Dă click pe tab-ul **Role assignments**. Găsește-ți contul și notează ce rol ai (probabil Owner).
3. Click **+ Add** → **Add role assignment**.
4. Alege rolul **Reader** → Next → selectează **contul tău** ca membru → Next → Review + assign.
5. Acum ai DOUĂ roluri — Owner ȘI Reader. RBAC e aditiv, deci rolul mai puternic câștigă. Pentru exercițiu, elimină ȘI asignarea de Owner (DOAR dacă un coleg sau alt Owner poate să ți-o restaureze după). Dacă nu poți elimina Owner în siguranță, sari la pasul 9 și citește ce comportament este așteptat.
6. După eliminarea Owner: deschide o fereastră incognito nouă, loghează-te în portal, navighează la `rg-web-apps-001`.
7. Click **+ Create** în partea de sus, încearcă să creezi o resursă nouă (orice tip). Încearcă să modifici o setare existentă. Încearcă să ștergi o resursă.
8. Observă: fiecare buton de create/modify/delete ar trebui să fie dezactivat sau să producă o eroare AuthorizationFailed.
9. Pune colegul (sau al doilea cont) să-ți restaureze rolul **Contributor**. Repetă pașii 7–8.
10. Acum poți crea și șterge — dar mergi la **Access control (IAM)** → **+ Add role assignment** și observă: butonul Grant access produce o eroare de autorizare. Contributor nu poate delega.
11. Restaurează-ți rolul original **Owner**. Verifică creând cu succes o asignare de rol test și apoi eliminând-o.

Ce ai simțit

- **Ca Reader:** portalul e complet navigabil. Fiecare pagină de setări se încarcă. Dar fiecare buton de acțiune este blocat.
- **Ca Contributor:** poți reconstrui tot resource group-ul, dar nu poți delega permisiuni nimănui. Acesta e punctul dulce pentru majoritatea engineer-ilor — destul cât să-ți faci treaba, nu atât de mult încât să acorzi din greșeală acces unor terți.
- **Ca Owner:** fără bariere. Poți șterge RG-ul, adăuga utilizatori externi, acorda Owner oricui. De asta Owner trebuie să fie rar și temporar.

ÎN VIAȚA REALĂ

Într-o echipă bine organizată, pattern-ul tipic este: un cont de serviciu sau o identitate de automatizare are Contributor la nivel de RG; engineer-ii individuali au Reader la nivel de subscripție și Contributor pe scope-uri specifice de RG pe care lucrează; exact 1–2 oameni pe echipă au Owner, și doar la nivel de subscripție sau management group. **Owner permanent pe subscripție pentru o întreagă echipă este un red flag**, chiar dacă este "convenabil".

4.2 Echivalent în Azure CLI

Aceleași operații din CLI. Acestea sunt comenzile pe care le vei folosi în script-uri, pipeline-uri CI/CD și întrebări la interviuri.

```
# Listează role assignments pentru resource group
az role assignment list \
  --resource-group rg-web-apps-001 \
  --output table

# Acordă unui utilizator rolul Reader pe RG
az role assignment create \
  --assignee "coleg@tenant-ul-tau.onmicrosoft.com" \
  --role "Reader" \
  --scope "/subscriptions/<sub-id>/resourceGroups/rg-web-apps-001"

# Elimină un role assignment
az role assignment delete \
  --assignee "coleg@tenant-ul-tau.onmicrosoft.com" \
  --role "Reader" \
  --scope "/subscriptions/<sub-id>/resourceGroups/rg-web-apps-001"

# Listează toate rolurile built-in (pentru explorare)
az role definition list --output table | head -40

# Găsește subscription ID
az account show --query id --output tsv
```

> 💡 **SFAT**
> Începe să folosești Azure CLI devreme în carieră. Portalul e excelent pentru învățat, dar lent pentru munca reală. Echipele care operează Azure la scară fac aproape nimic în portal — scriu Bicep/Terraform și rulează CLI/PowerShell.

5. Exercițiul 2 — Default-ul periculos de NSG

Network Security Groups (NSG) sunt regulile de firewall ale rețelelor virtuale Azure. Un NSG prost configurat este cel mai comun mod în care engineer-ii juniori expun accidental servicii la internet. Acest exercițiu creează greșeala și îți arată ce se întâmplă apoi.

5.1 Scenariul

Vei crea un NSG standalone (fără VM atașat — economisim costul și lecția e aceeași) cu o regulă care deschide portul RDP 3389 la întregul internet. Apoi vei vedea Defender for Cloud semnalând regula, apoi o vei repara.

5.2 Creează NSG-ul nesigur

1. În Portal, caută **Network security groups** → **+ Create**.
2. Pe tab-ul Basics:
   - **Subscription:** subscripția de lab
   - **Resource group:** `rg-web-apps-001`
   - **Name:** `nsg-insecure-demo-001`
   - **Region:** West Europe
3. Click **Review + create** → **Create**.
4. După deployment, deschide NSG-ul și mergi la **Settings** → **Inbound security rules**.
5. Click **+ Add** și configurează:
   - **Source: Any**
   - **Source port ranges:** `*`
   - **Destination: Any**
   - **Service: RDP**
   - **Destination port ranges:** `3389` (auto-completat)
   - **Protocol: TCP**
   - **Action: Allow**
   - **Priority:** `100`
   - **Name:** `ALLOW-RDP-from-anywhere-DANGEROUS`
6. Click **Add**. Regula e activă acum.

> ⚠ **ATENȚIE**
>
> **Expunerea reală începe ACUM.** Chiar și fără VM atașat, acest NSG există în subscripția ta și Defender îl va evalua. Dacă ar fi atașat la un VM real cu parolă slabă, ai vedea încercări de brute-force în câteva minute. Vom repara în 15 minute.

5.3 Ce vede Defender for Cloud

1. În Portal, caută **Microsoft Defender for Cloud** și deschide-l.
2. Dacă e prima vizită, Defender are nevoie de 15–30 minute pentru evaluarea inițială. Dacă tocmai ai creat NSG-ul, întoarce-te după exercițiul următor.
3. Pe Overview, notează **Secure Score** (procent pentru toată subscripția).
4. Click pe **Recommendations** în meniul stâng.
5. Filtrează după severitate High. Caută recomandări de genul:
    - `Management ports of virtual machines should be protected with just-in-time network access control`
    - `Internet-facing virtual machines should be protected with network security groups`
    - `All network ports should be restricted on network security groups associated to your virtual machine`
6. Click într-una dintre recomandări. Defender arată resursele afectate, riscul exact și pașii de remediere. Acesta este semnalul real pe care echipa ta de securitate îl urmărește în fiecare dimineață.

5.4 Repară — metoda corectă

Există trei soluții comune pentru "am nevoie de acces RDP la un VM, dar nu vreau ca toată lumea să aibă". În ordinea crescătoare a securității:

- **Restrânge la propriul tău IP:** schimbă Source din Any în **IP Addresses** și introdu IP-ul tău public cu mască /32 (sau un interval CIDR mic pentru rețeaua ta de acasă). În portal: dă click pe **My IP** ca scurtătură.
- **Restrânge la un VNet/Service Tag:** schimbă Source la **Service Tag** și alege ceva de genul VirtualNetwork sau un public IP range specific. Folosit când caller-ul este altă resursă Azure.
- **Folosește Azure Bastion sau JIT VM Access:** răspunsul de producție — fără port RDP public. Bastion îți oferă RDP prin browser peste TLS; JIT deschide portul la cerere pentru un caller specific și pentru o fereastră scurtă.

1. În NSG-ul tău, click pe regula `ALLOW-RDP-from-anywhere-DANGEROUS`.
2. Schimbă Source la **IP Addresses** și introdu IP-ul tău public (găsește-l la https://whatismyipaddress.com — ca în Lab 1).
3. Redenumește regula `ALLOW-RDP-from-my-ip`.
4. Click **Save**.

5.5 Curățenie (șterge NSG-ul de test)

1. Înapoi la `rg-web-apps-001` → deschide `nsg-insecure-demo-001`.
2. Click **Delete** în partea de sus și confirmă.

> 🧠 **REFLECȚIE**
>
> Acest exercițiu este **greșeala** clasică de junior și cea pe care echipele de securitate cloud o găsesc în aproape fiecare audit. Diferența dintre un engineer care creează regula ("ca să meargă") și unul care o scope-uiește corect este adesea un singur checkbox — dar acel checkbox separă un estate sigur de unul despre care vei citi în știri.

6. Exercițiul 3 — Public vs Private: momentul GDPR

În Lab 1 ai creat un container public pentru imaginile aplicației web. Era corect — assets publici, nesensibili. În acest exercițiu vom crea un AL DOILEA container, vom urca "date de clienți" false, le vom accesa anonim, apoi le vom securiza corect. Lecția: același storage account poate conține date publice și private una lângă alta, iar diferența este complet în modul de configurare al containerului.

6.1 Creează un container accidental public

1. Navighează la storage account-ul tău `<your-storage>` → **Data storage** → **Containers**.
2. Click **+ Add container**:
   - **Name:** `customer-exports-demo`
   - **Anonymous access level: Container (anonymous read access for containers and blobs)** — cea mai proastă dintre cele trei opțiuni; oricine poate LIST și READ tot
3. Click **Create**.

6.2 Încarcă date false de clienți

Creează un fișier numit customers.csv (sau excel) pe calculatorul tău cu acest conținut (NU folosi date reale):

```
id,first_name,last_name,email,phone,birth_date,notes
1,Anna,Popescu,anna.p@faketest.local,+40-700-000-001,1991-03-15,VIP customer
2,Mihai,Ionescu,mihai.i@faketest.local,+40-700-000-002,1985-11-02,Past complaint
3,Elena,Georgescu,elena.g@faketest.local,+40-700-000-003,1978-06-20,Premium tier
4,Vlad,Dumitrescu,vlad.d@faketest.local,+40-700-000-004,1995-09-11,New signup
5,Maria,Stoica,maria.s@faketest.local,+40-700-000-005,1982-01-28,Churn risk
```

> ⚠ **ATENȚIE**
> Aceste date sunt în întregime sintetice și folosesc adrese @faketest.local, care nu sunt domenii RFC valide. Dacă folosești email-uri reale, numere de telefon reale sau date de naștere reale — chiar și ale tale — atunci procesezi date personale și ai obligații GDPR. Nu o face.

1. Deschide containerul `customer-exports-demo`.
2. Click **Upload**, selectează fișierul `customers.csv` și dă click Upload.
3. Click pe blob-ul încărcat. În Overview, copiază **URL**-ul afișat în Properties. Va arăta așa: `https://<your-storage>.blob.core.windows.net/customer-exports-demo/customers.csv`.

6.3 Experimentează „scurgerea” de informație

1. Deschide o fereastră NOUĂ de browser în modul incognito/privat. Acum ești un utilizator anonim de oriunde din lume.
2. Lipește URL-ul blob-ului. CSV-ul se descarcă fără prompt de login, fără autentificare. Poți citi toate datele "clienților".
3. Acum încearcă să listezi conținutul containerului. Lipește acest URL (înlocuiește numele storage account-ului):

```
https://<your-storage>.blob.core.windows.net/customer-exports-demo?restype=container&comp=list
```

4. Primești un listing XML cu fiecare blob din container. Un atacator ar itera lista, ar descărca fiecare fișier și ar pleca înainte să observi.

> 🧠 **REFLECȚIE**
>
> **Așa se întâmplă breșele reale.** Nu hackeri în glugi care tastează furios. Nu zero-day-uri. Doar un container prost configurat în care cineva a încărcat date reale, pe care Google sau un bot scanner l-a indexat, și care a ajuns într-o "bază de date scursă" pe un forum. Fiecare știre mare de data breach din ultimii cinci ani a avut *cel puțin* o variantă a acestei greșeli undeva în timeline.

6.4 Prima reparație — fă containerul Private

1. Înapoi în portal, deschide containerul `customer-exports-demo`.
2. Click **Change access level** în partea de sus.
3. Setează **Anonymous access level** la **Private (no anonymous access)**. Click OK.
4. Reîncarcă tab-ul incognito. Atât URL-ul blob-ului, cât și URL-ul de listing returnează acum AuthorizationFailure.

6.5 Varianta corectă — acces partajat cu expirare (SAS)

Dar dacă un analist legitim are nevoie de acces temporar de citire la un fișier specific? Nu faci tot containerul public. Emiți un Shared Access Signature — un URL limitat în timp și scope care acordă exact accesul pe care îl specifici.

1. Deschide blob-ul `customers.csv`.
2. Click **Generate SAS** în bara de sus.
3. Configurează:
    - **Signing method: Account key** (pentru lab; în producție preferi User Delegation Key)
    - **Permissions: Read** doar — NU Write, Delete, List
    - **Start:** acum
    - **Expiry:** 1 oră de acum (default-ul)
    - **Allowed IP addresses:** (opțional) doar IP-ul tău public
    - **Allowed protocols: HTTPS only**
4. Click **Generate SAS token and URL**. Copiază **Blob SAS URL**.
5. În tab-ul incognito, lipește URL-ul SAS. CSV-ul se descarcă — acces acordat de tine, explicit, pentru un timp limitat, fără drept de scriere.
6. După o oră (sau mâine), același URL returnează AuthorizationFailure. Acordarea a expirat singură.

> **ÎN VIAȚA REALĂ**
> **În producție:** SAS tokens sunt potrivite pentru upload-uri/descărcări de scurtă durată de la client (browser → blob). Pentru acces server-la-server, **Managed Identities** sunt mai bune — identitatea de compute însăși este autorizată via RBAC și niciun token nu părăsește Azure. Pentru partajare cu parteneri externi, uită-te la **User Delegation SAS** (semnat de o identitate Entra, nu de account key) ca să poți audita și revoca fără să rotești storage key-ul.

6.6 Curățenie

1. Șterge containerul `customer-exports-demo`. Click dreapta → **Delete**.
2. Confirmă tastând numele containerului. Asta șterge toate blob-urile din el.
3. Șterge și fișierul local `customers.csv`. Nici datele false nu au motiv să rămână pe laptop.

7. Exercițiul 4 — Verificare MFA

Multi-Factor Authentication este controlul de securitate cu cea mai mare valoare pe care îl poți adăuga oricărui cont. Cifrele Microsoft: conturile cu MFA activat au cu peste 99% mai puține șanse să fie compromise. Nu e slogan — este ce arată telemetria lor pe miliarde de sign-in-uri.

7.1 Verifică propriul tău cont Entra

1. Deschide `https://mysignins.microsoft.com` (sau loghează-te pe portal.azure.com și click pe poza de profil → View account).
2. Mergi la **Security info**.
3. Ce vezi?

- **Doar Password listat:** NU ai MFA. Starea "ușa descuiată". Repară la pasul 4.
- **Password + Phone (SMS):** ai MFA, dar cel mai slab tip. SMS-ul poate fi SIM-swapped. Preferă aplicația.
- **Password + Microsoft Authenticator (sau app similar):** bun. Acesta e nivelul de bază pentru munca profesională în cloud.
- **Password + Passkey / FIDO2 / Windows Hello:** excelent. MFA rezistent la phishing. Acolo se îndreaptă industria.

7.2 Activează sau îmbunătățește MFA

1. Dacă MFA lipsește sau e doar SMS, click **+ Add sign-in method**.
2. Alege **Microsoft Authenticator** → **Next**. (Instalează aplicația pe telefon, din App Store sau Google Play.)
3. Scanează QR-ul cu Authenticator → confirmă notificarea de test.
4. Setează Authenticator ca metoda **default** de sign-in.
5. Dacă tenant-ul tău suportă, adaugă și un **Passkey** — folosește biometria device-ului (Touch ID, Face ID, Windows Hello) și e rezistent la phishing.

> ⚠ **ATENȚIE**
>
> **Nu este un exercițiu pe care îl poți sări.** Orice profesionist cloud al cărui cont Entra nu are MFA în 2026 este o pasivă (liability) pentru orice echipă în care intră. Activează MFA acum, pe contul de lab ȘI pe contul principal Microsoft/Google/Apple — astăzi, înainte să treci mai departe.

7.3 (Bonus) Activează MFA pentru utilizatorii din tenant

Dacă deții sau administrezi tenant-ul Entra: activează Security Defaults (abordarea cea mai simplă) sau o politică de Conditional Access care cere MFA pentru toți utilizatorii. Sunt operații de un singur click, cu impact masiv.

1. Portal → caută **Microsoft Entra ID** → **Properties** → **Manage security defaults**.
2. Dacă Security defaults e Disabled și ai mai puțin de ~10 utilizatori fără licențe Conditional Access, click **Enable**.
3. Asta cere imediat înregistrare MFA la următorul sign-in pentru toți utilizatorii și blochează protocolele de autentificare legacy. Configurație de 2 minute, reducere de risc de 99%.

> **NOTĂ**
> Security Defaults este gratis și funcționează pe orice tenant Entra. Conditional Access (mai flexibil, pe bază de politici) cere licență Entra ID P1 și e ceea ce folosesc organizațiile mai mari. Pentru învățare și echipe mici, Security Defaults e răspunsul corect.

8. Exercițiul 5 — Igienă de credențiale și Git secrets

Un secret făcut commit în Git este un secret pe care trebuie să-l consideri scurs, chiar dacă ștergi commit-ul după cinci secunde. Bot-urile scanează GitHub continuu pentru AWS keys, Azure connection strings, API tokens și SSH private keys. Un secret real scurs în producție poate ajunge la un atacator în mai puțin de un minut de la push.

8.1 Greșeala simulată

Vei crea un mini-repo Git, vei face commit la un secret fake, vei observa cât de trivial este de recuperat chiar și după ce îl "ștergi", și vei învăța răspunsul corect.

1. Deschide un terminal. Alege un folder de unică folosință:

```
mkdir git-secrets-demo && cd git-secrets-demo
git init
echo "# Proiect demo" > README.md
git add README.md && git commit -m "Initial commit"
```

2. Creează un fișier cu un secret fake (NU credențiale reale):

```
cat > appsettings.json <<'EOF'
{
  "ConnectionStrings": {
    "DefaultConnection":
"Server=tcp:fakeserver.database.windows.net,1433;Database=demo;User
ID=sqldba;Password=FakeP@ssw0rd123NeverUseThis;"
  },
  "ApiKeys": {
    "ThirdParty": "sk-FAKE-1234567890abcdef1234567890abcdef"
  }
}
EOF

git add appsettings.json
git commit -m "Add config"
```

8.2 Observă cât e de ușor de găsit

1. Îți dai seama de greșeală. Șterge fișierul și fă commit din nou:

```
rm appsettings.json
git add -A
git commit -m "Remove config file"
```

2. Fișierul e "dispărut" din working tree-ul curent. Dar acum încearcă:

```
git log --all --full-history -- appsettings.json

# Afișează conținutul fișierului din commit-ul părinte al ștergerii
git show HEAD~1:appsettings.json

# Sau caută în toată istoria după secret
git log -p | grep -iE "password|apikey|connectionstring"
```

3. Fiecare dintre comenzi recuperează secretul. Este în istoria commit-urilor pentru totdeauna, decât dacă istoria este activ rescrisă.

> ⚠ **ATENȚIE**
>
> **Dacă ar fi fost un push real pe GitHub, rescrierea istoriei DUPĂ push nu ar ajuta.** Fork-urile, clone-urile și SHA-urile imutabile de commit din GitHub persistă. Bot-urile care ți-au scanat repo-ul în primele 60 de secunde au deja o copie. Singurul răspuns corect la un secret real scurs este: **rotește-l imediat** — schimbă parola, regenerează cheia, invalidează token-ul — și monitorizează folosirea.

8.3 Prevenirea greșelii în primul rând

Trei straturi de apărare, în ordinea eficacității:

- **.gitignore** fișierele care conțin de obicei secrete — `.env`, `appsettings.Development.json`, `*.pem`, `secrets.json`. Repo-ul trebuie să aibă un `.gitignore` bun înainte de primul commit.
- **Pre-commit hooks** care scanează după secrete. Unelte: `gitleaks`, `git-secrets` sau `detect-secrets`. Rulează înainte de fiecare commit; blochează commit-ul dacă un pattern de secret se potrivește.
- **Niciodată nu pune secrete în cod.** Folosește Azure Key Vault (cu Managed Identity pe App Service / Functions / VMs) sau variabile de mediu injectate la deploy-time. Acesta e singurul răspuns arhitectural.

8.4 Verifică proiectele tale reale

1. Ia orice proiect pe care lucrezi acum și care e în Git. Rulează:

```
git log -p | grep -iE "password|secret|apikey|connectionstring|BEGIN RSA PRIVATE"
```

2. Dacă găsești ceva — chiar și ceva care "pare placeholder" — tratează-l ca pe o scurgere. Rotește orice credențial mapează pe el.
3. Pentru repo-urile de pe GitHub, activează **GitHub secret scanning** (gratuit pentru public repos, inclus în Advanced Security pentru private). Scanează continuu și alertează la pattern-uri cunoscute de secrete.

8.5 Curățenie

1. Șterge complet folder-ul `git-secrets-demo` — a fost doar pentru învățare.

> 🌍 **ÎN VIAȚA REALĂ**
> Combinația Azure Key Vault + Managed Identity este **singurul** pattern de secret management care scalează în producție reală. App Service citește Key Vault la startup folosind propria managed identity, nicio parolă nu atinge code base-ul, rotația e o acțiune unică în Key Vault fără redeployment. Dacă reții un singur lucru din acest exercițiu: **următorul tău proiect Azure trebuie să folosească Key Vault + Managed Identity din ziua 1**, nu să fie adăugate ulterior.

9. Exercițiul 6 — Verificarea backup-urilor

Backup-ul este unul din acele subiecte la care toată lumea dă din cap aprobator, apoi nimeni nu verifică. Exercițiul ăsta e trei minute de navigare și un minut de gândire, și e suficient cât să știi postura reală de recovery pentru fiecare dintre resursele din Lab 1.

9.1 Azure SQL Database — ce backup am?

1. Navighează la SQL Database-ul tău `dbwebapp` → **Data management** → **Backups**.
2. Vezi trei tab-uri: Available backups, Retention policies, Long-term retention.
3. Pe **Retention policies**:
   - **Point-in-time restore (PITR):** default 7 zile. Fiecare secundă din ultimele 7 zile este restorabilă. Aceasta e plasa principală de siguranță.
   - **Long-term retention (LTR):** dezactivat by default. Activează pentru conformitate (snapshot-uri anuale păstrate până la 10 ani).
4. Notează **Earliest restore point** pe tab-ul Overview al bazei de date. Acela e cel mai vechi moment la care ai putea face rollback acum.

🧠 **REFLECȚIE**

Număr critic de știut pe de rost:
RPO (Recovery Point Objective) = pierderea acceptabilă de date într-un dezastru.
RTO (Recovery Time Objective) = downtime-ul acceptabil. Pentru un lab: RPO=1h, RTO=1h sunt fine. Pentru un sistem e-commerce către clienți: RPO poate fi 1 minut, RTO 5 minute. Dacă cifrele par abstracte, întreabă-te: *dacă am pierde ultimele X ore de date chiar acum, am fi concediați / dați în judecată / în faliment?* Asta îți spune RPO-ul real.

9.2 Storage Account — soft delete și versioning

1. Navighează la storage account-ul tău `<your-storage>` → **Data protection**.
2. Verifică starea:
   - **Soft delete for blobs:** activat în Lab 1 cu retenție de 7 zile. Blob-urile șterse sunt recuperabile 7 zile.
   - **Soft delete for containers:** același, 7 zile.
   - **Versioning:** DEZACTIVAT în Lab 1. Acceptabil pentru imagini, dar NU pentru fișiere de date. Activează dacă stochezi ceva important.
   - **Point-in-time restore for containers:** avansat — cere versioning, change feed și soft delete. Util pentru recovery din ransomware.

9.3 Testează un restore (nu sări peste)

Un backup pe care nu l-ai restaurat niciodată nu e un backup, e o speranță. În medii profesioniste, se face restore într-un mediu de test pe un calendar — cel puțin trimestrial — ca să cunoști procedura și cifrele să fie reale.

1. În Overview-ul SQL Database, click **Restore** sus.
2. În panoul de restore:
   - **Source: Point-in-time restore**
   - **New database name:** `dbwebapp-restoretest`
   - **Point-in-time:** orice moment de mai devreme azi (default: cel mai recent)
3. Click **Review + create** → **Create**. Restore-ul va lua 2–10 minute.
4. După ce DB-ul restaurat e disponibil, rulează `SELECT COUNT(*) FROM dbo.Students` pe el prin Query Editor. Confirmă că returnează date ca la acel moment.
5. Curățenie: șterge `dbwebapp-restoretest` după. Costă la fel ca o bază de date normală și nu ai nevoie de două copii.

💡 **SFAT**

Baza de date restaurată este o bază de date complet separată — nu o înlocuire a originalului. Așa funcționează modelul de restore din Azure: restaurezi ca DB nou, validezi și apoi faci cutover-ul (rename sau repoint al aplicației). Previne ca un "restore" să-ți distrugă datele curente.

10. Exercițiul 7 — Defender for Cloud

Microsoft Defender for Cloud este serviciul de cloud security posture management (CSPM) și threat protection built-in al Azure. Free tier-ul îți oferă un Secure Score, recomandări continue contra Microsoft Cloud Security Benchmark, și o vedere de regulatory compliance. Este cea mai utilă unealtă pentru un engineer junior care învață securitate — îți spune, în limbaj simplu, ce să repari următor.

10.1 Navighează Defender și găsește Secure Score

1. Portal → caută **Microsoft Defender for Cloud**.
2. Pe Overview, notează cele patru tile-uri:
    - **Secure Score:** un singur procent 0–100% pentru subscripția ta. Ponderat după impactul controalelor.
    - **Regulatory compliance:** postura ta împotriva standardelor ca ISO 27001, NIST, CIS etc.
    - **Workload protections:** care dintre planurile Defender plătite sunt pornite (probabil niciunul — folosim free tier).
    - **Inventory:** toate resursele monitorizate și scorurile individuale.

10.2 Revizuiește recomandările

1. Click **Recommendations** în meniul stâng.
2. Filtrează după:
    - **Severity: High** — acestea sunt cele de reparat prima dată
    - **Resource type: Storage accounts** — apoi scroll prin ce se aplică pentru lab-ul tău
3. Alege o recomandare High și click pe ea. Citește:
    - **Description:** ce e riscul, în limbaj clar
    - **Remediation steps:** cum exact se repară, de obicei cu screenshots de portal sau comandă CLI
    - **Affected resources:** care dintre resursele tale au nevoie de reparare

10.3 Implementează trei reparări

Alege trei recomandări din listă și aplică-le chiar. Sugestii care probabil apar pe subscripția de lab:

- **"Secure transfer to storage accounts should be enabled."** Deja activat în Lab 1 — deci nu ar trebui să apară ca violare.
- **"Storage account public access should be disallowed."** Intenționat permitem acces public pe containerul `images`. Acesta e pentru producție, nu lab.
- **"Auditing on SQL server should be enabled."** Du-te la `<your-sqlserver>` → Auditing → pornește, target Log Analytics workspace. Două click-uri.
- **"Azure Defender for SQL servers should be enabled."** Asta e plătită — sari peste pentru lab, dar notează că există.
- **"App Service authentication should be enabled."** Nu se aplică aplicației demo, dar merită știut.
- **"TLS should be updated to the latest version for the web app."** Du-te la `<your-webapp>` → Configuration → General settings → setează Minimum TLS Version la 1.2 (sau 1.3 dacă e disponibil).

10.4 Vederea de compliance

1. Click **Regulatory compliance** în meniul stâng.
2. Vezi standarde ca **Microsoft cloud security benchmark** (default), **ISO 27001**, **PCI DSS** etc. — cu numere pass/fail pe fiecare control.
3. Expandează un grup de controale (ex. Network Security). Fiecare control mapează la recomandări specifice din Defender. Exact așa revizuiesc auditorii mediul tău.

> **ÎN VIAȚA REALĂ**
> Echipele reale revizuiesc Secure Score săptămânal ca metric, cu recomandări specifice asignate inginerilor ca tichete. O echipă care trece de la 55% la 85% Secure Score într-un trimestru este un rezultat măsurabil, pe care îl poți apăra în fața conducerii. E și una dintre cele mai ușoare moduri prin care un engineer junior să aibă impact vizibil în primele luni la job — ia trei recomandări pe săptămână și rezolvă-le, și managerul tău va observa.

11. Cele cinci scenarii — judecata ta profesională

Această secțiune ia exercițiul de discuție din S11 și îl transformă în evaluare scrisă. Pentru fiecare scenariu, răspunde la toate cele trei coloane, cu cuvintele tale. Așa arată un incident write-up bun, și e aceeași structură pe care o vei folosi pentru pull requests și design docs în cariera ta.

#	Scenariu	Ce e greșit?	Ce ar trebui făcut?
1	Un VM de test cu port RDP 3389 deschis la 'Any' Source, lăsat pornit peste weekend.		
2	Un cont Azure nou fără MFA — "oricum nu are date sensibile".		
3	Un VM de development pornit 24/7 pentru că "durează prea mult să pornească în fiecare dimineață".		
4	Un container Blob marcat Public pentru imagini — cineva urcă apoi un export CSV cu email-uri de clienți în el.		
5	Un coleg are nevoie urgent de acces, i se acordă Owner temporar — "schimbăm după".		

> 💡 **SFAT**
> Dacă faci lab-ul în clasă, 10 minute de scris singur, apoi discuție cu un partener. Vei vedea că răspunsurile "corecte" converg, dar raționamentele variază — raționamentul e locul unde se face învățarea reală.

11.1 Răspunsuri de referință

Nu te uita la acestea decât după ce ai scris tu singur.

Scenariul 1 — RDP deschis

- **Ce e greșit:** Portul e expus la tot internetul. Încercări de brute-force încep în minute. VM-ul se și facturează continuu peste weekend.
- **Ce ar trebui făcut:** Source = My IP doar (sau Azure Bastion). VM deallocated când nu e folosit. Dacă e un pattern real de muncă, un jump-box partajat cu Bastion e arhitectura corectă.

Scenariul 2 — Fără MFA

- **Ce e greșit:** Orice cont Azure poate crea resurse costisitoare sau — mai rău — asuma permisiuni în alte scope-uri. "Nu are date sensibile" azi nu e la fel cu "luna viitoare". Un cont compromis este un beachhead pentru atacator.
- **Ce ar trebui făcut:** MFA activat imediat, înainte de orice altceva. Security Defaults pe tenant dacă e permis. MFA nu e "nice-to-have" pentru conturi "importante".

Scenariul 3 — VM de dev non-stop

- **Ce e greșit:** 500 ore/lună de cost compute fără valoare. De asemenea: suprafață de atac mai mare, mai puțin patched, mai mult de monitorizat.
- **Ce ar trebui făcut:** Auto-shutdown configurat (Portal: VM → Operations → Auto-shutdown). Developer-ii intră în obicei cu `az vm deallocate` la sfârșitul zilei. Sau: mută dev pe un serviciu PaaS care nu taxează când e idle.

Scenariul 4 — Container public, upload sensibil

- **Ce e greșit:** Un container cu access level public contaminat cu date personale. Breșă GDPR. Notificare, investigație, amenzi potențiale până la 4% din cifra de afaceri anuală sau 20M EUR.
- **Ce ar trebui făcut:** Containere pentru assets publice (imagini, fișiere statice) sunt separate de containere pentru orice altceva. Accesul public e configurat per-container și dublu-verificat în PR review. Orice conține date de clienți e servit via SAS cu expirare și restricție IP, sau via un API autentificat care injectează identitatea.

Scenariul 5 — Owner temporar

- **Ce e greșit:** "Temporar" e cel mai permanent cuvânt în access management. Fără o amintire în calendar, un tichet sau Privileged Identity Management (PIM), accesul persistă indefinit. Colegul poate acorda acum Owner altora.
- **Ce ar trebui făcut:** Chiar și sub presiunea timpului, 5 minute să faci scope corect de rol. Folosește PIM (cere Entra ID P2) pentru acces privilegiat limitat în timp, cu workflow de aprobare. Dacă PIM nu e disponibil, setează amintire în calendar la 24 ore să elimini rolul — și chiar elimină-l.

12. Curățenie

Verifică că fiecare schimbare nesigură din acest lab a fost revocată. Nu e opțional — a lăsa artefactele de test active înseamnă a te lăsa expus.

Checklist

- ☐ NSG-ul `nsg-insecure-demo-001` e **șters** (Exercițiul 2)
- ☐ Containerul `customer-exports-demo` e **șters** (Exercițiul 3)
- ☐ Fișierul local `customers.csv` e **șters** de pe mașina ta (Exercițiul 3)
- ☐ Orice role assignment temporar din Exercițiul 1 e **eliminat** și rolul original e restaurat (Exercițiul 1)
- ☐ Folder-ul `git-secrets-demo` e **șters** (Exercițiul 5)
- ☐ Baza de date `dbwebapp-restoretest` e **ștearsă** (Exercițiul 6)
- ☐ MFA pe contul de lab rămâne **activat** — e singurul lucru pe care **nu** îl revoci (Exercițiul 4)
- ☐ Modificările de configurație din Defender for Cloud rămân (Exercițiul 7) — fac lab-ul mai sigur, nu mai puțin

> ⚠ **ATENȚIE**
> Resurse nesigure rămase sunt un security smell chiar și pe o subscripție de lab. Treci prin checklist înainte să închizi documentul.

Pentru curățenia completă a seriei (Lab 1–4), șterge întregul resource group:

```
# Portal: Resource groups → rg-web-apps-001 → Delete resource group
# Sau via CLI:
az group delete --name rg-web-apps-001 --yes --no-wait
```

13. Temă scrisă (Reflecție finală)

Partea hands-on s-a încheiat. Această secțiune este muncă scrisă. Este cea mai importantă parte din lab. Abilitățile tehnice se pot învăța din documentație; mentalitatea de securitate se formează prin reflecție.

> 🧠 **REFLECȚIE**
> Fă aceste secțiuni cu cuvintele tale. Nu căuta răspunsuri. Calitatea gândirii contează, nu polish-ul scrisului. Instructorul va citi ce ai scris, dar la fel și **tu** — peste șase luni, când vei înfrunta o decizie reală la muncă și îți vei aminti ce ai scris astăzi.

13.1 — Cele cinci reguli personale de securitate

Scrie cinci reguli pe care te angajezi să le respecți în cariera ta în cloud. Nu reguli dintr-un slide. Reguli de la tine, pentru tine, cu cuvintele tale, pe care chiar intenționezi să le ții. Un rând pentru regulă, plus un rând pentru justificare.

Regula 1:

_____

Pentru că:

_____

Regula 2:

_____

Pentru că:

_____

Regula 3:

_____

Pentru că:

_____

Regula 4:

_____

Pentru că:

_____

Regula 5:

_____

Pentru că:

13.2 — Least Privilege în viața reală

Explică Least Privilege folosind un exemplu din afara IT. Folosește ceva din viața ta de zi cu zi care deja urmează (sau ar trebui să urmeze) acest principiu. Exemple de la care poți pleca: cine are cheile casei tale, cine are acces la contul tău bancar, ce aplicații au permisiune de locație pe telefon, cine poate vedea un document partajat în familie. Trei până la cinci propoziții.

13.3 — MFA vs. Credential Stuffing

Explică, ca și cum ai explica unui membru de familie care nu lucrează în IT: ce e credential stuffing și cum îl oprește MFA? Poți folosi analogia cheie-alarmă dacă vrei. Ține-o suficient de scurt cât să poți explica la masa de cină.

13.4 — Bonus: auto-evaluare Defender for Cloud

Scrie un paragraf scurt (4–6 propoziții) răspunzând la: (a) care e Secure Score-ul tău curent pe rg-web-apps-001, (b) care sunt top 3 recomandări High văzute, (c) pe care ai repara-o prima și de ce, (d) cum ai integra o revizuire săptămânală Defender în munca ta dacă te-ai alătura unei echipe care nu face asta.

Anexa A — Azure CLI Security Cheat Sheet

Referință pentru comenzile folosite în lab, plus câțiva vecini care merită știuți.

A.1 RBAC

```
# Cine are acces la ce într-un RG?
az role assignment list --resource-group rg-web-apps-001 --output table

# Listează toate rolurile built-in
az role definition list --query "[?roleType=='BuiltInRole'].roleName" -o tsv | sort

# Acordă un rol
az role assignment create \
  --assignee <user@tenant.onmicrosoft.com> \
  --role "Reader" \
  --scope "/subscriptions/<sub-id>/resourceGroups/rg-web-apps-001"

# Elimină un rol
az role assignment delete \
  --assignee <user@tenant.onmicrosoft.com> \
  --role "Reader" \
  --scope "/subscriptions/<sub-id>/resourceGroups/rg-web-apps-001"
```

A.2 Reguli NSG

```
# Listează NSG-urile din RG
az network nsg list --resource-group rg-web-apps-001 -o table

# Afișează regulile unui NSG
az network nsg rule list \
  --resource-group rg-web-apps-001 \
  --nsg-name nsg-demo-001 \
  --output table

# Adaugă o regulă RDP cu scope (doar IP-ul tău)
az network nsg rule create \
  --resource-group rg-web-apps-001 \
  --nsg-name nsg-demo-001 \
  --name ALLOW-RDP-from-my-ip \
  --priority 100 \
  --source-address-prefixes <your-public-ip>/32 \
  --destination-port-ranges 3389 \
  --protocol Tcp \
  --access Allow
```

A.3 Acces la storage

```
# Setează un container ca privat
az storage container set-permission \
  --account-name <your-storage> \
  --name customer-exports-demo \
  --public-access off

# Generează un SAS read-only pentru un blob, valid 1 oră
az storage blob generate-sas \
  --account-name <your-storage> \
  --container-name customer-exports-demo \
  --name customers.csv \
  --permissions r \
  --expiry $(date -u -d "1 hour" '+%Y-%m-%dT%H:%MZ') \
  --https-only \
  --output tsv
```

A.4 Defender for Cloud

```
# Afișează Secure Score curent
az security secure-scores list --output table

# Listează recomandările de securitate
az security assessment list --output table

# Afișează detaliile unei recomandări specifice
az security assessment show --name <assessment-id>
```

A.5 MFA / Entra

Majoritatea operațiilor MFA se fac prin portalul Entra sau via Microsoft Graph, nu direct din az CLI. Câteva comenzi utile:

```
# Afișează utilizatorul logat
az ad signed-in-user show

# Listează utilizatorii din tenant (necesită permisiuni)
az ad user list --output table

# Pentru Security Defaults / Conditional Access, folosește Microsoft Graph SDK
# (dincolo de scope-ul acestui lab).
```

Anexa B — Greșelile clasice ale juniorilor

Listă de referință. Print-eaz-o și pin-uiește-o lângă monitor pentru primele șase luni.

Greșeală	De ce se întâmplă	Ce să faci în schimb
Deschide tot "ca să meargă"	Sub presiunea timpului, calea cea mai rapidă spre un rezultat verde	Înțelege de ce e blocat ÎNAINTE să deschizi. 5 min de citit salvează săptămâni de răspuns la incident.
Copiază configurații din tutoriale orbește	Tutorialele au adesea default-uri nesigure pentru simplitate	Citește fiecare linie. Dacă nu înțelegi o linie, nu o rula.
Commit la credențiale în Git	Grabă + fără .gitignore + nefamiliar cu Key Vault	Template .gitignore mai întâi; Key Vault al doilea; pre-commit hook al treilea
Ignoră alertele	Zgomot, alert fatigue sau senzația de copleșeală	Fiecare alertă e triajată. Dacă e zgomot, schimbă regula — nu o muta pe mute.
Lasă resursele de test pornite	"Mă întorc după" — dar nu te întorci	Fiecare resursă de test are dată de expirare. Tag cu owner și delete-by. Auto-shutdown pe VM-uri.
Nu documentează	"Evident" azi e misterios peste 6 luni	Scrie README înainte de cod. Dacă nu e scris, nu a existat.
Nu întreabă	Frica de a părea prost	A întreba e profesionalism, nu slăbiciune. "Prost" e să execuți ceva ce nu înțelegi.
Sare peste backup	"E doar dev" / "putem reconstrui"	Fiecare serviciu de date are o postură de backup pe care o poți articula. "Nu știu" nu e răspuns acceptabil.
Acordă Owner pentru comoditate	Mai rapid decât scope-ul corect al rolului	Contributor acoperă 95% din cazuri. Owner e rar și temporar. PIM pentru limitat în timp.
Aceeași parolă peste tot	Încărcătură cognitivă	Password manager + unică pe serviciu. O parolă master, o aplicație.
MFA dezactivat "doar pentru testing"	Evitarea frecvenței	MFA nu e negociabil. Dacă fluxul tău de dev face MFA dureros, repară dev flow-ul — nu MFA.

Anexa C — Lecturi suplimentare

O listă scurtă și curatoriată. Preferă adâncimea în locul lățimii — parcurge 3 dintre acestea temeinic decât toate 10 superficial.

Resurse oficiale Microsoft

- **Microsoft Cloud Security Benchmark** — standardul normativ de securitate după care Defender te verifică. Scurt, practic, revizuit trimestrial.
- **Microsoft Security Best Practices** — ghidul "security baseline". Un checklist realist.
- **Cloud Adoption Framework - Secure methodology** — cum să gândești securitatea din perspectivă de governance și arhitectură, nu doar controale tehnice.

Deep-dive-uri specifice Azure

- **Azure RBAC documentation** — citește definițiile rolurilor built-in. A ști că "Storage Blob Data Reader" există previne greșeala "acordă Storage Account Contributor".
- **Azure Key Vault + Managed Identity tutorial** — cel mai util pattern de securitate din Azure. Dacă faci un lucru real după acest lab, fă-l pe acesta.
- **Azure Private Endpoint documentation** — pasul următor după regulile de firewall. Răspunsul de producție la "cum restricționez accesul la SQL / storage / key vault?"

Dincolo de Azure

- **OWASP Top 10** — riscuri la nivel de aplicație. Securitatea cloud nu se oprește la infrastructură.
- **NIST Cybersecurity Framework** — limbajul universal al posturii organizaționale de securitate. Identify, Protect, Detect, Respond, Recover.
- **Have I Been Pwned** — verifică dacă email-ul tău apare în breșe cunoscute. De obicei lămurește.

Dacă vrei un traseu de certificări

- **AZ-900 (Azure Fundamentals)** — acoperă mult din aceste concepte la nivel de awareness. Următorul tău examen.
- **SC-900 (Security, Compliance, Identity Fundamentals)** — complement la AZ-900, focusat pe portofoliul de securitate Microsoft.
- **AZ-500 (Azure Security Engineer Associate)** — trasul specialist de securitate. Adâncime reală.

Ce au construit zece sesiuni

Ai trecut, de-a lungul cursului, de la a nu ști nimic despre cloud la:

- **Construit** — rețele, compute, storage, baze de date, aplicații
- **Deployat** — cod, configurație, connection strings
- **Observat** — metrics, logs, traces, alerte, dashboards
- **Gândit responsabil** — cine are acces, ce e expus, ce se întâmplă dacă ceva merge prost

Tehnica se învață din documentație. Mentalitatea se formează doar prin experiență, reflecție și conversații ca cele din acest curs și din S11.

> 🧠 **REFLECȚIE**
> Nu mai ești începător în Azure.
> Ești **practicant responsabil**. Diferența nu e în ce știi — **e în cum decizi.**

Ne vedem în Sesiunea 12.