



Deploy Azure Kubernetes Service (AKS) în Azure Portal

Ghid complet, de la 0 la hero — actualizat pe baza documentației Microsoft Learn disponibile până în martie 2026. Acest webinar îți explică nu doar *ce să apeși*, ci și *de ce* alegi fiecare opțiune.

WEBINAR • CURS PRACTIC

AKS STANDARD • AZURE PORTAL

Agenda webinarului

Parcurgem împreună toate etapele esențiale pentru a înțelege, proiecta și deploya un cluster AKS din Azure Portal.

01

De ce AKS?

Comparație cu alte servicii Azure și cazuri de utilizare reale.

02

Arhitectura laboratorului

Resource Group, VNet, subnet și planul IP recomandat.

03

Creare cluster pas cu pas

Toate filele din Azure Portal explicate: Basics, Node pools, Networking, Integrations, Monitoring, Advanced.

04

Autoscaling, Monitoring, Backup

Mecanismele de scalare, observabilitate și protecție a datelor.

05

AKS Standard vs. Automatic

Când alegi fiecare variantă și exerciții recomandate.

De ce AKS și nu alt serviciu?

Gândește-te la opțiunile de hosting ca la patru moduri de a călători. Fiecare are locul lui — cheia este să alegi corect în funcție de nevoile aplicației tale.

Mașini Virtuale

Tu administrezi aproape tot: OS, patching, runtime, orchestrare.
Bun pentru control total, dar cere multă muncă manuală.

Mult control / multă muncă

Azure App Service

Excelent pentru aplicații web clasice. Platforma ascunde serverele, dar flexibilitatea pentru containere complexe este mai mică.

Simplu pentru web apps

Azure Container Apps

Foarte bun pentru microservicii și API-uri containerizate fără să administrezi Kubernetes. Ideal când vrei simplitate și scale rapid.

Serverless pentru containere

Azure Kubernetes Service

Alegi AKS când ai nevoie de orchestrare Kubernetes completă, mai multe servicii, politici, rețea avansată, observabilitate și control enterprise.

Putere + flexibilitate

Figura 1. Poziționarea AKS față de alte opțiuni Azure pentru hosting de aplicații. Mai multă putere și control enterprise →

Când alegi AKS — și când NU

✓ Alege AKS când...

- Ai mai multe microservicii, API-uri, workers, joburi batch sau componente stateful care trebuie coordonate.
- Ai nevoie de Kubernetes real, nu doar de „rulat containere”.
- Vrei politici enterprise, separare între workload-uri, control fin pe networking, identity, scaling și observabilitate.
- Ai nevoie de portabilitate: aceeași logică Kubernetes poate fi mutată sau replicată mai ușor între medii.
- Ai echipe DevOps / Platform Engineering care pot valorifica flexibilitatea Kubernetes.

✗ NU alege AKS când...

- Pentru o aplicație web simplă fără nevoi speciale — **App Service** este de obicei mai simplu.
- Pentru containere stateless cu nevoie minimă de administrare — **Azure Container Apps** este adesea mai rapid de pus în producție.
- Dacă echipa nu are încă maturitate operațională pe Kubernetes, AKS poate aduce complexitate inutilă.

De ce AKS Standard în acest curs?

Microsoft oferă și **AKS Automatic**, o experiență mai gestionată. Este excelentă pentru anumite scenarii, însă pentru un curs *zero to hero* vrem să înțelegem mecanica din spate: rețea, node pools, autoscaling, monitoring și opțiuni de securitate. De aceea construim un cluster AKS Standard din portal, pas cu pas.

Ghid rapid: ce serviciu pentru ce scenariu

Scenariu	Serviciul mai potrivit	De ce
Website sau API simplu	Azure App Service	Mai puține decizii operaționale și administrare minimă.
Microservicii simple, event-driven, fără Kubernetes	Azure Container Apps	Scale ușor și ascunde o bună parte din complexitatea infrastructurii.
Aplicație legacy care cere acces complet la OS	Virtual Machines	Ai control total asupra sistemului și middleware-ului.
Platformă modernă multi-serviciu, cu nevoi enterprise	AKS	Orchestrare Kubernetes completă și control avansat.

Arhitectura pe care o vom construi

Model recomandat pentru un laborator educațional și bun punct de plecare pentru producție.

Resource Group

rg-aks-lab-weu-01 — Un Resource Group dedicat păstrează toate resursele împreună și permite ștergerea ușoară a laboratorului.

VNet + Subnet

vnet-aks-lab-weu-01 (10.20.0.0/16) cu subnet dedicat nodurilor AKS **snnet-aks-nodes (10.20.0.0/23)**.

Cluster AKS

Control plane administrat de Azure + **System node pool** (kubernetes) + **User node pool** (App pods, API pods, Jobs).

Observabilitate

Integrare cu **Azure Monitor**: Container Insights, Managed Prometheus, Grafana și loguri control plane.

Backup

Opțiuni de backup prin **Azure Backup** și **Backup Vault** pentru resurse cluster și volume persistente.

Ingress / Load Balancer

Standard Load Balancer pentru expunerea serviciilor și endpointurilor aplicațiilor.

Figura 2. Arhitectura logică a laboratorului AKS prezentat în acest ghid.

Ce trebuie să ai înainte să începi

1 Abonament Azure activ

Permisuni suficiente pentru a crea Resource Groups, rețele, AKS și resurse de monitorizare.

2 Regiune aleasă

Pentru exemple vom folosi **West Europe**, dar poți adapta la orice regiune disponibilă.

3 Convenție de naming

Exemplu: `rg-aks-lab-weu-01`, `vnet-aks-lab-weu-01`, `aks-lab-weu-01`.

4 Claritate asupra scopului

Laborator educațional, nu cluster de producție critică.

5 Acces la terminal

Azure Cloud Shell sau terminal local cu `az CLI` și `kubectl`, pentru verificările post-deploy.

Notă practică

Documentația Microsoft precizează că ghidul rapid din portal este pentru evaluare și că pentru producție trebuie raportat la arhitectura de referință AKS. În acest material facem un pas mai departe și explicăm nu doar *ce să apeși*, ci și *de ce* alegem fiecare opțiune.

Planificarea rețelei: de ce contează înainte de AKS

Una dintre cele mai frecvente greșeli în AKS apare **înainte** de a crea clusterul: planificarea slabă a rețelei. AKS are nevoie de un plan IP suficient pentru noduri, poduri, servicii și operațiuni de upgrade.

VNet 10.20.0.0/16

Un /16 este generos și prietenos pentru cursuri și medii de test. Îți oferă spațiu să experimentezi și să adaugi ulterior subnets pentru ingress, private endpoints, jumpbox sau alte laboratoare.

Subnet noduri 10.20.0.0/23

Un /23 oferă echilibru bun între simplitate și spațiu de manevră: suficient pentru laborator, upgrade surge, node pools suplimentare și resurse auxiliare. Mai sănătos decât un /27 sau /28 care produce rapid blocaje.

Service CIDR 10.30.0.0/16

Trebuie să nu se suprapună cu VNet sau alte rețele conectate. DNS service IP: 10.30.0.10 — o adresă din Service CIDR pentru kube-dns / CoreDNS.

Pod CIDR Overlay 192.168.0.0/16

Cu Azure CNI Overlay, doar nodurile iau IP-uri din subnet, iar podurile folosesc acest CIDR overlay separat. Fiecare nod primește un /24 din spațiul overlay.

Figura 3. Un plan IP simplificat și ușor de înțeles pentru un laborator AKS. Exemplul separă clar rețeaua nodurilor de spațiul serviciilor.

Formula de sizing și alegerea networking mode

Formula de gândire pentru sizing

Microsoft explică faptul că subnetul trebuie să fie suficient de mare pentru noduri, poduri și resursele Kubernetes și Azure din cluster, iar în calcul trebuie să intre și noduri suplimentare pentru upgrade. Pentru Azure CNI tradițional, formula orientativă este:

$$(\text{noduri} + \text{max surge}) + ((\text{noduri} + \text{max surge}) \times \text{max pods/nod})$$

Chiar dacă recomandăm Azure CNI Overlay, această formulă te învață un principiu important: în Kubernetes nu planifici doar pentru ziua 1, ci și pentru scaling și upgrade.

De ce recomandăm Azure CNI Overlay

Pentru un curs modern în 2026, recomandăm să pornești de la **Azure CNI Overlay**:

- Nodurile primesc IP-uri din subnet; podurile dintr-un CIDR overlay separat.
- Simplifică enorm planificarea IP și reduce riscul de address exhaustion.
- Subnetul nodurilor nu mai este sufocat de IP-urile pentru poduri.
- Comunicarea între poduri rămâne performantă.
- Claritate și scalabilitate fără dureri timpurii de adresare.

Opțiune	Ce înseamnă	Când are sens
Azure CNI Overlay	Nodurile iau IP-uri din subnet; podurile dintr-un CIDR overlay separat.	Alegerea recomandată pentru curs și multe deploymenturi moderne.
Azure CNI Node Subnet	Atât nodurile, cât și designul rețelei sunt mai strâns legate de VNet.	Când ai cerințe specifice de adresare și integrare directă în rețeaua Azure.
Kubenet / modele mai vechi	Mai puțin recomandate pentru design nou în 2026.	Doar în contexte moștenite sau foarte specifice.

Creează Resource Group-ul

01

Navighează în portal

Intră în **portal.azure.com** și caută **Resource groups**.

02

Selectează Create

Apasă butonul **Create** pentru a începe configurarea.

03

Completează detaliile

Alege **Subscription**-ul corect. La **Resource group name** folosește:
`rg-aks-lab-weu-01`.

04

Alege regiunea

La **Region** alege aceeași regiune în care vei crea clusterul, de exemplu **West Europe**.

05

Finalizează

Apasă **Review + create**, apoi **Create**.

De ce acum și separat?

Un Resource Group dedicat îți oferă:

- **Ordine** — toate resursele laboratorului într-un singur loc.
- **Cost tracking** mai simplu — poți vedea exact cât costă laboratorul.
- **Ștergere ușoară** — la finalul cursului ștergi tot Resource Group-ul și nu rămân resurse orfane care generează costuri.

Creează VNet-ul și subnetul

01

Navighează la Virtual networks

În portal caută **Virtual networks** și selectează **Create**.

02

Configurează de bază

Subscription: același abonament ca pentru Resource Group. Resource group: `rg-aks-lab-weu-01`. Name: `vnet-aks-lab-weu-01`. Region: aceeași regiune ca AKS.

03

Setează spațiul de adrese

IPv4 address space: `10.20.0.0/16`.

04

Adaugă subnetul

Adaugă un subnet nou: `snet-aks-nodes` cu prefix `10.20.0.0/23`.

05

Finalizează

Apasă **Review + create** și apoi **Create**.



Notă practică

Nu complica laboratorul cu multe subnets din prima. Un subnet bine ales pentru noduri este suficient pentru început. Subnets suplimentare se adaugă când introduci cerințe noi: private endpoints, firewall, ingress dedicat sau integrare hibridă.

Harta portalului: fluxul de creare a clusterului AKS

Înainte să creezi clusterul, este important să înțelegi ce te așteaptă în fiecare filă din Azure Portal. Ordinea exactă poate varia ușor în funcție de preview-uri active, dar logica rămâne aceeași.



Figura 4. Fluxul logic al filelor din Azure Portal pentru crearea clusterului AKS.

Creează clusterul AKS: Basics și Identitate

În portal, selectează **Create a resource** → **Containers** → **Azure Kubernetes Service (AKS)**.

Setare	Ce recomand în curs	De ce
Subscription	Abonamentul de laborator	Separă costurile și permisiunile.
Resource group	rg-aks-lab-weu-01	Păstrezi toate resursele împreună.
Cluster preset configuration	Dev/Test sau echivalent laborator	Cost mai mic și complexitate redusă pentru învățare.
Kubernetes cluster name	aks-lab-weu-01	Naming clar și ușor de urmărit.
Region	West Europe	Apropiere și consistență cu celelalte resurse.
AKS pricing tier	Free pentru laborator; Standard pentru producție	În laborator reduci costul. În producție alegeți Standard pentru capacități suplimentare.
Availability zones	None în laborator de bază	Simplifici lecția. În producție ai discuție separată despre reziliență zonală.
Kubernetes version	O versiune suportată, nedefazată	Nu alege o versiune prea veche doar pentru „stabilitate” aparentă.

Identitate și acces

Preferința pentru un curs modern: **Managed Identity** (reduce povara secretelor față de service principals) și **Azure RBAC** pentru Kubernetes (simplifică guvernarea cu identitățile Azure deja cunoscute). Pentru primul laborator nu complica cu acces privat la API server, decât dacă obiectivul lecției este networking avansat. Dacă vezi opțiunea de Fleet manager, las-o pe None — Fleet este util când administrezți mai multe cluster.

Configurarea Node Pools

Node pool-ul este grupul de VM-uri pe care rulează podurile tale. În AKS, **system pool** găzduiește componente esențiale, iar **user pool** este locul natural pentru workload-urile aplicației.

Setare	Recomandare de curs	De ce
System node pool	Lasă pool-ul de sistem activ	Clusterul are nevoie de el pentru componentele de bază.
User node pool	Poți adăuga unul separat chiar de la început	Înveți separarea corectă între platformă și aplicație.
VM size	D2s_v5 sau echivalent mic, dar modern	Bun pentru laborator; 2 vCPU și memorie decentă pentru teste.
OS SKU	Ubuntu Linux sau Azure Linux 3	Linux este alegerea naturală pentru majoritatea workload-urilor AKS.
Node count	2 noduri pentru laborator serios; 1 doar pentru demo minimalist	2 noduri oferă reziliență minimă și comportament mai realist.
Mode	System pentru pool-ul principal, User pentru aplicații	Separare operațională sănătoasă.
Azure Spot	Nu pentru prima lecție	Util pentru optimizare de cost, dar complică explicația prin posibile evacuări.

Autoscaling pe node pools

Recomandare: **Enable autoscaler = On**, Min nodes: **2**, Max nodes: **5**. De ce nu 1-1? Pentru că atunci nu înveți nimic despre elasticitate. De ce nu 1-50? Pentru că laboratorul trebuie să fie controlat și predictibil. Manual scaling și cluster autoscaler nu sunt filosofii echivalente — dacă activezi autoscaler, lasă-l să facă treaba lui în intervalul min-max setat.

Configurarea rețelei clusterului

Setare	Recomandare	De ce
Network model	Azure CNI Overlay	Mai simplu pentru planificare IP și foarte potrivit pentru învățare.
Virtual network	vnet-aks-lab-weu-01	Folosim VNet-ul creat intenționat înainte.
Subnet	snet-aks-nodes	Separăm clar clusterul de alte resurse.
Pod CIDR	192.168.0.0/16	Spațiu suficient pentru overlay și ușor de reținut în curs.
Service CIDR	10.30.0.0/16	Trebuie să nu se suprapună cu VNet sau alte rețele conectate.
DNS service IP	10.30.0.10	O adresă din Service CIDR pentru kube-dns / CoreDNS.
Load balancer	Standard	Alegerea sănătoasă și curentă pentru majoritatea scenariilor.

Greșeală clasică și dureroasă

Service CIDR nu trebuie să se suprapună nici cu subnetul nodurilor, nici cu alte rețele conectate. Verifică de două ori înainte de a apăsa Create!

În această etapă poți lăsa configurarea de ingress la nivel minim și să o tratezi într-o lecție separată. Important este ca studentul să înțeleagă diferența dintre Service de tip **ClusterIP**, **NodePort** și **LoadBalancer**. În producție, foarte des vei introduce un controller de ingress și eventual Application Gateway sau NGINX.

Integrations, Monitoring și Advanced

Fila Integrations

Fii selectiv. Nu activa tot doar pentru că există.

- **Container Registry:** util când vrei să împingi imagini private.
- **Azure Policy:** foarte utilă în medii enterprise, dar poate fi introdusă după ce studentul înțelege bazele.
- **Secret management:** explică întâi principiul, apoi implementează Key Vault CSI sau alte integrări într-o lecție separată.

Fila Advanced / Security

- **Auto-upgrade channel:** bun pentru a discuta despre patching și responsabilitate partajată.
- **Maintenance windows:** foarte util în producție, ca să controlezi când se fac operațiuni automate.
- **Private cluster / private API server:** important pentru securitate avansată, dar nu obligatoriu în primul laborator.
- **Public access la API server:** acceptabil pentru laborator dacă este controlat și înțeles; în producție se analizează mult mai strict.

Fila Monitoring — ce activezi

Opțiune	Recomandare
Container Insights	✓ On
Managed Prometheus	✓ On
Azure Managed Grafana	✓ Recomandat în demo
Control plane logs	✓ On
Log Analytics Workspace	✓ Dedicat laboratorului

Dacă bugetul este sensibil, explică elevilor că monitorizarea completă aduce costuri. În producție merită aproape întotdeauna. În laborator poți decide dacă vrei experiență completă sau cost minim.

Review + Create: lista de verificare finală

1

Verifică toate filele

Parcurge fiecare filă și confirmă că setările sunt corecte înainte de a trimite.

2

Aceeași regiune

Confirmă că regiunea este aceeași pentru RG, VNet și AKS.

3

Service CIDR fără overlap

Confirmă că Service CIDR nu se suprapune cu VNet sau alte rețele conectate.

4

Autoscaler sensibil

Confirmă că autoscalerul are min și max sensibile (ex: min 2, max 5).

5

Rulează validarea

Apasă **Validate** și rezolvă orice erori înainte de a continua.

6

Apasă Create

Crearea clusterului poate dura câteva minute. După finalizare, intră în resursa AKS și verifică Overview, Node pools, Networking și Monitoring.

Verificări obligatorii după deploy

Nu închide lecția imediat după „deployment succeeded”. Un cluster sănătos trebuie verificat operațional, nu doar administrativ.

Verificări în Azure Portal

- Intră în **AKS** → **Overview** și confirmă că starea clusterului este **Running**.
- Intră în **Node pools** și verifică dacă pool-urile sunt **Ready**.
- Intră în **Networking** și confirmă VNet-ul, subnetul și CIDR-urile.
- Intră în **Monitoring** și verifică dacă integrarea a pornit corect.

Verificări cu kubectl

Obține credențialele clusterului:

```
az aks get-credentials \  
--resource-group rg-aks-lab-weu-01 \  
--name aks-lab-weu-01
```

Confirmă că nodurile sunt Ready:

```
kubectl get nodes
```

Observă podurile de sistem:

```
kubectl get pods -A
```

Autoscaling în AKS: toate mecanismele explicate

Autoscaling în AKS nu înseamnă un singur mecanism, ci mai multe mecanisme care operează la niveluri diferite. Cheia este să înțelegi **întrebarea la care răspunde fiecare opțiune.**

Mecanism	La ce răspunde	Unde îl folosești
Cluster Autoscaler	Avem suficiente noduri?	La nivel de node pool.
Horizontal Pod Autoscaler (HPA)	Avem suficiente replici de poduri?	La nivel de Deployment/ReplicaSet.
Vertical Pod Autoscaler (VPA)	Fiecare pod are request/limit potrivit?	La nivel de resurse CPU/memory per workload.
KEDA	Trebuie să scalăm după evenimente?	Queue-uri, mesaje, workload-uri event-driven.
Node Auto-Provisioning (NAP)	Ce tip de noduri ar trebui create automat?	Scenarii moderne, cu Karpenter și optimizare mai dinamică.

Mecanismele de autoscaling în detaliu

Cluster Autoscaler

Primul mecanism activat într-un curs de bază. Urmărește podurile care nu pot fi programate și decide dacă trebuie adăugate noduri. Poate reduce nodurile neutilizate când este sigur. Se configurează per node pool cu valori min și max. **Nu combina** logica lui cu alte mecanisme paralele de autoscaling pentru VM Scale Set.

Horizontal Pod Autoscaler (HPA)

Răspunde la câte replici de aplicație ai, nu câte noduri. Dacă CPU, memorie sau alte metrice cresc, HPA mărește numărul de poduri. Foarte bun pentru API-uri și servicii stateless. Lucrează excelent împreună cu cluster autoscaler: HPA cere mai multe poduri, iar cluster autoscaler adaugă noduri dacă nu mai există spațiu. Se configurează la nivel de workload Kubernetes, nu în wizard-ul inițial.

Vertical Pod Autoscaler (VPA)

Ajustează cererile și limitele de CPU și memorie ale containerelor pe baza utilizării istorice. Util când echipa nu a dimensionat bine workload-urile și vrea recomandări sau ajustări automate.

KEDA

Pentru autoscaling event-driven. Ca un ospătar care nu se uită doar câți clienți sunt deja la masă, ci câți oameni stau la coadă la intrare. Dacă ai queue-uri, topicuri sau evenimente externe, KEDA este extraordinar.

Node Auto-Provisioning (NAP)

Folosește tehnologia Karpenter și merge dincolo de cluster autoscaler-ul clasic. Poate decide mai dinamic ce SKU și ce tip de noduri sunt potrivite pentru workload-urile aflate în așteptare. Direcție modernă AKS — prezintă-l ca lecție avansată după ce studentul înțelege node pools și cluster autoscaler.

Monitoring în AKS: toate straturile de observabilitate

O companie matură nu consideră clusterul „gata” doar pentru că rulează. **Un cluster bun este un cluster observabil.** În AKS, observabilitatea vine din mai multe surse care lucrează împreună.



Platform Metrics

Metricile de bază ale resursei Azure. Se colectează automat și le poți vedea în **Metrics Explorer**. Excelente pentru alerte rapide și o primă vedere asupra sănătății.



Container Insights

Arată ce se întâmplă în cluster: noduri, poduri, containere, evenimente și perspective operaționale. Studentul vede imediat legătura dintre obiectele Kubernetes și resursele care consumă CPU sau memorie.



Managed Prometheus

Experiență administrată pentru metrici Prometheus. În lumea Kubernetes, Prometheus este limbajul nativ al metricilor. Dacă vrei să predai AKS serios, nu ocoli acest subiect.



Azure Managed Grafana

Partea vizuală care îi încântă pe studenți și îi ajută să înțeleagă rapid starea clusterului. Foarte utilă pentru dashboard-uri și demo-uri.



Control Plane Logs

Când apar probleme reale, logurile control plane pot face diferența dintre presupuneri și diagnostic serios. Recomandate în orice mediu care vrea un minim de disciplină operațională.

Backup în AKS: protecția datelor fără mituri

Backup-ul în AKS trebuie explicat fără mituri. Kubernetes nu înseamnă doar YAML; în multe aplicații ai și volume persistente, iar acestea trebuie protejate corect. **Azure Backup pentru AKS** folosește o extensie instalată în cluster și lucrează împreună cu un Backup Vault.

Ce poate proteja Azure Backup pentru AKS

- Resursele clusterului.
- Volume persistente bazate pe **Azure Disk**.
- Volume persistente bazate pe **Azure Files SMB**, cu limitările documentate de Microsoft.

Tier-uri de backup

Tier	Ce înseamnă	Când îl alegi
Operational Tier	Snapshot-uri operaționale pentru recovery mai rapid.	Backup operațional pe termen mai scurt.
Vault Tier	Stocare pe termen mai lung în Backup Vault.	Retention mai lungă și reziliență mai puternică pentru Azure Disk.

Important: pentru Azure Files, retenția este limitată și Vault Tier nu este suportat în același mod ca pentru Azure Disk — merită explicat clar că nu toate persistentele sunt egale.

Pași în portal

01

Creează un Backup Vault

Configurează redundanța și, dacă ai nevoie, Cross Region Restore.

02

Creează o Backup Policy

Pentru Kubernetes Services, cu politicile de retenție dorite.

03

Instalează Backup Extension

Intră în AKS → Settings → Backup. Indică storage account și blob container.

04

Configurează și validează

Selectează vault-ul, policy-ul, resursele de protejat, snapshot resource group și validează rolurile.

AKS Standard versus AKS Automatic

AKS Automatic este o experiență administrată care automatizează și mai mult setările și operațiunile clusterului. Merită menționat în 2026 deoarece face parte clar din peisajul actual AKS.

AKS Standard

- **Control:** Mai mult control explicit asupra fiecărei setări.
- **Învățare:** Excelent pentru înțelegerea mecanicii interne.
- **Node management:** Gestionat prin node pools și politici definite de tine.
- **Când îl predai primul:** Da, într-un curs zero-to-hero.
- **Ideal pentru:** Studenți care vor să înțeleagă rețea, autoscaling, monitoring și securitate în profunzime.

AKS Automatic

- **Control:** Mai multă automatizare și guardrails implicite.
- **Învățare:** Mai puțin pentru „mecanica internă”, mai mult pentru experiență accelerată.
- **Node management:** Mai mult automatizat, inclusiv cu Node Autoprovisioning.
- **Când îl predai primul:** Ca modul avansat sau comparativ.
- **Ideal pentru:** Echipe care vor să ajungă rapid în producție cu mai puțin overhead operațional.

Configurația recomandată de curs, pe scurt

Referință rapidă pentru toate setările recomandate în acest laborator educațional.

Componentă	Valoare recomandată
Resource Group	rg-aks-lab-weu-01
VNet	10.20.0.0/16
Subnet noduri	10.20.0.0/23
AKS mode	AKS Standard
Network mode	Azure CNI Overlay
Pod CIDR	192.168.0.0/16
Service CIDR	10.30.0.0/16
DNS service IP	10.30.0.10
System node pool	2 x D2s_v5 sau echivalent mic și modern
Autoscaler	On, min 2, max 5
Monitoring	Container Insights + Managed Prometheus + control plane logs
Grafana	Opțional, dar recomandat pentru demo
Backup	Explicat complet; activat după nevoie și buget

Greșeli frecvente pe care merită să le eviți

⚠ Subnet prea mic

Pare suficient în ziua 1, dar explodează la primul upgrade sau scale-out. Alege /23 sau mai mare, nu /27 sau /28.

⚠ CIDR-uri care se suprapun

Service CIDR, Pod CIDR și VNet trebuie să fie complet separate. O suprapunere produce erori greu de diagnosticat.

⚠ Prea multe opțiuni activate fără înțelegere

Activarea tuturor integrărilor fără să înțelegi ce fac și ce cost au duce la facturi surpriză și configurații fragile.

⚠ Fără separare system / user workloads

Lipsa separării dintre system și user node pools creează instabilitate și face troubleshooting-ul mult mai dificil.

⚠ Fără monitorizare

Fără monitorizare, dar cu așteptări mari de troubleshooting. Un cluster fără observabilitate este un cluster pe care nu îl poți opera serios.

⚠ Confuzia între scaling de noduri și scaling de poduri

Cluster Autoscaler ≠ HPA. Înțelege întrebarea la care răspunde fiecare mecanism înainte de a-l activa.

⚠ AKS fără justificare arhitecturală

Alegerea AKS doar pentru că „este cool”, fără să ai nevoie reală de orchestrare Kubernetes, aduce complexitate inutilă.

Exerciții recomandate pentru studenți

Practică este esențială. Aceste exerciții consolidează conceptele din webinar și te duc cu adevărat de la zero la hero.



kubectl de bază

Conectează-te cu kubectl și listează nodurile și podurile de sistem. Verifică starea fiecărui component.



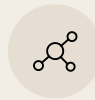
Primul deployment

Adaugă un deployment simplu cu 2 replici și observă-l în Container Insights. Urmărește metricile în timp real.



HPA + Cluster Autoscaler

Creează un HPA și observă relația dintre HPA și cluster autoscaler. Generează load și urmărește comportamentul.



Al doilea user node pool

Adaugă un al doilea user node pool și mută workload-ul prin nodeSelector sau taints/tolerations.



Cost-conscious design

Simulează un design orientat pe cost și explică de ce Spot nu a fost activat în lecția de bază.



Schema IP

Desenează schema IP a clusterului și explică de ce Service CIDR nu trebuie să se suprapună cu VNet.

Concluzie: de la zero la hero

AKS merită ales atunci când compania are nevoie de orchestrare Kubernetes reală, control operațional și flexibilitate enterprise. Pentru un student, cheia nu este doar să știe să apese **Create**, ci să înțeleagă de ce a ales acel subnet, acel network mode, acel autoscaler și acel model de monitoring. Dacă elevul înțelege aceste decizii, atunci a trecut cu adevărat de la zero la hero.

Arhitectură solidă

Resource Group dedicat, VNet /16, subnet /23, Azure CNI Overlay — fundamente clare pentru orice scenariu.

Configurare înțeleasă

Fiecare setare din portal are o justificare. Nu apeși butoane — iei decizii arhitecturale.

Cluster observabil

Container Insights, Managed Prometheus, Grafana și control plane logs — vizibilitate completă.

Pregătit pentru producție

Principiile din acest laborator sunt direct aplicabile în medii enterprise reale.

Anexă: Surse Microsoft Learn

Acest material a fost actualizat pe baza documentației Microsoft Learn disponibile până în **martie 2026**.

Quickstart & Arhitectură

Quickstart: Deploy an Azure Kubernetes Service (AKS) cluster using the Azure portal • Baseline Architecture for an AKS Cluster

Networking

IP address planning in Azure Kubernetes Service (AKS) • Overview of Azure CNI Overlay networking in AKS • Configure Azure CNI networking in AKS

Monitoring

Monitor Azure Kubernetes Service (AKS) • Enable monitoring for AKS clusters - Azure Monitor

Backup

Back up Azure Kubernetes Service by using Azure Backup • Prerequisites for backing up AKS with Azure Backup

Scaling

Scaling options for applications in AKS • Overview of node auto-provisioning (NAP) in AKS • Cluster autoscaler overview in AKS