# Azure Monitor, Observabilitate și SRE

Ghid complet pentru începători și practicieni — de la zero la hero: concepte, arhitectură, portal Azure, exemple reale, AKS, Application Insights, Log Analytics, alerte, workbooks și bune practici de operare.

## Ce vei înțelege

Cum funcționează Azure Monitor ca platformă de observabilitate cap-coadă: SLI, SLO, error budget, alert fatigue, cost control și triere operațională.

## Ce vei configura

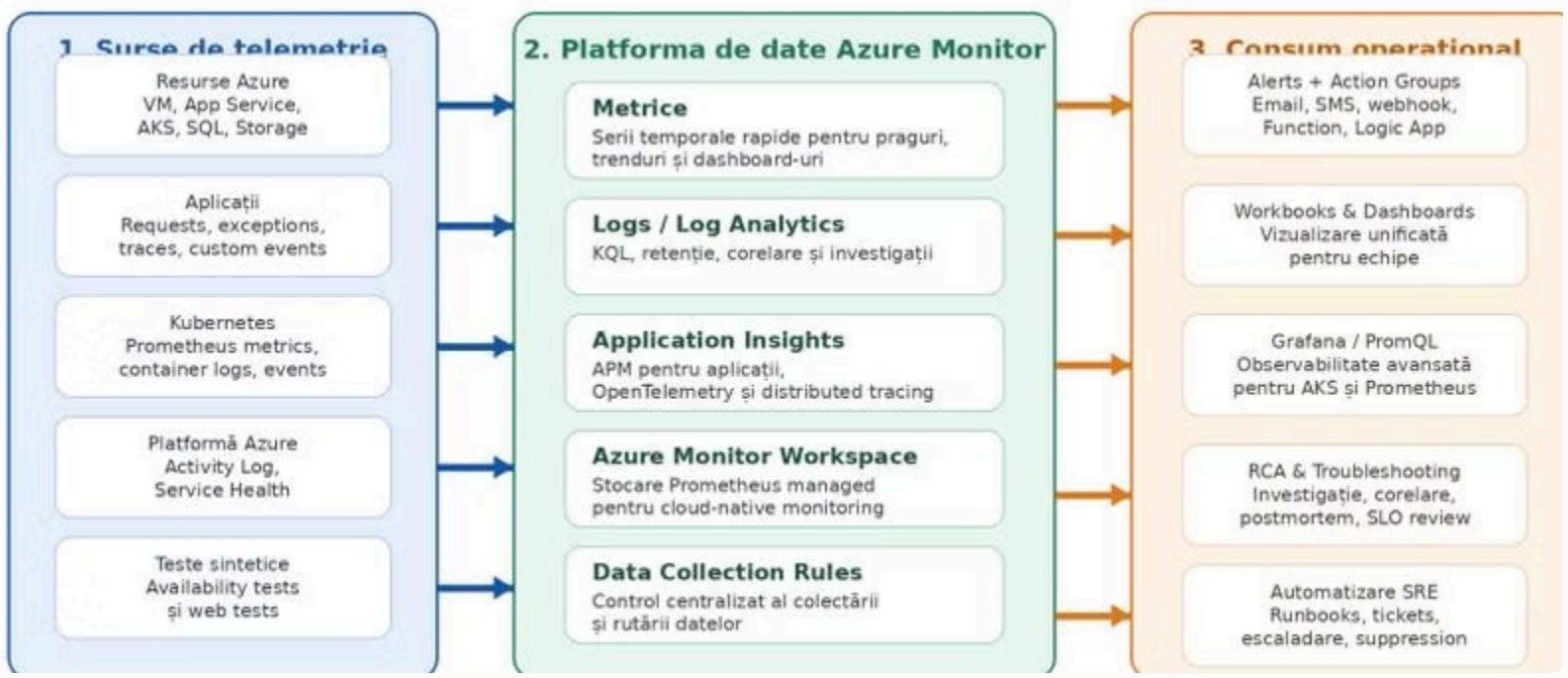
Workspace-uri, alerte, action groups, synthetic monitoring și vizualizare în portalul Azure.

## Unde aplici

Aplicații web, infrastructură Azure, workloads cloud-native și AKS.

# Arhitectura Azure Monitor — Vedere de Ansamblu

De la colectare de date la alertare, analiză și răspuns operațional. Observabilitatea nu este un singur produs, ci un lanț complet: colectezi semnale, le stochezi corect, le transformi în informații utile și reacționezi rapid.




### 1. Surse de telemetrie

- Resurse Azure: VM, App Service, AKS, SQL, Storage
- Aplicații: requests, exceptions, traces, custom events
- Kubernetes: Prometheus metrics, container logs, events
- Platformă Azure: Activity Log, Service Health
- Teste sintetice: Availability tests și web tests

### 2. Platforma de date Azure Monitor

- **Metrice**: Serii temporale rapide pentru praguri, trenduri și dashboard-uri
- **Logs / Log Analytics**: KQL, retenție, corelare și investigații
- **Application Insights**: APM, OpenTelemetry și distributed tracing
- **Azure Monitor Workspace**: Stocare Prometheus managed
- **Data Collection Rules**: Control centralizat al colectării și rutării datelor

### 3. Consum operațional

- Alerts + Action Groups: Email, SMS, webhook, Function, Logic App
- Workbooks & Dashboards: Vizualizare unificată pentru echipe
- Grafana / PromQL: Observabilitate avansată pentru AKS și Prometheus
- RCA & Troubleshooting: Investigație, corelare, postmortem, SLO review
- Automatizare SRE: Runbooks, tickets, escaladare, suppression

CAPITOL 1

# De ce este Azure Monitor Fundamental într-o Practică SRE

În multe organizații, monitorizarea este tratată prea târziu: după ce aplicația există, după ce utilizatorii se plâng, după ce apare primul incident sever. În realitate, monitorizarea nu este o activitate de "după" — este fundația pe care se construiește operarea sigură a unui serviciu digital.

> Dacă infrastructura este motorul, iar aplicația este mașina, Azure Monitor este bordul de instrumente, cutia neagră și sistemul de alarmă în același timp. Fără el, conduci rapid, dar aproape orb.

### SRE ca disciplină

SRE există pentru a ajuta organizația să atingă un nivel potrivit de fiabilitate într-un mod sustenabil — nu "zero incidente" naiv, ci gestionarea fiabilității cu obiective clare, date și feedback continuu.

### Semnale din toate straturile

Azure Monitor este relevant pentru SRE fiindcă oferă semnale din toate straturile: platformă Azure, resurse IaaS/PaaS, aplicații, Kubernetes, Prometheus, synthetic monitoring și health events.

### Puterea corelării

Valoarea nu vine doar din colectare, ci din corelare: metricele îți spun că există o problemă, logurile te ajută să o explici, iar traces/telemetry îți arată exact unde s-a produs.

O echipă matură nu întreabă "avem monitorizare?", ci "putem detecta rapid o degradare, o putem înțelege, putem alerta doar persoana potrivită și putem învăța ceva după incident?"

# Ce Înseamnă SRE în Limbaj Practic

Site Reliability Engineering este o disciplină inginească orientată spre fiabilitate, nu doar o funcție operațională. SRE înseamnă să tratezi producția ca pe un produs viu, măsurat continuu, nu ca pe o zonă unde "sperăm să meargă".

## SLI

**Service Level Indicator** — ce măsori efectiv: latența P95 sau procentul de request-uri 2xx/3xx.

## SLO

**Service Level Objective** — ținta internă: de exemplu 99.9% disponibilitate pe 30 de zile.

## SLA

**Service Level Agreement** — promisiunea contractuală făcută clientului.

## Error Budget

Spațiul permis pentru eșec fără a depăși SLO-ul; te ajută să echilibrezi viteză versus stabilitate.

## Golden Signals

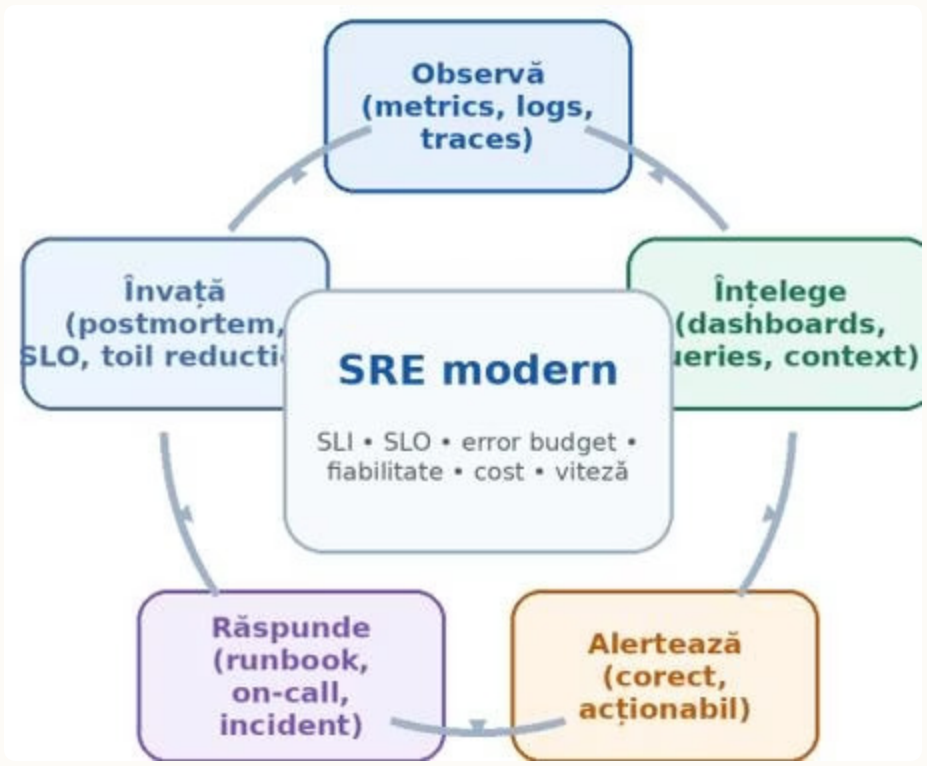
Latență, trafic, erori, saturație — model mental de pornire extrem de util.

## Toil

Muncă repetitivă, manuală, fără valoare inginească pe termen lung. Un SRE matur o reduce prin automatizare.

# Bucla SRE: De la Telemetrie la Îmbunătățire Continuă

Azure Monitor devine valoros doar când datele duc la decizii și automatizare. Bucla SRE modernă este un ciclu continuu de observare, înțelegere, alertare, răspuns și învățare.




1

### Observă

Metrics, logs, traces — colectezi semnale din toate straturile.

2

### Înțelege

Dashboards, queries, context — transformi datele în informații utile.

3

### Alertează

Corect, acționabil — notifici persoana potrivită la momentul potrivit.

4

### Răspunde

Runbook, on-call, incident — acționezi rapid și structurat.

5

### Înveți

Postmortem, SLO review, toil reduction — îmbunătățești continuu.

# Ce Problemă Rezolvă Azure Monitor într-o Companie Reală

| Situație | Fără observabilitate | Cu Azure Monitor bine configurat |
|---|---|---|
| Descoperirea incidentelor | Din reclamațiile utilizatorilor | Detectate înainte de impact major sau imediat după debut |
| Calitatea alertelor | Multe alerte, puține utile | Alerte mapate pe severitate, owner și runbook |
| Corelarea datelor | Date fragmentate în silozuri | Metrice, logs și traces corelate în aceeași investigație |
| Vizibilitate AKS | AKS este o "cutie neagră" | Prometheus, Container insights și Grafana dau vizibilitate detaliată |
| Învățare după incident | Nu se învață nimic | Postmortem-ul ajustează SLO, alerte și dashboard-uri |

CAPITOL 2

# Cum este Construit Azure Monitor

Azure Monitor nu este un singur produs cu un singur ecran. Este o platformă compusă din mai multe piese care lucrează împreună. Dacă înțelegi clar fiecare piesă, portalul devine mult mai ușor de urmărit.

## Azure Monitor

Umbrela principală — platforma centrală de observabilitate.

## Metrics

Date numerice time-series, rapide, excelente pentru grafice și alertare aproape în timp real.

## Logs

Date bogate, interogabile cu KQL, excelente pentru investigații, corelare și analiză complexă.

## Application Insights

APM pentru aplicații, bazat modern pe OpenTelemetry pentru multe scenarii.

## Log Analytics Workspace

Spațiul principal în care locuiesc logurile Azure Monitor — centrul pentru KQL și investigații.

## Azure Monitor Workspace

Workspace separat folosit în special pentru metricile Prometheus managed.

### Data Collection Rules (DCR)

Mecanism centralizat care definește ce se colectează și unde se trimite.

### Alerts, Action Groups, Alert Processing Rules

Lanțul prin care datele devin notificări și acțiuni.

### Workbooks și Grafana

Zona de consum și vizualizare avansată pentru echipe și management.

# Logs, Metrics și Traces: Când Alegi Fiecare

Regula simplă: dacă vrei să vezi **"cât de mult"** și să alertezi repede, începi cu **metrics**. Dacă vrei să înțelegi **"de ce s-a întâmplat"**, folosești **logs**. Dacă vrei să urmărești un request prin întregul flux aplicațional, ai nevoie de **traces**.

### 📊 Metrics

**Puncte forte:** Foarte rapid, time-series, cost/control bun pentru praguri.

**Exemple:** CPU, memory, request rate, node pressure, latency P95.

### 📋 Logs

**Puncte forte:** Bogate în context, interogare flexibilă, investigație detaliată.

**Exemple:** Erori aplicație, audit, kube events, stdout/stderr, SecurityEvent.

### 🔍 Traces / APM

**Puncte forte:** Legătură end-to-end între operații și dependențe.

**Exemple:** Un request care trece prin API, DB, queue și servicii externe.

## Log Analytics Workspace

**Rol principal:** Workspace pentru Azure Monitor Logs.

**Unde îl folosești:** KQL, retenție logs, workbook-uri, diagnostic settings, Container insights, Application Insights workspace-based.

## Azure Monitor Workspace

**Rol principal:** Workspace pentru Prometheus managed.

**Unde îl folosești:** Metrici Prometheus, PromQL, integrare cu Managed Grafana și monitoring cloud-native.

Nu confunda cele două workspace-uri! LAW = centrul pentru logs. AMW = esențial pentru ecosistemul Prometheus managed.

# Application Insights pe Scurt

Application Insights este componenta APM din Azure Monitor. Pentru multe scenarii moderne, Microsoft recomandă instrumentarea prin **OpenTelemetry** — o abordare mai portabilă și mai standard pentru telemetry de aplicație.

## → Ce colectează

Requests, dependencies, exceptions, traces, availability telemetry și metrice relevante de aplicație.

## → Ideal pentru

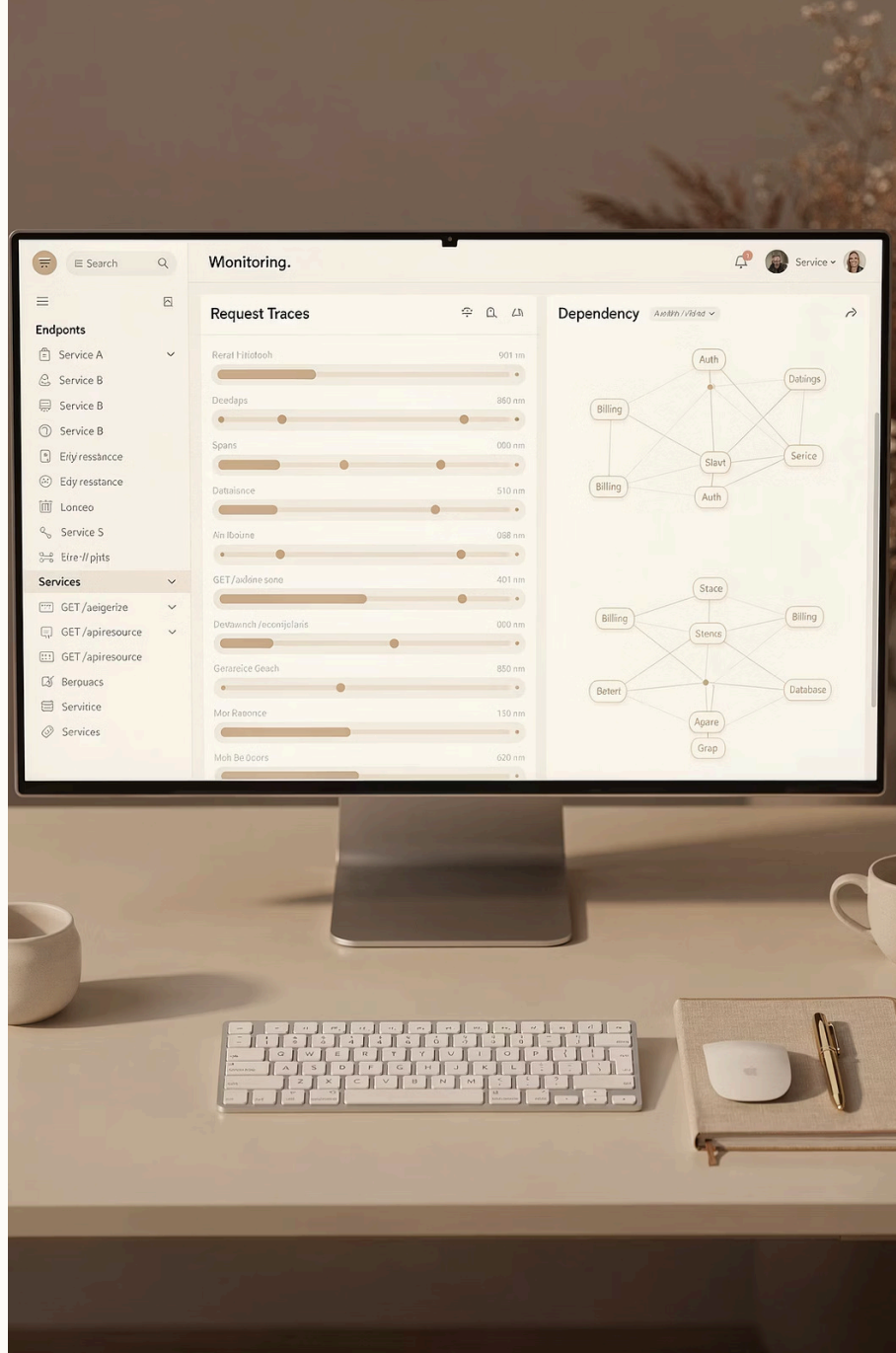
Web apps, API-uri, servicii backend și workload-uri în care contează experiența end-to-end a request-ului.

## → Control cost și volum

Poți folosi sampling și filtre pentru a controla costul și volumul de date ingerate.

## → Recomandare pentru aplicații noi

Dacă ai o aplicație nouă, încearcă să pornești cu OpenTelemetry, nu cu o strategie veche de instrumentare, dacă nu ai un motiv puternic.



# Cum Citești Portalul Azure când Lucrezi cu Azure Monitor

Un motiv pentru care Azure Monitor pare complex este că funcționalitatea este distribuită în mai multe locuri din portal. Trebuie să înveți **harta**, nu doar butoanele.

### Azure Monitor > Overview

Punctul central: alerts, metrics, logs, insights, workbooks, action groups, DCRs. Acesta este locul de start pentru orice investigație sau configurare globală.

### Resursă individuală > Monitoring

Metrice, diagnostic settings, alerts și insights direct pe resursa respectivă. Util pentru investigații punctuale pe o VM, App Service sau AKS.

### Log Analytics Workspace

Queries, tables, retention, solutions, access control. Centrul pentru KQL și investigații operaționale complexe.

### Application Insights

Failures, performance, application map, live metrics, availability. Perspectiva APM completă pentru aplicații.

### AKS > Monitoring

Container insights, Prometheus, Grafana, logs, node/workload health. Vizibilitate completă pentru workloads Kubernetes.

CAPITOL 4

# Laborator Ghidat: Construire de la Zero a unei Fundații Azure Monitor

Construim un setup didactic complet, simplu, dar suficient de realist pentru majoritatea claselor. Scopul nu este să configurăm absolut tot din prima, ci să înțelegem piesele și motivele fiecărei alegeri.

## Resurse recomandate pentru laborator

- 1 Resource Group pentru observability, separat de aplicații
- 1 Log Analytics Workspace
- 1 Application Insights workspace-based
- 1 Azure Monitor Workspace pentru Prometheus
- 1 Action Group
- 1-3 alert rules de exemplu
- 1 availability test / synthetic monitor
- 1 workbook
- Opțional: 1 Azure Managed Grafana și 1 cluster AKS

# Pașii 1–3: Resource Group, Log Analytics Workspace și Retenție

01

## Pasul 1: Creează Resource Group-ul pentru observability

Portal > Resource groups > Create. Nume exemplu: `rg-monitoring-dev-weu`. Regiune: West Europe. **De ce separat?** Un RG dedicat oferă guvernare mai bună, lifecycle mai clar, tagging mai coerent și separare mentală sănătoasă între workload și platforma de monitorizare.

02

## Pasul 2: Creează Log Analytics Workspace

Portal > Log Analytics workspaces > Create. Nume: `law-monitoring-dev-weu`. Aceeași regiune ca RG-ul. **De ce important?** LAW este "depozitul inteligent" pentru logs — aici rulezi KQL, investighezi incidente și stochezi date din diagnostic settings și Container insights.

03

## Pasul 3: Configurează retenția și cost awareness

Verifică setările de retenție după creare. **Laborator/curs:** 7-30 zile (cost mic, suficient pentru exerciții). **Producție standard:** 30-90 zile (echilibru investigații și cost). **Audit/compliance:** 90+ zile sau export (doar dacă există cerință reală).

Nu crea câte un workspace pentru fiecare resursă fără motiv. În medii mici și medii, câteva workspace-uri bine gândite sunt mai sănătoase decât zeci de workspace-uri haotice. Separă pe medii, regiuni sau echipe doar când există motiv clar: compliance, ownership, cost allocation sau limite operaționale.

# Pașii 4–6: Application Insights, Azure Monitor Workspace și Action Group

01

## Pasul 4: Creează Application Insights (workspace-based)

Portal > Application Insights > Create. Nume: `appi-demo-web-dev-weu`. Alege **Workspace-based mode** și leagă resursa de LAW-ul creat. **De ce workspace-based?** Centralizează analiza, evită silozuri și permite corelarea naturală a telemetriei de aplicație cu restul logurilor operaționale.

02

## Pasul 5: Creează Azure Monitor Workspace pentru Prometheus

Portal > Azure Monitor workspaces > Create. Nume: `amw-monitoring-dev-weu`. Aceeași regiune cu AKS dacă monitorizezi AKS prin managed Prometheus. **Regula simplă:** logs în LAW, Prometheus în AMW. Nu crea acest workspace dacă laboratorul nu include AKS sau Prometheus.

03

## Pasul 6: Creează Action Group

Azure Monitor > Alerts > Action groups > Create. Nume: `ag-ops-core`. Adaugă cel puțin o acțiune: email pentru laborator. În producție adaugi webhook, Azure Function, Logic App sau integrare ITSM. **Gândește-le ca pachete reutilizabile** — cine este notificat și ce acțiuni se întâmplă când alerta se declanșează.

# Fluxul unei Alerte Bune în Azure Monitor

Model recomandat pentru reducerea alert fatigue și accelerarea răspunsului operațional.


1

### Semnal

Metrică, log, trace, availability test, Activity Log

2

### Condiție

Threshold static/dinamic, KQL, PromQL, severitate

3

### Alert Rule

Scop, frecvență, window, split by dimension

4

### Action Group

Email, Teams via webhook, SMS, Function, ITSM

5

### Triere SRE

Acknowledge, investigație, runbook, RCA

### Severitate

Folosește Sev 0-4 cu reguli clare de business.

### Noise Control

Aplică processing rules și praguri dinamice.

### Ownership

Fiecare alertă trebuie să aibă owner și runbook.

### Learning Loop

După incident, ajustezi regula sau dashboard-ul.

CAPITOL 5

# Diagnostic Settings, DCR și Colectarea Datelor

Mulți începători configurează alerte înainte să se asigure că datele chiar ajung unde trebuie. Este o greșeală clasică. Mai întâi confirmi colectarea, apoi construiești vizualizare și alertare.

## Diagnostic Settings

Mecanismul prin care multe resurse Azure trimit logs și uneori metrics către destinații. Alege destinația în funcție de scop:

- **Log Analytics Workspace**: Investigație, workbook-uri, alerte log-based — aproape întotdeauna prima alegere
- **Storage Account**: Arhivă și retenție ieftină, bun pentru compliance
- **Event Hub**: Streaming către alte platforme, util pentru SIEM sau integrare enterprise

## Data Collection Rules (DCR)

DCR-urile definesc ce date se colectează, de unde și în ce destinație ajung. În multe scenarii portalul le creează automat, dar este important să înțelegi existența lor.

- Pentru Azure Monitor Agent, Prometheus managed și alte fluxuri moderne, DCR este piesa de control
- În scenarii enterprise, DCR devine obiect de guvernanță și versionare
- Când vrei customizare avansată, transformări sau control fin al datelor, ajungi aproape sigur la DCR

## Verificarea colectării — pași esențiali

1. Deschide resursa monitorizată și verifică secțiunea Monitoring
2. Asigură-te că diagnostic settings trimit în workspace-ul corect
3. În LAW, rulează o interogare simplă pe ultimele 30 minute
4. Abia după aceea construiește alerta

CAPITOL 6

# Alerte în Azure Monitor: Toate Opțiunile Importante

Alertele nu sunt scopul final — ele sunt mecanismul prin care observabilitatea devine acțiune. **O alertă bună** este clară, măsurabilă, ușor de triat și legată de un owner. **O alertă proastă** creează zgomot, panică sau indiferență.

| Tip alertă | Pe ce se bazează | Exemple bune |
|---|---|---|
| Metric alerts | Metrice platform, custom sau App Insights metrics | CPU, memory, response time, node pressure, availability |
| Log search alerts | Query KQL în LAW sau alte surse de logs | Erori aplicație, pattern de excepții, evenimente Kubernetes |
| Activity Log alerts | Evenimente de control plane Azure | Ștergere resource group, schimbări de role, stop/start VM |
| Service Health alerts | Starea serviciilor Azure și mentenanță planificată | Incident regional Azure, advisory, maintenance |
| Prometheus alerts | Metrice Prometheus / PromQL | Pod restarts, kube-state, latency cloud-native |

În practică, folosești mai multe tipuri simultan: metric alert pentru semnale rapide, log alert pentru context operațional, activity log alert pentru guvernare și Prometheus alert pentru Kubernetes.

# Metric Alert vs Log Alert și Dynamic vs Static Thresholds

## Metric Alert vs Log Alert

**Metric alert** — când ai un numeric bine definit și vrei reacție rapidă. Simplu, performant, bun pentru praguri și dynamic thresholds.

*Exemplu:* CPU > 80% timp de 10 minute pe o VM sau Average node CPU > prag pe AKS.

---

**Log alert** — când ai nevoie de logică bogată și context din query. Mai flexibil, dar poate fi mai costisitor și mai complex.

*Exemplu:* Număr de excepții de tip TimeoutException > 20 în 15 minute sau evenimente Kubernetes de tip Warning pe un namespace critic.

## Static vs Dynamic Thresholds

**Static threshold**: tu alegi pragul — de exemplu CPU > 80%. Consum simplu, explicabil, ușor de predat la curs. Bun când comportamentul este predictibil și ai limită clară.

**Dynamic threshold**: platforma învață comportamentul istoric și detectează anomalii. Reduce munca manuală, dar trebuie înțeles și validat. Bun când semnalul are variații sezoniere sau pattern dificil de setat manual.

> Pentru studenți: începe cu static thresholds. După ce înțeleg stabil ce măsoară, introdu dynamic thresholds ca maturizare.

## Action Groups și Alert Processing Rules

Action Group = ce se întâmplă când alerta se declanșează. Alert Processing Rule = cum modifici comportamentul: suprimi, rerutezi sau programezi excepții. Folosește Alert Processing Rules pentru ferestre de mentenanță și suppression controlat, nu pentru a ascunde probleme permanente.

# Exemplu Ghidat: Creează o Metric Alert și un Log Alert cu KQL

## Metric Alert — pași în portal

1. Intră pe o resursă (App Service, VM sau AKS cluster)
2. Monitoring > Alerts > Create > Alert rule
3. Scope: verifică resursa corectă
4. Condition: alege o metrică relevantă (Response Time, CPU Percentage sau Requests Failed)
5. Setează aggregation, operator, threshold, frequency și lookback window
6. Atașează Action Group-ul creat anterior
7. Setează severity și name clar: ex. `app-web-prod-high-latency`
8. Create

> Scope = unde observi. Condition = când e problemă. Frequency/window = sensibilitatea. Severity = prioritatea. Action Group = cine află și ce se execută.

## Log Alert cu KQL — pași în portal

1. Azure Monitor > Logs
2. Selectează workspace-ul corect
3. Testează query-ul înainte de a crea alertă
4. Când query-ul returnează exact ce vrei, alege New alert rule
5. Configurează evaluarea, threshold-ul și action group-ul

Exemplu simplu de KQL pentru excepții de aplicație:

```
exceptions
| where timestamp > ago(15m)
| summarize ExceptionCount = count() by type
| where ExceptionCount > 20
```

CAPITOL 7

# Workbooks, Dashboards și Vizibilitate Executivă

Workbooks sunt una dintre cele mai subestimate componente din Azure Monitor. Ele îți permit să combini text explicativ, metrice, query-uri, parametri și grafice într-un singur raport interactiv. **Un workbook bun nu este doar un dashboard frumos — este un instrument de decizie.**

### Cazuri de utilizare

Excelente pentru NOC, SRE, platform engineering, management reviews și workshop-uri de training. Poți construi un workbook de triere pentru incident, unul de sănătate săptămânală și unul executiv cu KPI-uri.

### Recomandări practice

- Pune sus indicatorii-cheie: availability, latency, error rate, saturation
- Folosește parametri pentru subscription, environment, region, cluster și namespace
- Separă clar vederea executivă de vederea tehnică
- Nu încerca să pui totul într-un singur workbook gigantic — mai bine 2-4 workbook-uri cu scop clar

### Exemplu de structură workbook pentru AKS

| Secțiune | Scop | Exemple de conținut |
|---|---|---|
| Secțiunea 1 | Health summary | Cluster state, alerts, node readiness, incidents active |
| Secțiunea 2 | Capacity | CPU/memory requests vs limits, node pressure, autoscaler activity |
| Secțiunea 3 | Workloads | Top namespaces, pod restarts, crash loops, deployments unhealthy |
| Secțiunea 4 | Networking | Ingress errors, DNS issues, latency, dropped packets |
| Secțiunea 5 | Cost & retention | Volum logs, top noisy tables, sampling notes |

CAPITOL 8

# Synthetic Monitoring și Availability Tests

Synthetic monitoring înseamnă să testezi proactiv aplicația chiar dacă încă nu există trafic real de utilizator. Este esențial pentru disponibilitate și degradări lente. Application Insights availability tests verifică disponibilitatea și răspunsul aplicației din mai multe puncte geografice.

### Ping / Standard Web Test

Pentru disponibilitate de bază și latență. Excelent punct de pornire pentru studenți. Validează status code și timpul de răspuns.

### Multi-step / Scenarii bogate

Pentru fluxuri funcționale unde simplul 200 OK nu este suficient. Folosește când maturitatea echipei crește.

### Synthetic Monitor pentru dependențe externe

Când vrei să verifici API-uri externe, DNS, endpoint-uri publice. Foarte util în operațiuni enterprise.

## Pași de configurare în portal

1. Deschide Application Insights
2. Selectează Availability sau Availability tests
3. Create test și completează URL-ul, locațiile, frecvența și criteriile de succes
4. Activează alertarea dacă testul eșuează sau latența depășește pragul
5. Salvează și validează după câteva cicluri

De ce este esențial în SRE? Unele probleme apar înainte ca utilizatorii să le raporteze. Dacă homepage-ul răspunde greu dintr-o regiune sau un endpoint dă intermitent 500, testele sintetice pot prinde situația foarte devreme.

CAPITOL 9

# Advanced Monitoring pentru AKS: Extensia Cloud-Native

Pentru AKS, Azure Monitor devine și mai valoros, deoarece Kubernetes introduce complexitate operațională reală: noduri, pods, namespaces, control plane, autoscaling, containers, Prometheus metrics, logs și evenimente. **În AKS nu te bazezi pe un singur tip de date.**

## Container Insights

Pentru logs stdout/stderr, evenimente și performanță de cluster și container. Oferă vizibilitate completă la nivel de workload.

## Prometheus Metrics

Pentru metrice Kubernetes native și custom exporters. Stocate în Azure Monitor Workspace, interogabile cu PromQL.

## Azure Managed Grafana

Pentru dashboard-uri bogate și interactivitate cloud-native. Integrare nativă cu AMW și Prometheus managed.

## Diagnostic Settings Control Plane

Pentru audit și troubleshooting al clusterului. Esențial pentru investigații de nivel avansat.

# Ce Urmărești în AKS și Exemple de Alertare

## Arii de monitorizare AKS

| Arie | Semnale utile |
| --- | --- |
| Disponibilitate cluster | Node Ready, API server health, workload health |
| Capacitate | CPU/memory usage, requests/limits, pending pods, autoscaler activity |
| Stabilitate workload | Restarts, CrashLoopBackOff, OOMKilled, failed probes |
| Rețea și trafic | Ingress errors, latency, DNS, connection failures |
| Cost și zgomot | Namespace-uri care produc prea multe logs sau metrice inutile |

## Exemple concrete de alertare AKS

- Node CPU sau memory pressure persistent
- Pod restart count anormal pe un namespace critic
- Număr mare de OOMKilled pe un deployment important
- Availability test eșuat pentru endpoint-ul principal expus prin ingress
- Prometheus alert pe `kube_pod_container_status_restarts_total` sau pe latency/throughput

## Exemplu PromQL orientativ

```
sum by (namespace, pod) (
 increase(
  kube_pod_container_status_restarts_total[15m]
 )
) > 3
```

Pentru începători, nu trebuie să intri adânc în PromQL din prima lecție. Dar e util să arăți că Azure Monitor poate găzdui metrice Prometheus fără să administrezi un Prometheus server tradițional.

CAPITOL 10

# Cazuri Reale de Utilizare

### 🔴 API business critic care răspunde lent

- Metric alert detectează creșterea latenței P95
- Availability test confirmă degradarea din două regiuni
- Application Insights arată dependency calls lente către baza de date
- Log Analytics confirmă spike de timeout exceptions
- Workbook-ul incidentului arată că problema a început după o schimbare de configurație

**Lecția SRE:** Nicio singură sursă nu era suficientă; puterea vine din corelare.

### 🟡 Subscription governance și schimbări neautorizate

- Activity Log alert detectează ștergerea unui NSG sau modificarea unei role importante
- Action Group trimite notificare și webhook către platform operations
- Alert processing rule evită spamul în fereastra programată de mentenanță

**Lecția SRE:** Monitorizarea nu înseamnă doar performanță, ci și control operațional și guvernare.

### 🔵 AKS cu incidente intermitente

- Container insights arată pod restarts și kube events
- Prometheus metrics indică memory saturation și request throttling
- Grafana evidențiază pattern pe namespace și workload
- Postmortem-ul duce la ajustarea requests/limits și la reguli de alertare mai bune

### 🟢 Echipa executivă cere "o singură vedere"

- Construiești workbook executiv cu availability, error rate, cost hotspots și incidente pe ultimele 30 zile
- Separi același domeniu în workbook tehnic pentru SRE și platform engineering

**Lecția SRE:** Stakeholderii au nevoie de niveluri diferite de detaliu; o singură vedere universală este rar optimă.

CAPITOL 11

# Cost Control și Bune Practici

Un Azure Monitor prost guvernat poate deveni scump și zgomotos. Un Azure Monitor bine guvernat devine o investiție clară în fiabilitate și viteză operațională.

## Bune practici esențiale

- **Colectează doar ce folosești** — nu activa toate categoriile de logs fără scop clar
- **Separă medii și ownership-ul** — evită amestecul haotic între dev/test/prod
- **Folosește sampling** — mai ales pentru Application Insights și volum mare
- **Optimizează retenția** — nu păstra luni întregi date fără valoare operațională
- **Controlează cardinalitatea** — mai ales în Prometheus și cloud-native monitoring
- **Revizuiește alertele lunar** — șterge sau ajustează alertele zgomotoase

## Anti-pattern-uri frecvente

- Workspace nou pentru fiecare resursă
- Alertă pentru orice mic spike fără context
- Severity mare pentru evenimente minore
- Diagnostic settings activate "ca să fie", fără owner
- Dashboard frumos, dar fără runbook și fără accountability
- AKS monitorizat doar la nivel de nod, nu și la nivel de workload

## Principiu simplu de maturitate

Începi cu **vizibilitate minimă sănătoasă**, nu cu perfecțiune. Apoi crești maturitatea în pași: semnale de bază → dashboard-uri → alerte curate → synthetic monitoring → corelare aplicație-infrastructură → automatizare și SLO-driven operations.

CAPITOL 12

# Model de Implementare Recomandat pentru Curs

Această progresie funcționează foarte bine pentru studenți. Nu îi sufoci de la început cu tot portofoliul, dar nici nu simplifici excesiv până la caricatură.

## Nivel 1 — Fundamentals

Overview Azure Monitor, metrics vs logs, LAW, o alertă simplă, un workbook de bază.

## Nivel 2 — Operations

Action groups, log alerts, Service Health, Activity Log, diagnostic settings.

## Nivel 3 — Application Observability

Application Insights, exceptions, traces, availability tests.

## Nivel 4 — Cloud-Native

AKS monitoring, Prometheus, Grafana, Container insights.

## Nivel 5 — SRE Practice

SLI/SLO, error budgets, alert tuning, postmortem și toil reduction.

CAPITOL 13

# Checklist de Implementare Minimă pentru Orice Organizație

1. ## Definește ce este critic

   Aplicații, endpoint-uri, procese și active de business. Fără această claritate, totul pare la fel de important.

2. ## Stabilește 3-5 SLI-uri principale

   Pentru fiecare serviciu important. Acestea devin baza pentru SLO-uri și error budgets.

3. ## Creează cel puțin un Log Analytics Workspace

   Cu un model clar de ownership. Configurează diagnostic settings pe resursele-cheie.

4. ## Activează Application Insights

   Pentru aplicațiile importante. Configurează Action Groups și cel puțin câteva alerte utile, nu zeci de alerte zgomotoase.

5. ## Adaugă availability tests și workbook-uri

   Pentru punctele publice critice. Construiește 1-2 workbook-uri cu adevărat utile.

6. ## Dacă folosești AKS

   Activează Container insights și Prometheus managed. Planifică revizuirea lunară a alertelor, costului și retenției.

CAPITOL 14

# Recomandări Finale pentru Studenți

### Înțelege fluxul, nu meniurile

Când înveți Azure Monitor, nu memora meniuri. Înțelege fluxul: **sursă de date → stocare → analiză → alertare → răspuns**.

### Pornește cu un scenariu real

Nu porni cu totul dintr-o dată. Începe cu un scenariu real: o aplicație, o resursă, un endpoint critic.

### Justifică fiecare alertă

Dacă nu poți explica de ce există o alertă, probabil nu ar trebui să existe. Dacă nu știi unde ajung datele, nu ești încă în control.

### SRE matur înseamnă mai mult decât dashboard-uri

Dacă ai numai dashboard-uri, dar fără runbook-uri și ownership, nu faci încă SRE matur. Pentru AKS, combină întotdeauna perspectivele de infrastructură, Kubernetes și aplicație.

> Cel mai bun mod de a preda și învăța Azure Monitor este să pornești de la incidente credibile: latență mare, excepții, resource deletion accidental, pod restarts, certificat expirat, endpoint indisponibil. Acolo studentul înțelege imediat de ce observabilitatea contează.

ANEXA A

# Exemple Utile de KQL pentru Laborator

Aceste query-uri sunt puncte de pornire practice pentru investigații și alerte în Log Analytics Workspace.

## Heartbeat — agenți și mașini care raportează

```
Heartbeat
| where TimeGenerated > ago(15m)
| summarize LastSeen=max(TimeGenerated) by Computer
```

## Erori de aplicație

```
AppExceptions
| where TimeGenerated > ago(30m)
| summarize Errors=count() by ProblemId
```

## Request-uri lente

```
AppRequests
| where TimeGenerated > ago(30m)
| summarize AvgDuration=avg(DurationMs)
   by bin(TimeGenerated, 5m)
```

## Kubernetes warnings

```
KubeEvents
| where TimeGenerated > ago(30m)
| where Type == 'Warning'
```

## Top tabele cu volum

```
Usage
| where TimeGenerated > ago(1d)
| summarize QuantityGB=sum(Quantity)/1024
   by DataType
| sort by QuantityGB desc
```

Testează întotdeauna query-ul în LAW înainte de a crea o alertă bazată pe el. Asigură-te că returnează exact ce vrei și că tabelul există și are date.

ANEXA B

# Întrebări Bune de Pus într-un Workshop SRE

Aceste întrebări ajută echipele să identifice lacunele de observabilitate și să prioritizeze eforturile de monitorizare.

> Care este cel mai important serviciu digital pentru business și cum îl măsurăm?

> Ce alertă ne trezește noaptea și chiar merită asta?

> Ce semnal avem astăzi, dar nu știm cine îl deține?

> Putem vedea rapid diferența dintre incident de aplicație și incident de platformă?

> Ce date colectăm degeaba și ne costă inutil?

> Ce endpoint public ar trebui monitorizat sintetic chiar azi?

> În AKS, putem distinge între problemă de nod, de workload și de aplicație?

ANEXA C

# Resurse Oficiale Microsoft Learn Recomandate

Interfața exactă din portal poate evolua. Pentru materiale de curs, menține principiile și structura mentală, apoi validează pașii finali în tenantul și regiunea în care predai efectiv.

### Fundamente Azure Monitor

Azure Monitor overview · Azure Monitor data platform · Azure Monitor Logs overview · Types of Azure Monitor alerts

### Alerte și Action Groups

Action groups in Azure Monitor · Alert processing rules · Application Insights overview și availability tests

### Vizualizare și Colectare

Azure Workbooks overview · Data collection rules (DCRs) in Azure Monitor

### AKS și Kubernetes

Monitor Azure Kubernetes Service (AKS) · Best practices for monitoring Kubernetes with Azure Monitor

Toate resursele sunt disponibile gratuit pe **learn.microsoft.com**. Recomandăm parcurgerea lor în paralel cu laboratoarele practice pentru o înțelegere completă.