



UNIVERZITET U NOVOM SADU  
FAKULTET TEHNIČKIH NAUKA  
KATEDRA ZA PRIMENJENE RAČUNARSKE NAUKE

# Računarstvo u oblaku

ms Helena Anišić

Zimski semester 2025/2026.

Studijski program: Računarstvo i automatika

Modul: Računarstvo visokih performansi



**Microsoft Azure**

# Azure

- **Microsoft Azure** je cloud platforma, skup usluga za izgradnju, pokretanje i upravljanje aplikacijama i infrastrukturoom preko interneta.
- Umesto da kupujete i održavate sopstvene servere, koristite **resurse na zahtev**:
  - računanje (VM, kontejneri),
  - mreže (VNet, Load Balancer),
  - baze podataka,
  - alati za bezbednost, nadzor i automatizaciju.

# Azure

## Model plaćanja

- Azure je uglavnom **pay-as-you-go**: plaćaš ono što koristiš.
- Najskuplji deo je obično **compute** (VM, AKS nodovi, duže pokrenuti servisi).
- Neki resursi imaju trošak po:
  - vremenu rada (npr. compute),
  - potrošnji (npr. storage kapacitet),
  - broju operacija (npr. određene managed usluge).

## Gde se vidi potrošnja

- Azure Portal → **Cost Management + Billing**

# Azure

U Azure-u, **resurs** je pojedinačna “stavka” koju korisnik kreira i kojom Azure upravlja, ima svoje ime, lokaciju, konfiguraciju, dozvole i često cenu.

## Šta resursi pokrivaju (kategorije)

- **Compute:** VM, AKS, ACI, App Service, Functions
- **Storage:** Storage Account (Blob/File/Queue), Managed disks
- **Networking:** VNet, Subnet, NSG, Load Balancer, Public IP, DNS
- **Databases:** Azure SQL, Cosmos DB, PostgreSQL/MySQL managed
- **Security/Identity:** Key Vault, Entra ID integracije, policy, Defender
- **Observability:** Log Analytics workspace, Application Insights, Monitor alerts

# Resursi

Resurs je nešto što:

- se kreira (Container Registry, Container Instance, itd.),
- ima **tip** (npr. `Microsoft.ContainerRegistry/registries`),
- pripada nekom **Resource Group-u**,
- nalazi se u određenom **regionu** (ili je globalan),
- ima **Access control (IAM)**, RBAC pravila,
- ima životni ciklus: create, update, delete,
- može da ima **zavisnosti** od drugih resursa (npr. AKS koristi VNet).

# Tip resursa

U Azure-u, **tip resursa** je “klasa” ili **kategorija** resursa, tj. *šta je taj resurs po definiciji* i kojim provajderom se upravlja.

Azure to formalno zapisuje kao:

```
<Resource Provider>/<Resource Type>
```

Primeri:

- **Microsoft.ContainerRegistry/registries** → Azure Container Registry (ACR)
- **Microsoft.ContainerInstance/containerGroups** → Azure Container Instances (ACI container group)
- **MongoDB.Atlas/organizations** → MongoDB Atlas (Azure Native Integration), organizacija kao Azure resurs

# Resource Group

**Resource Group** je logički “kontejner” u Azure-u u koji se smeštaju povezani resursi

## Zašto postoji

- lakše upravljanje: resursi se grupišu po projektu, predmetu, okruženju (*dev/test/prod*), timu,
- lakša kontrola dozvola (RBAC) i politika (Policy) na nivou grupe,
- lakše praćenje troškova (cost management, tags).

## Životni ciklus

- kreira se RG, zatim se kreiraju resursi “unutra”,
- kad se obriše RG, Azure briše i resurse u njemu

# Resource group

`RG=cc-rg-yourid`

`LOC=germanywestcentral`

`az group create -n $RG -l $LOC`

## Napomena

- RG ima svoju lokaciju, ali resursi u RG-u mogu ponekad biti u drugim lokacijama, zavisi od tipa resursa; za vežbe držimo sve u istom regionu radi jednostavnosti.

# Regioni

## Azure regioni (Regions)

- **Region** je geografska lokacija (Azure data centar/klaster data centara) gde se resurs fizički hostuje.  
Primeri: germanywestcentral, northeurope, swedencentral.
- **Zašto je region važan**
  - **Latencija**: bliži region, manja latencija prema korisnicima i servisima.
  - **Dostupnost servisa**: nisu svi Azure servisi i SKU-ovi dostupni u svim regionima.
  - **Compliance i politika**: organizacije i studentske pretplate često imaju **Azure Policy** koja dozvoljava samo određene regione.
  - **Trošak**: cene mogu da se razlikuju po regionima.

# Regioni

- Na vežbama izaberemo **jedan dozvoljeni region** i sve resurse držimo tamo (npr. `germanywestcentral`) radi jednostavnosti.
- Ako se pojavi greška “disallowed by policy”, potrebno je promeniti region na neki koji je dozvoljen.

# ACR (Azure Container Registry)

- **Container registry** je mesto gde se čuvaju **container image-i**, odakle ih runtime (ACI/AKS) i CI/CD povlače.
- **ACR** je registry unutar Azure pretplate, tipično privatni i namenjen deployment-u u Azure-u.
  - **Docker Hub** je javni, opšti registry, čest za base image-e i deljenje open-source image-a.
- Naredba za postavljanje container image-a je ista kao za Docker Hub: `docker push`

# ACR VS. Docker Hub

- **Identitet i prava**

- Docker Hub: pristup se kontroliše Docker nalozima, timovima, tokenima.
- ACR: koristi **Entra ID (Azure Active Directory) + Azure RBAC**, i može se dati tačno pravo tipa **AcrPull/AcrPush** na tačno jedan registry.

- **AKS integracija**

- ACR se može direktno “povezati” sa AKS (**-attach-acr**), pa klaster povlači privatne image-e bez da se ručno ubacuju Docker kredencijali/tajne u Kubernetes.

# ACR VS. Docker Hub

- **CI/CD bez dugoročnih tajni**
  - U Azure svetu često koristiš OIDC/managed identity pristup, pa pipeline ne mora da čuva registry password.
  - Sa Docker Hub privatnim repo-om se obično koristi username/token kao tajna.
- **Policy i governance**
  - ACR je “u tvojoj pretplati”, pa može da podleže Azure Policy, tagging pravilima, cost tracking-u, audit-u.
  - Docker Hub je eksterni servis van Azure governance okvira.

## Push a container image



Complete the steps below to get started using Container Registry. [View documentation](#)

### 1 Install Docker

Before you can try out the Azure Container Registry, you should install Docker. Refer to the [Mac OS](#), [Windows](#), or [Linux](#) getting started instructions for Docker.

### 2 Log in to your registry

Have [Azure CLI](#) and [Docker CLI](#) installed locally on your host machine. Docker provides packages that easily configure Docker on any [Mac OS](#), [Windows](#), or [Linux](#) system.

```
az login  
az acr login --name vezbe10
```



### 3 Pull and tag your image

For this example, you will pull a public "hello world" image locally and provide a tag before you push the image to your registry.

```
docker pull mcr.microsoft.com/mcr/hello-world  
docker tag mcr.microsoft.com/mcr/hello-world vezbe10-  
gehqxgtbggybsea.azurecr.io/samples/hello-world
```



### 4 Push the image to your registry

Now that you've tagged the "hello world" image with the fully qualified path to your private registry, you can push it to the registry with [docker push](#)

```
docker push vezbe10-gehqxgtbggybsea.azurecr.io/samples/hello-world
```



# Azure Container Instances (ACI)

- **Šta je ACI**  
Servis za pokretanje kontejnera u Azure-u “na zahtev”, bez upravljanja VM-ovima i bez Kubernetes-a. Korisnik definiše image, CPU/RAM, portove i environment varijable, Azure pokrene kontejner.
- **Osnovni koncept: Container Group**  
ACI resurs je **container group**:
  - može sadržati jedan ili više kontejnera (npr. app + sidecar),
  - svi kontejneri dele isti network namespace (IP/portovi) i mogu deliti volume.
- **Kada se koristi**
  - brzi demo “pokretanje kontejnera u cloudu”
  - batch job / jednokratni task
  - testiranje image-a iz ACR pre AKS deployment-a
  - jednostavan HTTP servis sa public IP

# FQDN

FQDN znači **Fully Qualified Domain Name**.

To je **cela DNS adresa** nekog hosta, koja uključuje:

- naziv računara/servisa (host),
- sve nivoe domena,
- i završava se TLD-om (npr. `.com`, `.io`, `.rs`).

Primer:

- `myapp.eastus.azurecontainer.io` je FQDN
  - `myapp` je host (ime servisa)
  - `eastus.azurecontainer.io` je domen
- FQDN se koristi za pristup servisu preko imena (DNS), umesto preko IP adrese.

## Primer sa nginx (ACI)

```
az container create --resource-group cc-26 --file azureacideploy.yaml
```

# Azure Storage Account

- Osnovni Azure resurs za skladištenje podataka, "krovni" servis koji u sebi sadrži više tipova storage-a
- Tipovi podataka:
  - **Blob Storage**: fajlovi, objekti, slike, backup, logovi, data lake scenariji
  - **File Shares**: SMB fajl šerovi
  - **Queue Storage**: jednostavne poruke za asinhronu obradu
  - **Table Storage**: NoSQL key value (jednostavniji scenariji)
- Bezbednost i pristup:
  - Access keys, SAS tokeni, Azure AD, Managed Identity
  - Enkripcija podataka u mirovanju, opcije za private access (VNet, private endpoint)
- Tipične upotrebe: statički sadržaj, upload/download fajlova, deljeni fajl sistem, poruke između servisa, logovi i backup.

# SMB file share

SMB file share je **deljeni folder na mreži** koji koristi protokol **SMB (Smb, Server Message Block**, poznat kao Windows file sharing).

- “Mrežni disk” kome pristupa više korisnika ili aplikacija.
- Fajlovi izgledaju kao da su na lokalnom disku, ali su zapravo na serveru.
- **Azure Files** pruža **SMB file share** u okviru Storage Account-a.

Kada se koristi:

- aplikacije koje očekuju klasični fajl sistem (npr. upload folder, shared config),
- više instanci aplikacije koje treba da dele iste fajlove.

Ograničenja u odnosu na Blob:

- SMB je “fajl sistem” interfejs; Blob je objektno skladište (bolje za velike objekte i web sadržaj).

# Blob Storage

## Šta je Blob

- **Blob (Binary Large Object)** je **objektno skladište**.
- Blobovi se čuvaju u **container**-ima.
- Ime bloba može da izgleda kao putanja (`images/2026/a.png`), ali to je samo **string u nazivu**, nije pravi direktorijum.
- Ne može se “normalno” koristiti kao shared folder za aplikacije koje očekuju disk (SMB).
- Operacije su tipično: upload/download/list preko HTTP(S) API-ja ili SDK-a.

## Kada se Blob koristi

- slike, video, backup, logovi, statički web sadržaj,
- veliki fajlovi i jeftino skladištenje,
- integracija sa CDN-om i web aplikacijama.

# Storage account

Home > Storage center | Blob Storage > fajlovi

**Storage center | Blob Storage** <<

Univerzitet u Novom Sadu (uns.ac.rs)

Search

Summary **Resources**

+ Create ↻ Restore ...

You are viewing a new version of Browse experience. [Click here to access the old experience.](#)

Name 1
catalogweb31447
fajlovi
fec665d50867b4305bc144a

- Overview
- All storage resources
- Object storage
  - Blob Storage**
- File storage
  - Azure Files
  - File Sync
  - NetApp Files
  - Managed Lustrre
- Block storage
- Data management

**fajlovi | File shares** ...

Storage account

Search

+ File share ↻ Refresh Give feedback

**File share settings**

Identity-based access: [Not configured](#) Default share-level permissions: [Disabled](#) Soft delete: [14 days](#) Maximum capacity: [100 TiB](#)

Security: [Maximum compatibility](#)

Search file shares by prefix (case-sensitive)   Show deleted shares

Name	Modified	Access tier	Quota
nginx-store	1/12/2026, 6:22:08 PM	Transaction optimized	100 TiB

- Overview
- Activity log
- Tags
- Diagnose and solve problems
- Access Control (IAM)
- Data migration
- Events
- Storage browser
- Storage Mover
- Partner solutions
- Resource visualizer

# Azure Container Apps (ACA)

- Managed, serverless platforma za pokretanje kontejnera u Azure-u.
- Fokus na aplikacije, ne na infrastrukturu, nema upravljanja Kubernetes klasterom.
- Radi na Kubernetes-u u pozadini.
- Pogodno za mikroservisne arhitekture i event driven servise

## Karakteristike:

- Scale to zero, idealno za servise sa povremenim saobraćajem
  - Autoscaling, skaliranje od 0 do N instanci na osnovu HTTP zahteva ili eventova
- Skaliranje po metrikama
- Plaćanje prema potrošnji resursa, CPU i memorija po instanci.
- Podrška za background worker-e, cron job-ove i event processing (npr. poruke iz queue-a)

# Azure Container Apps (ACA)

- Container Apps Environment kao logički “prostor” za više aplikacija
- Interna komunikacija između aplikacija u istom okruženju, privatni endpoint-i
- VNet integracija (kada treba privatna mreža), kontrola ulaza, policy-ji
- Managed Identity za pristup Azure resursima bez čuvanja lozinki, Key Vault za tajne
- Ingress i routing, automatski HTTPS endpoint (FQDN), opciono interno (bez javnog pristupa)
- Revisions, verzionisanje deploy-a, brzo vraćanje, kontrola saobraćaja po verzijama (traffic split)

# Kreiranje environmenta

Da bi se kreirao ACA, mora prvo se da se napravi environment.

- Ukoliko se ACA kreira preko Azure Portala, environment se kreira automatski.

`RG=cc-26`

`LOC=germanywestcentral`

`ENV=my-env`

```
az containerapp env create -g $RG -n $ENV -l $LOC
```

# Računarstvo u oblaku 2025/2026.

Enable ingress for applications that need an HTTP or TCP endpoint.

Ingress ⓘ



Ingress traffic



Limited to Container Apps Environment

Select this option if you want to restrict traffic to this container app from within the Container App Environment



Accepting traffic from anywhere

Select this option if you want to allow traffic to this container app from anywhere

Ingress type ⓘ



HTTP



TCP

Client certificate mode ⓘ



Ignore



Accept



Require

Transport

Auto 

Insecure connections



Target port ⓘ

0

Endpoint(s)

<https://primer2.jollyplant-49d48b79.germanywestcentral.azurecontainerapps.io>

Session affinity ⓘ



---

^ Additional TCP ports

---

## Skaliranje na 0 (ACA)

- Container Apps automatski menja broj replika na osnovu “trigera” (najčešće HTTP saobraćaj, konkurentnost), i granica minReplicas / maxReplicas.
- Kada je minReplicas = 0 i nema saobraćaja, aplikacija se može ugasiti, broj replika postaje 0. Sledeći zahtev je “budi” (cold start), pa prvi odgovor kasni nekoliko sekundi.
- Za baze to uglavnom nije dobro: DB treba da bude stalno dostupna.
- Ako je baza na 0 replika, nema procesa koji sluša port, backend dobija timeout i ne može da se poveže.
- Praksa: **DB minReplicas=1**, aplikacije mogu minReplicas=0 ako je prihvatljiv cold start.

# Skladišta u ACA

## Ephemeral volume (replica scoped)

- To je **lokalni, privremeni storage** vezan za **jednu repliku**.
- Podaci traju dok replika postoji, nisu namenjeni za dugotrajno čuvanje, niti za deljenje između više replika.
- Konfiguriraju se kada se postavi **novi revision**, u "Ephemeral storage" delu.
- Kada se koristi: cache, temp fajlovi, build artefakti, kratkotrajna obrada.

## Azure Files volume (persistent, shared)

ACA može da montira **Azure Files share** kao volume, i to:

- **SMB** ili **NFS** protokol.
- Fajlovi su **persistirani** u file share-u.
- Isti share može da montira **više kontejnera**, čak i u **drugoj replici, revision-u, ili drugoj container app**; svi vide iste fajlove.
- U **Container Apps environment** se dodaje "storage definition" (poveže se storage account + file share).
- U konkretnom **revision-u** se dodaju volumes, pa u kontejneru volumeMounts (mount path, opcionalno subPath)

## Skladišta u ACA - DB

MongoDB koristi storage engine **WiredTiger** koji očekuje “lokalni disk” sa punim Linux (POSIX) ponašanjem.

**Azure Files (SMB share)** nije isti kao lokalni disk, ima drugačija pravila za dozvole, zaključavanje fajlova i neke “atomic” operacije.

Zato WiredTiger pri startu ne može da otvori/zaključa ključne fajlove u `/data/db` i dobija grešku “**Operation not permitted**”, pa Mongo prekida rad.

Zaključak: Azure Files je ok za **fajlove/logove**, ali za DB je bolje:

- **managed MongoDB** (Atlas ili Mongo kompatibilan servis), ili
- **AKS + Azure Disk** (block storage) ako DB mora u klaster.

# Kreiranje automatskog deploymenta

**backend** | Deployment ☆ ...  
Container App

Search ◊ << Refresh Send us your feedback

Overview  
Activity log  
Access control (IAM)  
Tags  
Diagnose and solve problems  
Resource visualizer  
Application  
Revisions and replicas  
Containers  
Scale  
Volumes  
Settings  
Deployment

**Continuous deployment** Artifact (preview)

① You're currently deploying to your app automatically using GitHub Actions. If you would like to disconnect continuous deployment.

Set up GitHub Actions to automatically build and deploy your code to your Container App a new revision.

**GitHub settings**

Signed in as	cloud-ftn
Organization	cloud-ftn
Repository	fullstack-backend
Branch	main
Workflow file	<a href="#">.github/workflows</a>

# ACA vs ACI

## 1) Namena

- **ACI**: “pokreni kontejner”, jednostavno i brzo, često za kratke job-ove, test, batch.
- **ACA**: platforma za aplikacije, microservice-i i servisi koji treba da rade stabilno u produkciji, **bez upravljanja Kubernetes-om**.

## 2) Skaliranje

- **ACI**: nema “scale to zero” i autoscaling po eventovima u smislu platforme; skaliranje se radi ručno
- **ACA**: autoscaling od **0 do N**, po HTTP saobraćaju ili eventovima, idealno za promenljiv load.

## 3) Ingress i endpoint

- **ACI**: može public IP/DNS label, ali nema napredan routing, revizije, traffic split.
- **ACA**: ugrađen ingress, HTTPS endpoint, interna komunikacija u okruženju, lakše izlaganje servisa.

# ACA vs ACI

## 4) Deploy i verzije

- **ACI:** pokrene image; update je praktično “zameni”.
- **ACA:** **revisions**, rollback, i podela saobraćaja između verzija (npr. 90/10).

## 5) Umrežavanje

- **ACI:** container group, kontejneri u istoj grupi dele mrežu; između grupa treba VNet ili public.
- **ACA:** “environment” za više aplikacija, servisna komunikacija i mrežni model pogodniji za skup aplikacija.

# Podešavanje pravila rutiranja

Komanda za definisanje jednog zajedničkog “entrypoint” (FQDN) na nivou Container Apps environment-a i pravila rutiranja (path-based routing) ka više container app-ova.

```
az containerapp env http-route-config create \  
  
  --http-route-config-name $ROUTES \  
  
  --resource-group $RG \  
  
  --name $ENV \  
  
  --yaml routing.yaml \  
  
  --query properties.fqdn -o tsv
```

Kao rezultat dobije se **jedan FQDN** (iz `--query properties.fqdn -o tsv`) tipa:

- web.<hash>.<region>.azurecontainerapps.io

# Azure Kubernetes Service (AKS)

**Šta je:** Azure-ov upravljani Kubernetes servis, Microsoft vodi control plane; korisnik upravlja aplikacijama i node pool-ovima.

**Zašto se koristi:** standardni Kubernetes API, lakše skaliranje i održavanje klastera, integracija sa Azure servisima.

**Ključne komponente:**

- **Control plane** (managed),
- **Node pools** (VMSS, Linux/Windows),
- **Pods/Deployments/Services/Ingress** za aplikacije.

**Tipični use-case:** mikroservisi, API-jevi, event-driven servisi, batch poslovi, ML serving, multi-tenant okruženja.

**Skaliranje i dostupnost:** autoscaling node pool-a i pod-ova (HPA), zone-aware deployment, rolling update.

**Bezbednost i integracije:** Azure AD/RBAC, Network Policies, private cluster, Key Vault (CSI driver), Container Registry (ACR), monitoring (Azure Monitor).

# Odabir tipa node-a (region: italnorth)

## Node pools

Node pools	2 node pools
Kubernetes versions	1.33.5
Node sizes	Standard_D4s_v3, Standard_B2as_v2
Node auto-provisioning	Not enabled

# Azure Key Vault

- **Šta je:** Azure servis za bezbedno čuvanje i upravljanje **tajnama (secrets)**, **ključevima (keys)** i **sertifikatima (certificates)**, van koda i van image-a.
- **Zašto se koristi:** centralizuje tajne, smanjuje rizik curenja (nema hardkodovanih lozinki u repo-u, YAML-u, CI logovima).
- **Tipični primeri:** connection string za bazu, API ključevi, JWT signing key, TLS sertifikati, lozinke za servise.
- **Kada se koristi:** čim postoji više okruženja (dev, test, prod), CI/CD, više servisa, ...
- **Kako aplikacije pristupaju:** preko **Managed Identity / Workload Identity** (bez korisničkih lozinki), i autorizacije kroz **RBAC ili Access Policies**.
- **U AKS-u:** preko **Secrets Store CSI Driver** (montiranje tajni kao fajlova u pod)

# Azure Key Vault

Azure-u ima **separacija između Management Plane i Data Plane** permisija.

Kada se kreira Key Vault, automatski se dobije **Owner** role što dozvoljava da se **upravlja samim Key Vault-om** (create, delete, configure).

- Ne dobije se automatski pristup **sadržaju** Key Vault-a (read/write secrets).

Da bi se se dobio pristup Data Plane, mora se eksplicitno dodeliti **Key Vault Administrator** rola.

Ovo je Azure **security best practice** - 'least privilege by default'.

Omogućava fine-grained delegacije gde admini mogu kreirati resurse ali ne nužno pristupati senzitivnim podacima."

# Omogućavanje upotrebe key-vault u okviru AKS

Omogućava key-vault dodatak.

- Instalira CSI driver.
- Secrets Provider aktiviran.

```
az aks enable-addons \  
--addons azure-keyvault-secrets-provider \  
--name <naziv_klastera> \  
--resource-group <naziv_rg> \  
--enable-secret-rotation
```

## Dodela pristupa key-vault-u

```
AKS_IDENTITY=$(az aks show \  
--name <naziv_klastera> \  
--resource-group <naziv_rg> \  
--query addonProfiles.azureKeyvaultSecretsProvider.identity.clientId -o tsv)
```

```
az role assignment create \  
--role "Key Vault Secrets User" \  
--assignee $AKS_IDENTITY \  
--scope $(az keyvault show --name $KEY_VAULT_NAME --query id -o tsv)
```

# Managed Identity

**Managed Identity** je Azure-ov način da eliminiše potrebu za kredencijalima.

- Umesto da se čuvaju database password-i i API ključevi u kodu ili environment varijablama, **Azure automatski dodeljuje identitet** AKS cluster-u.
- Taj identitet koristi **Azure Active Directory** za autentifikaciju prema drugim Azure resursima kao što su Key Vault, Storage, ili ACR.

## **Prednosti:**

- Nula kredencijala u kodu
- Automatsko upravljanje
- Audit trag kroz Azure AD
- Enterprise-grade security

# Managed Identity

Nakon kreiranja AKS klastera, Managed Identity postoji, ali nema permisije ni za šta.

Primer za davanje permisije AKS klasteru da povuče slike iz određenog ACR-a.

```
az aks update \ --name $AKS_CLUSTER_NAME \  
  
--resource-group $RESOURCE_GROUP \  
  
--attach-acr $ACR_NAME
```

**Bez ovog koraka:** Pod-ovi ne mogu pull-ovati slike!

# Azure Identity pojmovi

## TENANT ID

- ID tvoje organizacije u Azure Active Directory
- Isti za sve korisnike i resurse u firmi
- Primer: Fakultet ima jedan Tenant ID za sve

## CLIENT ID

- ID specifične aplikacije ili managed identity
- Svaka aplikacija ima svoj jedinstveni Client ID
- Koristi se za autentifikaciju

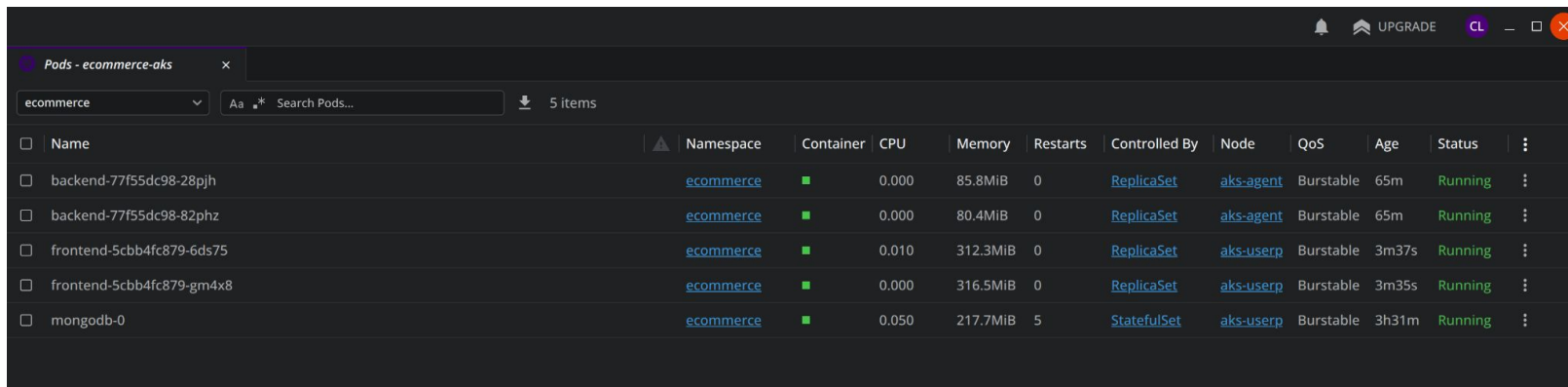
## PRINCIPAL ID (Object ID)

- Interni ID za dodelu dozvola (RBAC)
- Koristi se za autorizaciju

## MANAGED IDENTITY

- Nema email ni lozinku
- Ima svoj Client ID i Principal ID
- Omogućava sigurnu komunikaciju između servisa bez hardkodiranih lozinki

# Lens - <https://lenshq.io/>



The screenshot displays the Lens Kubernetes dashboard interface. At the top, there's a navigation bar with a bell icon, a home icon, the text 'UPGRADE', and a 'CL' logo. Below this, a tab labeled 'Pods - ecommerce-aks' is active. A search bar contains 'ecommerce' and a search icon. To the right of the search bar, it says '5 items'. The main area is a table listing pods with columns for Name, Namespace, Container, CPU, Memory, Restarts, Controlled By, Node, QoS, Age, Status, and a menu icon.

<input type="checkbox"/>	Name	Namespace	Container	CPU	Memory	Restarts	Controlled By	Node	QoS	Age	Status	
<input type="checkbox"/>	backend-77f55dc98-28pjh	<a href="#">ecommerce</a>	■	0.000	85.8MiB	0	<a href="#">ReplicaSet</a>	<a href="#">aks-agent</a>	Burstable	65m	Running	⋮
<input type="checkbox"/>	backend-77f55dc98-82phz	<a href="#">ecommerce</a>	■	0.000	80.4MiB	0	<a href="#">ReplicaSet</a>	<a href="#">aks-agent</a>	Burstable	65m	Running	⋮
<input type="checkbox"/>	frontend-5cbb4fc879-6ds75	<a href="#">ecommerce</a>	■	0.010	312.3MiB	0	<a href="#">ReplicaSet</a>	<a href="#">aks-userp</a>	Burstable	3m37s	Running	⋮
<input type="checkbox"/>	frontend-5cbb4fc879-gm4x8	<a href="#">ecommerce</a>	■	0.000	316.5MiB	0	<a href="#">ReplicaSet</a>	<a href="#">aks-userp</a>	Burstable	3m35s	Running	⋮
<input type="checkbox"/>	mongodb-0	<a href="#">ecommerce</a>	■	0.050	217.7MiB	5	<a href="#">StatefulSet</a>	<a href="#">aks-userp</a>	Burstable	3h31m	Running	⋮