



# Azure Fundamentals — Sesiunea 7

## IaaS vs PaaS & Cloud Native Mindset

De la mașini virtuale la platforme gestionate — astăzi învățăm cum să conducem mașina fără să deschidem capota.

SESIUNEA 7

AZURE APP SERVICE

WEB APPS



## Bun revenit! 🖐️

Săptămâna trecută am creat două mașini virtuale. Ați instalat Windows. Ați intrat prin RDP. Ați deschis un terminal Linux prin SSH. Practic, **ați închiriat două calculatoare în cloud și le-ați controlat de la distanță.**

Ați simțit puterea. Ați simțit și responsabilitatea.

🗒️ 💡 **Întrebarea zilei:** Dacă vreau doar să public un site simplu, am nevoie de un calculator complet? Am nevoie să instalez un sistem de operare, să configurez un firewall, să aplic actualizări de securitate în fiecare lună?

Răspunsul este **nu**. Și sesiunea de astăzi vă arată de ce.

# Două moduri de a gândi infrastructura

Înainte de a deschide Azure Portal, trebuie să înțelegem o diferență fundamentală de filozofie. Există două mari modele de servicii cloud pentru a rula aplicații — și alegerea dintre ele definește cât de mult timp petreceți gestionând infrastructura versus construind aplicația.

## IaaS — Infrastructure as a Service

Tu controlezi totul de la sistemul de operare în sus. Libertate maximă, responsabilitate maximă. Exemple: Azure Virtual Machines.

## PaaS — Platform as a Service

Nu mai administrezi infrastructura. Cineva a pregătit-o deja pentru tine. Tu te concentrezi pe aplicație. Exemple: Azure App Service, Azure Web Apps.

Niciuna nu este mai bună în mod absolut. **Alegerea depinde de cerință, de echipă și de contextul tehnic.** Sesiunea de astăzi vă ajută să înțelegeți când și de ce alegeți una sau alta.

# Analogia #1 — Casa vs. Apartamentul Mobilat



## 🏠 Casa goală — IaaS

Primești pereții și instalația electrică. Atât. Tu cumperi mobila, tu o montezi, tu chemi meșterul când se strică ceva, tu plătești toate reparațiile. **Control total. Responsabilitate totală.**

- Instalezi sistemul de operare
- Configurezi firewall-ul
- Aplici patch-uri de securitate lunar
- Monitorizezi disc și memorie



## 🛋️ Apartamentul mobilat — PaaS

Totul este deja pregătit. Canapea, pat, frigider, mașină de spălat. Tu vii cu bagajul și te muți imediat. **Simplitate maximă. Responsabilitate minimă.**

- Nu te ocupi de infrastructură
- Azure gestionează OS-ul
- Tu încarci codul și rulează
- Focus pe aplicație, nu pe server

# Analogia #2 — Mașina Personală vs. Ride Sharing

## Mașina personală — VM



Tu o întreții, tu o duci la service, tu plătești asigurarea, tu o reperi când se strică. Ai control total. Dar ai și costuri și responsabilități constante — în fiecare zi, indiferent dacă o folosești sau nu.

O mașină virtuală este mașina ta personală din cloud. Rulează non-stop, consumi resurse și plătești chiar dacă nu o folosești intens.

## Ride Sharing — Web App

Spui unde vrei să ajungi. Atât. Cineva altcineva se ocupă de mașină, de combustibil, de asigurare, de revizie. Tu plătești doar pentru călătoria efectuată.

Un Web App este ride sharing-ul din cloud. Plătești pentru ce folosești, Azure se ocupă de tot ce este „sub capotă” — sistemul de operare, web server-ul, patch-urile de securitate, load balancing-ul.

  **Concluzie:** Ambele te duc la destinație. Dar una necesită mult mai mult efort de întreținere.

# Analogia #3 — Generatorul Electric

În trecut, companiile aveau **generatoare electrice proprii în curte**. Se ocupau de combustibil, de mentenanță, de reparații. Era scump, complicat și distrazea atenția de la activitatea principală a companiei.

Astăzi, nimeni nu mai face asta. Pur și simplu folosim curentul din rețea. Plătim ce consumăm și nu ne gândim la ce este în spatele prizei.

## Cloud Native Mindset

Folosești infrastructura ca un **serviciu de utilitate**, nu ca ceva pe care îl construiești de la zero. Întrebarea permanentă este: *„Chiar am nevoie să administrez un server? Sau pot folosi un serviciu deja construit?”*

## Valoarea reală

Companiile moderne aleg simplitatea nu pentru că nu știu să administreze servere, ci pentru că **vor să se concentreze pe aplicație, nu pe infrastructură**. Timp salvat = valoare creată.

# De ce nu folosim mereu VM?

Mașinile virtuale sunt extraordinar de flexibile și puternice — dar această flexibilitate vine cu un cost ascuns: **timp și efort de administrare.**



## Configurare complexă

Instalezi sistemul de operare, configurezi web server-ul (IIS, Nginx, Apache), setezi regulile de firewall, creezi certificatele SSL. Ore de muncă înainte să ruleze prima linie de cod.



## Monitorizare permanentă

Discul s-a umplut? Memoria este la capacitate maximă? Procesul web server-ului a căzut? Cineva trebuie să verifice constant starea serverului — 24/7.



## Patch-uri constante

Windows Update, actualizări de securitate Linux, vulnerabilități noi descoperite săptămânal. Fiecare trebuie aplicată, testată și monitorizată. Este un job în sine.



## Cost chiar și în repaus

Un VM rulează și consumă resurse chiar dacă nu ai trafic. Plătești pentru capacitate rezervată, nu pentru utilizare reală. Pentru proiecte mici, este inefficient.

# Analogia Restaurantului

Dacă vrei să gătești și să servești mâncare, ai două opțiuni fundamentale:

1

## Opțiunea 1: Clădirea goală

Închiriezi o clădire goală, cumperi aragazul, angajezi bucătarul, faci toată mentenanța singur. Tu ești responsabil de tot — de la instalația electrică la curățenia zilnică.

**Echivalent: Virtual Machine**

2

## Opțiunea 2: Bucătăria echipată

Închiriezi o bucătărie complet echipată, cu aragaz, ustensile, hotă, mese de lucru. Te concentrezi exclusiv pe gătit — pe ceea ce știi să faci cel mai bine.

**Echivalent: Azure Web App**

3

## Diferența esențială

Cu PaaS, **nu mai vedeți sistemul de operare**. Nu există din perspectiva voastră. Azure îl gestionează complet. Voi încărcați codul și aplicația rulează.

**Focus 100% pe valoare**

# Creăm primul Web App

Suficientă teorie — hai să punem în practică. Urmăți pașii de mai jos pentru a crea primul vostru Azure Web App.

1

## Resource Group

Creați un Resource Group nou. Numiți-l `rg-s07-[prenumele vostru]`. Selectați regiunea **West Europe**.

2

## Create Web App

În Azure Portal → **Create a Resource** → căutați **Web App**. Completați câmpurile conform instrucțiunilor de mai jos.

3

## Configurare

Name: `s07-webapp-[prenumele vostru]`  
Publish: **Code** | Runtime: **.NET / Node / PHP** Region: **West Europe**

4

## App Service Plan

Selectați tier-ul **Basic sau Free**. Acesta este motorul aplicației — definește CPU, memorie și numărul de instanțe disponibile.

5

## Deploy & Verificare

Dați **Create** și așteptați. Accesați URL-ul generat: `s07-webapp-[prenume].azurewebsites.net`

# App Service Plan — Conceptul Cheie

## Ce este App Service Plan?



App Service Plan este **motorul care rulează aplicația voastră**. Nu este aplicația în sine — este infrastructura de calcul pe care rulează aplicația.

Definește:

- Câtă putere de procesare (vCPU)
- Câtă memorie RAM
- Câte instanțe simultane
- Regiunea geografică
- Funcționalități disponibile (SSL custom, scaling, etc.)

## Tier-urile disponibile

Tier	Utilizare	Cost
Free (F1)	Testare, learning	Gratuit
Shared (D1)	Dev/Test	Foarte mic
Basic (B1-B3)	Producție mică	Redus
Standard (S1-S3)	Producție	Mediu
Premium (P1-P3)	High-performance	Ridicat

  Pe un singur App Service Plan poți rula **mai multe Web Apps**, împărțind resursele între ele — eficient și economic.

# ✓ Web App-ul este live!

Când deployul este gata, accesați URL-ul generat automat de Azure:

📄 [https://s07-webapp-\[prenumele vostru\].azurewebsites.net](https://s07-webapp-[prenumele vostru].azurewebsites.net)

## 🔒 HTTPS automat

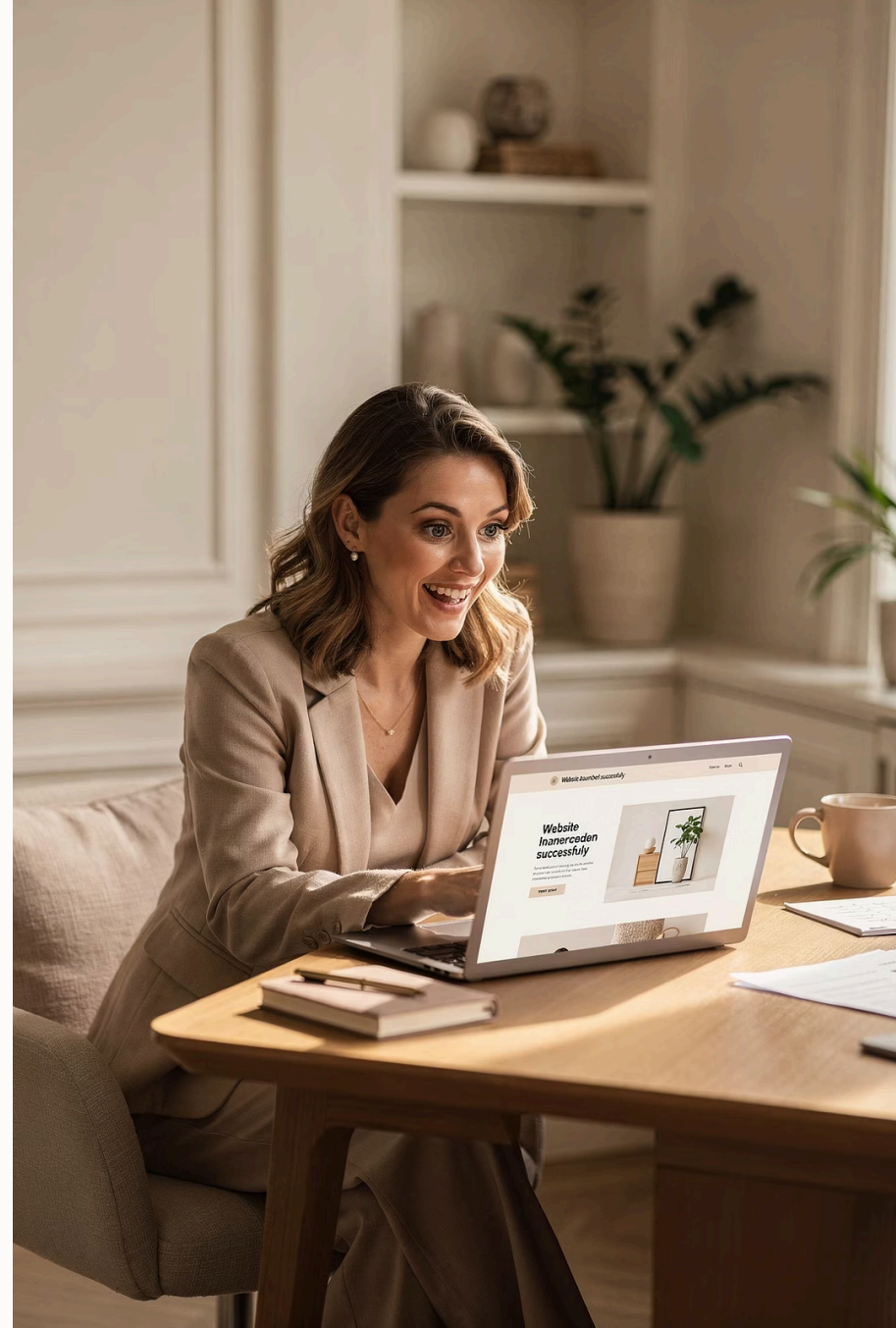
Certificatul SSL este generat și gestionat automat de Azure. Zero configurare din partea voastră.

## 🌐 Domeniu public

URL-ul este accesibil din orice colț al lumii, imediat după deploy. Fără DNS manual, fără IP public configurat de voi.

## ⚡ Zero VM

Nu ați creat nicio mașină virtuală. Nu ați instalat nimic. Nu ați configurat niciun firewall. Azure a gestionat toată infrastructura.



# Publicăm o aplicație reală 🎉

Acum facem ceva care va părea aproape magic. Vom publica un fișier HTML simplu pe Web App-ul nostru și îl vom vedea **live pe internet în câteva minute**.

## Pregătiți fișierul index.html

Creați un fișier simplu cu conținut personalizat:

```
<html>
<head>
  <title>Azure Web App</title>
</head>
<body
  style="background:#3A3A3A;
  color:white;
  font-family:sans-serif;
  text-align:center;
  padding:50px">
  <h1>Hello from Azure Cloud!</h1>
  <p>Publicat de: [Prenumele vostru]</p>
  <p>Sesiunea 7 — Azure Fundamentals</p>
</body>
</html>
```

## Pașii de upload

01

---

### Advanced Tools (Kudu)

În Azure Portal, mergeți la Web App → **Advanced Tools** → Go

02

---

### Debug Console

Deschideți **Debug Console** → **CMD** și navigați la folderul `site/wwwroot`

03

---

### Upload fișier

Trageți fișierul `index.html` în browser (drag & drop) pentru a-l încărca în `wwwroot`

04

---

### Verificare

Accesați URL-ul Web App-ului. Site-ul vostru este **live pe internet!**



# Ce NU mai administrăm?

Hai să facem un inventar rapid al lucrurilor pe care **nu le-ați făcut** astăzi — dar care ar fi existat pe agenda voastră dacă folosiți o mașină virtuală.


**Adresă IP publică**  
Nu ați configurat nicio adresă IP publică. Azure a generat-o și a gestionat-o automat. Voi nu ați văzut-o și nu ați atins-o.

**Sistem de operare**  
Nu ați instalat Windows sau Linux. Nu ați ales versiunea. Nu ați aplicat niciun update. Azure gestionează OS-ul complet, în fundal, transparent pentru voi.

**Patch-uri de securitate**  
Nu ați aplicat niciun patch. Azure se ocupă de patching automat — zero vulnerabilități descoperite și netratate din cauza întârzierii voastre.

**Web Server (IIS / Nginx / Apache)**  
Nu ați instalat și configurat niciun web server. Azure a configurat totul în spatele scenei, adaptat la runtime-ul ales de voi.

**Monitorizare disc și memorie**  
Nu ați configurat alerte, nu ați verificat spațiul liber pe disc. Azure are monitoring integrat și gestionează resursele infrastructurii.

 **🎯 Esența PaaS:** Nu mai suntem administratori de sistem. Suntem dezvoltatori și arhitecți care se concentrează pe **valoarea aplicației**, nu pe mentenanța infrastructurii.

# Scaling — Ce se întâmplă când ești popular?

Imaginați-vă că aplicația voastră devine virală. Mâine apare pe un site de știri și un milion de oameni încearcă să o acceseze simultan. Fără o strategie de scaling, serverul se supraîncarcă și aplicația cade. Există două abordări fundamentale:



## Scaling Vertical (Scale Up)

Folosești un server mai puternic. În loc de un motor de 2 litri, pui unul de 4 litri. Același server, dar cu mai mult CPU și RAM.

**Limitat** — nu poți crește la infinit. La un moment dat, cel mai mare server disponibil nu mai este suficient.

Analogie: *Înlocuiești motorul mașinii cu unul mai mare.*




## Scaling Horizontal (Scale Out)

Adaugi mai multe servere. În loc de o singură mașină care preia tot traficul, ai 10 mașini care împart traficul între ele. Fiecare primește o zecime din cereri.



**Practic nelimitat** — poți adăuga oricâte instanțe sunt necesare, automat sau manual.

Analogie: *Adaugi mai multe mașini în flotă.*

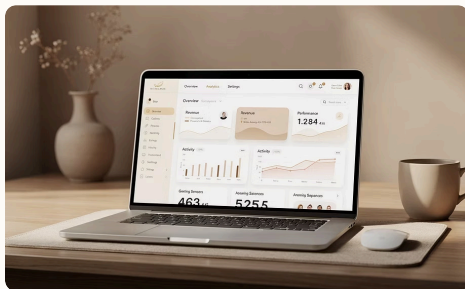
- ❏  În **Azure App Service**, Scaling Horizontal se numește **Scale Out** și poate fi activat **manual sau automat** bazat pe utilizarea CPU, numărul de cereri sau alți indicatori. Explorați Settings → Scale Out pentru a vedea opțiunile — fără să activați astăzi (implică costuri suplimentare).

# VM vs. Web App — Comparație Directă

Hai să punem față în față cele două lumi pe care le-am explorat în ultimele două sesiuni. Nici una nu este mai bună în mod absolut — alegerea corectă depinde de context.

Criteria	 Virtual Machine (IaaS)	 Web App (PaaS)
Control OS	Total — tu instalezi și configurezi	Zero — Azure gestionează
Patch-uri securitate	Responsabilitatea ta	Responsabilitatea Azure
Web server	Tu instalezi IIS/Nginx/Apache	Inclus și gestionat automat
HTTPS / SSL	Configurare manuală	Automat, inclus
Scaling	Complex, necesită configurare	Simplu, 1-click sau automat
Software custom	Orice software posibil	Limitat la runtime-uri suportate
Cost administrare	Ridicat (timp + efort)	Minim
Flexibilitate	Maximă	Moderată
Timp până la live	Ore (setup complet)	Minute

# Când alegem ce? — Ghid de decizie



## Alege Web App (PaaS) când...

- Ai un site web, o aplicație de business sau o API REST
- Echipa ta este mică și nu are administrator de sistem dedicat
- Vrei să te concentrezi pe cod, nu pe infrastructură
- Bugetul de timp este limitat
- Aplicația folosește un runtime standard (.NET, Node, PHP, Python, Java)
- Vrei deployment rapid și scalare simplă



## Alege Virtual Machine (IaaS) când...

- Ai o aplicație legacy cu dependențe specifice la nivel de OS
- Trebuie să instalezi software care nu este suportat pe PaaS
- Ai cerințe de rețea complexe sau integrări cu sisteme on-premises vechi
- Trebuie să rulezi Windows Server cu configurații specifice
- Ai nevoie de acces complet la nivel de sistem de operare
- Workload-urile nu se potrivesc modelului PaaS

# Jocul Security Consultant — Arhitectură în acțiune

Hai să gândim ca arhitecți cloud. Analizați fiecare scenariu și luați o decizie. Nu există un singur răspuns corect — important este raționamentul.

## Scenariul A — Startup mic

Un client mic vrea să lanseze un site de prezentare. Buget limitat, echipă de 2 persoane, fără administrator de sistem dedicat. Trafic estimat: câteva sute de vizitatori pe zi.

✓ **Răspuns recomandat: Web App (PaaS)** Simplu, rapid, fără administrare. Clientul se concentrează pe conținut, nu pe infrastructură. Costul lunar este predictibil și mic. Deploy în minute, nu în ore.

## Scenariul B — Companie mare, aplicație legacy

O companie mare vrea să migreze o aplicație care rulează pe Windows Server 2012, cu software specific instalat la nivel de sistem, configurații de rețea complexe și integrări cu sisteme on-premises vechi.

✓ **Răspuns recomandat: Virtual Machine (IaaS)** Controlul total este necesar. Aplicația are dependențe care nu se pot abstractiza la nivel PaaS. Costul de administrare este justificat de cerințele tehnice speciale.

## Scenariul C — E-commerce în creștere

Un magazin online cu trafic variabil — calm în timpul săptămânii, explodează în weekend și în perioadele de reduceri. Aplicația este construită în Node.js. Echipa vrea să evite downtime-ul în perioadele de vârf.

✓ **Răspuns recomandat: Web App + Auto-scaling (PaaS)** Runtime suportat, scaling automat activat, zero administrare OS. Azure crește automat numărul de instanțe în perioadele de vârf și le reduce când traficul scade.

*Aceasta este gândirea unui arhitect cloud. Nu există o soluție universală. Există **soluția potrivită pentru contextul specific**.*



# Cleanup — Ștergem resursele

## De ce este important cleanup-ul?

Resursele Azure care rulează consumă credite sau generează costuri, chiar dacă nu le folosiți activ. Un Resource Group șters înseamnă zero costuri reziduale.

Este o **bună practică** pe care orice arhitect cloud o respectă: dacă nu mai ai nevoie de o resursă, o ștergi.

## Pașii de cleanup

01

---

### Deschideți Resource Groups

În Azure Portal, navigați la **Resource Groups**

02

---

### Selectați grupul vostru

Găsiți `rg-s07-[prenumele vostru]` și dați click pe el

03

---

### Delete Resource Group

Dați click pe **Delete Resource Group**, introduceți numele pentru confirmare

04

---

### Confirmare

Confirmați și așteptați. **Toate resursele din grup dispar complet.**



# Tema — Sesiunea 7

Tema voastră consolidează înțelegerea conceptuală din sesiunea de astăzi. Scopul nu este să copiați din documentație, ci să **reformulați cu cuvintele voastre** — semn că ați înțeles cu adevărat.

1

## Diferența IaaS vs. PaaS

Explicați diferența dintre IaaS și PaaS. **Nu copiați din documentație.** Folosiți una dintre analogiile discutate astăzi (casa vs. apartament, mașina vs. ride sharing, generatorul electric) sau inventați o analogie proprie care să fie relevantă pentru voi.

2



## Când alegem Virtual Machine?

Dați un exemplu concret din lumea reală — o situație sau un tip de aplicație în care alegerea corectă este o mașină virtuală. Justificați de ce PaaS nu ar fi potrivit în acel caz. Poate fi din experiența voastră profesională sau un scenariu imaginat.

3

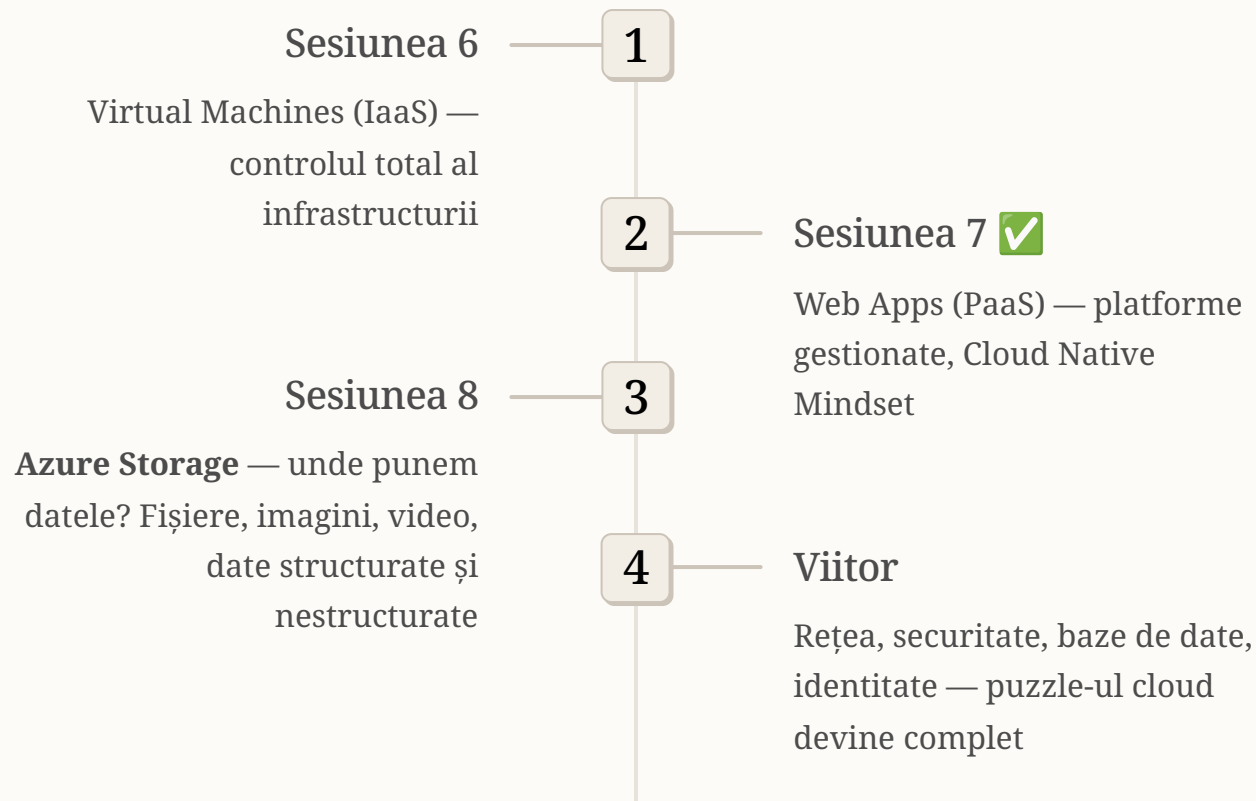
## Când alegem Web App?


Dați un exemplu concret, **diferit de exemplul din sesiunea de astăzi** (site de prezentare simplu). Poate fi o aplicație de business, o API, un portal intern — orice situație în care PaaS este alegerea corectă. Justificați.

  **Format:** Câteva paragrafe per punct, cu cuvintele voastre. Nu există răspuns perfect — contează raționamentul, nu lungimea textului.

# Ce urmează?

Astăzi ați finalizat un capitol important din înțelegerea cloud-ului.



- ❏  **Întrebarea pentru sesiunea 8:** Acum avem compute — știm unde să rulăm aplicațiile. Dar unde punem **datele**? Unde stochează Azure fișierele, imaginile, videoclipurile și bazele de date? Răspunsul la sesiunea 8.





# Rezumatul Sesiunii 7 🏆

Felicitări pentru participare! Iată ce ați învățat și realizat astăzi:

## 🧠 Concepte

IaaS vs. PaaS, Cloud Native Mindset, App Service Plan, Scaling Vertical vs. Horizontal

## 🔧 Skills practice

Ați creat un Azure Web App, ați publicat o pagină HTML live pe internet, ați explorat Kudu / Advanced Tools

## 🎯 Mindset

Gândirea arhitecturală — când alegem VM vs. Web App în funcție de context, nu de preferință

„Săptămâna trecută am învățat cum funcționează motorul. Astăzi am învățat cum să conducem mașina fără să deschidem capota.”

Vă mulțumim pentru participare. **Ne vedem la Sesiunea 8!** ☁️