|  |  |  |
| --- | --- | --- |
|  | **Univerzitet u Nišu, Elektronski fakultet**  **Katedra za računarstvo** |  |

**Seminarski rad**

**High availability rešenja kod InfluxDB baze podataka**

Darko Jotev br. ind. 1727

**Predmet: Sistemi za upravljanje bazama podataka**

**Mentor: Doc. dr Aleksandar Stanimirović**

**Niš, 2025**

**High availability rešenja kod InfluxDB baze podataka**

**Darko Jotev br. ind. 1727**

**Predmet:**

Sistemi za upravljanje bazama podataka

**Mentor:**

Doc. dr Aleksandar Stanimirović

**Sadržaj**

[1. Uvod 4](#_Toc209073881)

[1.1. Opis teme i cilj seminarskog rada 4](#_Toc209073882)

[2. Teorijska osnova 4](#_Toc209073883)

[2.1. Istorijat InfluxDB 4](#_Toc209073884)

[2.2. InfluxDB Enterprise 5](#_Toc209073885)

[2.2.1. Glavne karakteristike Core izdanja uključuju: 6](#_Toc209073886)

[2.2.2. Instalacija InfluxDB 3 Enterprise 6](#_Toc209073887)

[2.2.3. Model podataka 7](#_Toc209073888)

[2.2.4. Alati za korišćenje 8](#_Toc209073889)

[2.2.5. Podešavanje InfluxDB 3 Enterprise 9](#_Toc209073890)

[2.2.6. Podešavanje autorizacije 12](#_Toc209073891)

[2.2.7. Korišćenje MinIO kao object store za InfluxDB 3 Enterprise 13](#_Toc209073892)

[2.2.9. Primeri konfiguracije klastera 17](#_Toc209073893)

[2.2.9.1. High availability klaster 17](#_Toc209073894)

[2.2.9.2. High availability sa dedicated Compactor čvorom 18](#_Toc209073895)

[2.2.9.3. High availability sa read replicas i dedicated Compactor čvorom 20](#_Toc209073896)

[2.2.10. Upisivanje i izvršavanje upita u multi-node klasterima 23](#_Toc209073897)

[2.2.11. Upis podataka u InfluxDB 3 Enterprise 23](#_Toc209073898)

[2.2.12. Demonstracija High availability sa dedicated Compactor čvorom 26](#_Toc209073899)

[5. Zaključak 27](#_Toc209073900)

[6. Literatura 28](#_Toc209073901)

# Uvod

## Opis teme i cilj seminarskog rada

Tema ovog seminarskog rada odnosi se na problematiku postizanja visoke dostupnosti (High Availability – HA) u radu sa InfluxDB bazom podataka. InfluxDB predstavlja specijalizovanu bazu za vremenske serije podataka, koja se koristi u sistemima gde je neophodno kontinuirano prikupljanje, čuvanje i obrada velikih količina podataka u realnom vremenu. S obzirom na to da se ovi sistemi često koriste u kritičnim oblastima, kao što su nadzor IoT uređaja, mrežna i serverska infrastruktura, industrijski procesi ili finansijski tokovi, obezbeđivanje neprekidnog rada i pouzdanog pristupa podacima postaje od suštinskog značaja.

Visoka dostupnost podrazumeva projektovanje sistema na način da on nastavi sa radom i u situacijama kada dođe do otkaza pojedinih komponenti. U kontekstu InfluxDB baze podataka, to se postiže kroz mehanizme klasterizacije, replikacije i mirrora podataka, čime se eliminišu tačke pojedinačnog otkaza i obezbeđuje kontinuirana dostupnost podataka korisnicima.

Cilj ovog rada jeste da detaljno istražimo i prikažemo postojeća HA rešenja kod InfluxDB baze, sa posebnim naglaskom na načine implementacije multi-node klastera, podele uloga čvorova (ingest, query, compact, process), replikaciju i mirroriranje podataka. Takođe, kroz praktične primere i konfiguracione komande prikazaćemo kako se u realnom okruženju može postići otpornost sistema, smanjenje zastoja i povećanje pouzdanosti, što predstavlja osnovni zahtev savremenih informacionih sistema.

# 2. Teorijska osnova

## 2.1. Istorijat InfluxDB

**InfluxDB 1.x**

* Prva stabilna verzija InfluxDB-a, fokusirana na **vremenske serije podataka**.
* Arhitektura: jednostavna server-klijent, lokalno skladištenje u TSDB formatu.
* Podržava **line protocol**, tagove i polja, upite preko InfluxQL jezika.
* Retention policy (politika zadržavanja podataka) za automatsko brisanje starih podataka.
* Ograničena skalabilnost i visoka dostupnost kroz vanjske alate (npr. Kapacitor, Chronograf).

**InfluxDB 2.x**

* Reorganizacija i unapređenje arhitekture, uvodi **buckets** umesto klasičnih baza i retention policy-ja.
* Integracija **InfluxQL** i **Flux** jezika za upite i analizu podataka.
* Ugrađeni web interfejs (UI) za upravljanje bazom, upite i vizualizaciju podataka.
* Podrška za **token-based autorizaciju**, jednostavnija administracija i bolje skaliranje.
* Cilj: olakšati rad sa vremenskim serijama i omogućiti moderni DevOps pristup.

**InfluxDB 3 Core**

* Fokus na **real-time monitoring** i upravljanje aktuelnim podacima.
* Koristi **diskless arhitekturu** i objekt skladište (Parquet format), može raditi samo sa lokalnim diskom ili cloud objekt skladištem.
* Primarni ključ u tabelama: redosled tagova + vreme.
* Token-based autorizacija po defaultu; admin i operator tokeni.
* InfluxDB 3 Enterprise nadograđuje Core sa podrškom za istorijske podatke, read replike, visoku dostupnost i dodatne administrativne funkcije.

A diagram of a cloud

AI-generated content may be incorrect.

Slika 2.1. Istorijat InfluxDB-a

## 2.2. InfluxDB Enterprise

InfluxDB 3 Enterprise je baza podataka izgrađena za prikupljanje, obradu, transformaciju i skladištenje događaja i vremenskih serija podataka. Idealna je za slučajeve upotrebe koji zahtevaju unos podataka u realnom vremenu i brze odgovore na upite za potrebe izrade korisničkih interfejsa, nadzora i rešenja za automatizaciju.

Uobičajeni slučajevi upotrebe uključuju:

* Nadzor podataka sa senzora
* Nadzor servera
* Nadzor performansi aplikacija
* Nadzor mreže
* Analitiku finansijskog tržišta i trgovanja
* Bihejvioralnu analitiku

InfluxDB je optimizovan za scenarije u kojima je nadzor podataka u realnom vremenu od suštinskog značaja i gde upiti moraju da vraćaju rezultate brzo kako bi podržali korisnička iskustva kao što su kontrolne table (dashboards) i interaktivni korisnički interfejsi.

InfluxDB 3 Enterprise je zasnovan na InfluxDB 3 Core, open source izdanju InfluxDB 3.

### 2.2.1. Glavne karakteristike Core izdanja uključuju:

* Arhitektura bez diska sa podrškom za objekt skladište (ili lokalni disk bez zavisnosti)
* Brzo vreme odziva na upite (ispod 10ms za upite poslednje vrednosti ili 30ms za upite različitih metapodataka)
* Ugrađeni Python VM za dodatke (plugins) i trigere
* Persistencija fajlova u Parquet formatu
* Kompatibilnost sa InfluxDB 1.x i 2.x write API-jima

**Enterprise izdanje dodaje sledeće funkcionalnosti u odnosu na Core:**

* Istorijske mogućnosti upita i indeksiranje pojedinačnih serija
* Visoka dostupnost
* Replike za čitanje
* Poboljšana sigurnost (uskoro dostupna)
* Podrška za brisanje na nivou reda (uskoro dostupna)
* Integrisani administratorski korisnički interfejs (uskoro dostupan)

### 2.2.2. Instalacija InfluxDB 3 Enterprise

**Sistemski zahtevi**

*Operativni sistem*

InfluxDB 3 Enterprise radi na Linux, macOS i Windows operativnim sistemima.

*Object storage*

Ključna karakteristika InfluxDB 3 je korišćenje object storage-a za čuvanje vremenskih serija podataka u Apache Parquet formatu. Može se izabrati da se ovi fajlovi čuvaju na lokalnom fajl sistemu. Performanse na lokalnom fajl sistemu su verovatno bolje, ali object storage ima prednost jer ne može ponestati prostora i može mu se pristupiti sa drugih sistema preko mreže. InfluxDB 3 Enterprise nativno podržava Amazon S3, Azure Blob Storage i Google Cloud Storage. Takođe je moguće koristiti mnoge lokalne implementacije object storage-a koje obezbeđuju S3-kompatibilni API, kao što je Minio.

**Instalacija**

InfluxDB 3 Enterprise radi na Linux, macOS i Windows operativnim sistemima.

Mogu se koristiti sledeće metode instalacije InfluxDB 3 Enterprise:

* Brza instalacija za Linux i macOS
* Preuzimanje i instalacija najnovijih build artefakata
* Povlačenje Docker imidža

**Brza instalacija za Linux i macOS**

Za instalaciju InfluxDB 3 Enterprise na Linux ili macOS, preuzima se i pokreće skripta za brzu instalaciju – na primer, korišćenjem curl za preuzimanje skripte:

curl -O https://www.influxdata.com/d/install\_influxdb3.sh \

&& sh install\_influxdb3.sh enterprise

Skripta za brzu instalaciju se ažurira sa svakim izdanjem InfluxDB 3 Enterprise, tako da uvek instalira najnoviju verziju.

**Preuzimanje i instalacija najnovijih build artefakata**

Takođe je moguće direktno preuzeti i instalirati build artefakte InfluxDB 3 Enterprise:

* Linux binarni fajlovi
* macOS binarni fajlovi
* Windows binarni fajlovi

*Windows (AMD64, x86\_64) binarni fajl*

**Verifikacija instalacije**

Nakon instalacije InfluxDB 3 Enterprise, unosi se sledeća komanda da bi se proverilo da li je instalacija uspešno izvršena:

influxdb3 –version

### 2.2.3. Model podataka

InfluxDB 3 Enterprise server sadrži logičke baze podataka; baze podataka sadrže tabele; a tabele se sastoje od kolona.

U poređenju sa prethodnim verzijama InfluxDB-a, baza podataka se može posmatrati kao *InfluxDB v2 bucket* u verziji 2 ili kao *InfluxDB v1 db/retention\_policy*. Tabela je ekvivalentna *measurement* u InfluxDB v1 i v2.

Kolone u tabeli predstavljaju vreme, tagovi (*tags*) i polja (*fields*). Kolone mogu biti sledećih tipova:

* String dictionary (*tag*)
* int64 (*field*)
* float64 (*field*)
* uint64 (*field*)
* bool (*field*)
* string (*field*)
* time (*time sa nanosekundskom preciznošću*)

U InfluxDB 3 Enterprise, svaka tabela ima primarni ključ – uređen skup oznaka (*tags*) i vremena – za svoje podatke. Primarni ključ jedinstveno identifikuje svaki zapis i određuje redosled sortiranja svih Parquet fajlova koji se odnose na tu tabelu. Kada se kreira tabela, bilo eksplicitnim pozivom ili pisanjem podataka u tabelu po prvi put, primarni ključ se postavlja na oznake u redosledu u kojem su pristigle. Iako je InfluxDB i dalje baza podataka tipa *schema-on-write*, definicije kolona oznaka za tabelu su nepromenljive.

Tagovi (*tags*) treba da sadrže jedinstvene identifikatore kao što su *sensor\_id*, *building\_id* ili *trace\_id*. Svi ostali podaci treba da budu skladišteni kao polja (*fields*)

### 2.2.4. Alati za korišćenje

Sledeća tabela upoređuje alate koji se mogu koristiti za interakciju sa InfluxDB 3 Enterprise. Ovaj vodič pokriva mnoge od preporučenih alata.

A screenshot of a computer

AI-generated content may be incorrect.

Slika 2.2. Alati za korišćenje sa InfluxDB Enterprise

### 2.2.5. Podešavanje InfluxDB 3 Enterprise

**Preduslovi**

* InfluxDB 3 Enterprise: instalirati i verifikovati najnoviju verziju na sistemu.
* Ako je potrebno trajno čuvanje podataka, potrebno je imati pristup jednom od sledećih:
  + Direktorijum na lokalnom disku za čuvanje podataka (koristi se u primerima ovog vodiča)
  + S3-kompatibilno objekt skladište i odgovarajuće akreditive

**Pokretanje InfluxDB-a**

Koristi se komanda influxdb3 serve za pokretanje InfluxDB 3 Enterprise. Potrebno je navesti sledeće parametre:

* --node-id: String identifikator koji razlikuje pojedinačne serverske instance unutar klastera. Ovaj parametar formira poslednji deo putanje za skladištenje: <CONFIGURED\_PATH>/<CLUSTER\_ID>/<NODE\_ID>. U višečvornom okruženju ovaj ID se koristi za referenciranje specifičnih čvorova.
* --cluster-id: String identifikator koji određuje deo hijerarhije putanje za skladištenje. Svi čvorovi unutar istog klastera dele ovaj identifikator. Putanja za skladištenje prati obrazac <CONFIGURED\_PATH>/<CLUSTER\_ID>/<NODE\_ID>. U višečvornom okruženju ovaj ID se koristi za referenciranje čitavog klastera.
* --object-store: Navodi tip objekt skladišta koje se koristi. InfluxDB podržava sledeće:
  + file: lokalni fajl sistem
  + memory: u memoriji (bez trajnog čuvanja podataka)
  + memory-throttled: kao *memory*, ali sa latencijom i protokom sličnim cloud objekt skladištu
  + s3: AWS S3 i S3-kompatibilni servisi poput Ceph ili Minio
  + google: Google Cloud Storage
  + azure: Azure Blob Storage
* Ostali parametri objekt skladišta zavise od izabranog tipa. Na primer, za s3 potrebno je navesti ime *bucket*-a i akreditive.

**Arhitektura bez diska**

InfluxDB 3 podržava arhitekturu bez diska, koja može da funkcioniše isključivo sa objekt skladištem, čime se eliminiše potreba za lokalnim diskovima. InfluxDB 3 Enterprise takođe može raditi isključivo sa lokalnim diskom kada je to potrebno.

Kombinovana putanja <CONFIGURED\_PATH>/<CLUSTER\_ID>/<NODE\_ID> obezbeđuje pravilnu organizaciju podataka u objekt skladištu, omogućavajući jasno razdvajanje između klastera i pojedinačnih čvorova.

Za potrebe ovog vodiča koristi se file objekt skladište za trajno čuvanje podataka na lokalnom disku.

**Primer – fajl sistem objekt skladište**

influxdb3 serve \

--node-id host01 \

--cluster-id cluster01 \

--object-store file \

--data-dir ~/.influxdb3

U našem slučaju ovo izgleda ovako:

PS C:\Users\Darko Jotev\Documents\influxdb3-enterprise-3.4.2-windows\_amd64> .\influxdb3.exe serve --node-id host01 --cluster-id cluster01 --object-store file --data-dir ~/.influxdb3

Catalog initialized with uuid: '8fe9a48d-5ce8-4abf-b4aa-b28122fb46c9' and storage hash: 'sha256:XHdcAi2QPzL9c4KyVCxY5covhOu6ROjTpZJ0YeA5TcM'

Didn't find license in object store: Access of Object Store failed: Object at location C:\Users\Darko Jotev\Documents\influxdb3-enterprise-3.4.2-windows\_amd64\~\.influxdb3\cluster01\trial\_or\_home\_license not found: The system cannot find the file specified. (os error 2)

No valid license found: Requested license not available.

InfluxDB 3 Enterprise Setup

To get started, please select a license type:

1) FREE TRIAL

└─ Full features for 30 days (up to 256 cores), perfect for evaluating all Enterprise capabilities

2) COMMERCIAL

└─ Paid commercial license, enabling flexible production environments with dedicated support

3) HOME USE

└─ Free for non-commercial use. Max 2 cores and single node only, ideal for hobbyists and personal projects

Enter option [1-3]:

1

Please enter your email for a Trial license:

darkojotev@elfak.rs

Email sent to darkojotev@elfak.rs. Please check your inbox to verify your email address and proceed.

Waiting for verification...

2025-09-17T12:18:24.351856Z INFO influxdb3\_lib::commands::serve: Creating io runtime executor w/ 2 threads

2025-09-17T12:18:24.352710Z INFO influxdb3\_lib::commands::serve: InfluxDB 3 Enterprise server starting node\_id=host01 mode=[All] git\_hash=5763f603fee63a2024302fcbdd69bde7f57988cf version=3.4.2 uuid=da212490-a21b-4021-aff5-f57e4884e546 licensed\_cores=16

…

**Primeri objekt skladišta**

* Fajl sistem objekt skladište: čuva podatke u navedenom direktorijumu lokalnog fajl sistema (podrazumevani tip).
* Docker sa montiranim fajl sistem objekt skladištem
* Docker Compose sa montiranim fajl sistem objekt skladištem
* S3 objekt skladište
* Objekt skladište zasnovano na memoriji

Za dodatne informacije o opcijama servera koristi se CLI pomoć ili se pregledaju referentne komande za InfluxDB 3 CLI:

influxdb3 serve --help

**Podešavanje licenci**

Kada se prvi put pokrene nova instanca, InfluxDB 3 Enterprise traži izbor tipa licence.

Licence za InfluxDB 3 Enterprise:

* Ovlašćavaju korišćenje softvera za jedan klaster.
* Primjenjuju se po klasteru, uz ograničenja zasnovana prvenstveno na CPU jezgrima.
* Razlikuju se po tipu licence, pri čemu svaka nudi različite mogućnosti i ograničenja.

**Dostupni tipovi licenci**

* *Trial*: probna licenca u trajanju od 30 dana sa punim pristupom mogućnostima InfluxDB 3 Enterprise.
* *At-Home*: licenca za kućnu upotrebu sa ograničenim pristupom mogućnostima InfluxDB 3 Enterprise.
* *Commercial*: komercijalna licenca sa punim pristupom mogućnostima InfluxDB 3 Enterprise.

**Korišćenje InfluxDB 3 Explorer interfejsa**

Preostali koraci u vodiču mogu se obaviti korišćenjem InfluxDB 3 Explorer-a, web interfejsa za upite i administraciju InfluxDB 3. Explorer omogućava vizuelno upravljanje bazama podataka i tokenima, kao i jednostavan način za pisanje i izvršavanje upita nad vremenskim serijama podataka.

### 2.2.6. Podešavanje autorizacije

InfluxDB 3 Enterprise koristi autorizaciju zasnovanu na tokenima da bi se odobrile akcije u bazi podataka. Autorizacija je podrazumevano omogućena kada pokrenemo server. Kada je autorizacija uključena, moramo obezbediti token pri korišćenju **influxdb3 CLI** komandi i **HTTP API** zahteva.

InfluxDB 3 Enterprise podržava sledeće tipove tokena:

* **admin token**: Omogućava pristup svim CLI akcijama i API endpointima.
* **resource tokeni**: Tokeni koji daju prava čitanja i pisanja za određene resurse (baze podataka i sistemske informacione endpoint-e) na serveru.
  + **database token** daje pristup za upis i upite podataka u bazi.
  + **system token** daje pravo čitanja sistemskih informacija i metrika za server.

**Kreiranje operatorskog tokena**

Nakon što pokrenemo server, kreiramo naš prvi admin token. Prvi admin token koji kreiramo postaje operatorski token za server.

Koristimo komandu:

influxdb3 create token –admin

Ova komanda vraća niz tokena koji koristimo za autentifikaciju CLI komandi i API zahteva.

U našem slučaju:

PS C:\Users\Darko Jotev\documents\influxdb3-enterprise-3.4.2-windows\_amd64> .\influxdb3.exe create token --admin

New token created successfully!

Token: apiv3\_N8myzkQXSeNTkedYtN64DpwuOknaNHCezHSVND9MGZJFNRrNnpPFJ7BkZx-FqgmrWj0vikE\_XLh1S6sbkwRrjg

HTTP Requests Header: Authorization: Bearer apiv3\_N8myzkQXSeNTkedYtN64DpwuOknaNHCezHSVND9MGZJFNRrNnpPFJ7BkZx-FqgmrWj0vikE\_XLh1S6sbkwRrjg

IMPORTANT: Store this token securely, as it will not be shown again.

**Bezbedno skladištenje tokena**

InfluxDB prikazuje niz tokena samo u trenutku kada ga kreiramo. Moramo ga bezbedno sačuvati jer ga kasnije nije moguće preuzeti iz baze podataka.

**Postavljanje tokena za autorizaciju**

Operatorski token koristimo za autentifikaciju server akcija u InfluxDB 3 Enterprise, uključujući kreiranje dodatnih tokena, izvršavanje administrativnih zadataka, kao i upis i upite nad podacima.

Možemo koristiti jedan od sledećih metoda za prosleđivanje tokena pri autentifikaciji **influxdb3 CLI** komandi:

* **Kroz promenljivu okruženja (preporučeno)**
* **Kroz opciju komande**

Primer upotrebe kroz opciju komande (zamenjujemo *YOUR\_AUTH\_TOKEN* sopstvenim tokenom):

influxdb3 show databases --token YOUR\_AUTH\_TOKEN

### 2.2.7. Korišćenje MinIO kao object store za InfluxDB 3 Enterprise

Koristimo MinIO kao objektno skladište za našu InfluxDB 3 Enterprise instancu. InfluxDB koristi MinIO S3-kompatibilni API za interakciju sa našim MinIO serverom ili klasterom.

MinIO je visokoperformansno, S3-kompatibilno rešenje za objektno skladištenje, objavljeno pod GNU AGPL v3.0 licencom. Dizajnirano je za brzinu i skalabilnost i omogućava rad sa AI/ML, analitikom i data-intenzivnim radnim opterećenjima uz vrhunske performanse.

MinIO pruža i open source verziju (MinIO Community Edition) i enterprise verziju (MinIO AIStor). Dok obe mogu biti korišćene kao objektno skladište za InfluxDB 3 Enterprise, u ovom vodiču prolazimo kroz korišćenje MinIO Community Edition.

**Setup MinIO**

1. Instaliramo i deploy-ujemo MinIO server ili klaster.
   * Možemo instalirati MinIO lokalno za testiranje i razvoj ili možemo deploy-ovati produkcioni MinIO klaster na više mašina.
2. Preuzimamo i instaliramo MinIO Client (mc).
   * MinIO client, odnosno mc CLI, omogućava nam da izvršavamo administrativne zadatke na našem MinIO serveru ili klasteru, kao što su kreiranje korisnika, dodeljivanje pristupnih politika i slično. Preuzimamo i instaliramo mc CLI za naš lokalni operativni sistem i arhitekturu.
   * Pogledamo sekciju MinIO Client na MinIO downloads stranici.
3. Konfigurišemo mc CLI da se poveže na naš MinIO server ili klaster.
   * mc CLI koristi „alias-e“ za povezivanje na MinIO server ili klaster. Alias se odnosi na set kredencijala koji se koriste za autentifikaciju.
   * Koristimo komandu mc alias set i pružamo sledeće:
     + Alias: jedinstveno ime ili identifikator za ovaj set kredencijala (ALIAS)
     + MinIO URL: URL našeg MinIO servera ili klastera (<https://localhost:9000> ako se pokreće lokalno)
     + Root username: root korisničko ime koje smo definisali pri postavljanju MinIO servera ili klastera (ROOT\_USERNAME)
     + Root password: root lozinku koju smo definisali pri postavljanju MinIO servera ili klastera (ROOT\_PASSWORD)

mc alias set alias http://localhost:9000 minio minio123

1. Kreiramo MinIO bucket.
   * Koristimo MinIO Console ili mc mb komandu da kreiramo novi bucket u našem MinIO serveru ili klasteru.
     + Primer korišćenja mc CLI:

mc mb alias/influxdb3

1. Kreiramo MinIO korisnika.
   * Koristimo mc admin user add komandu i pružamo sledeće:
     + MinIO alias: alias servera (kreiran u koraku 3)
     + Username: jedinstveno korisničko ime (MINIO\_USERNAME)
     + Password: lozinka korisnika (MINIO\_PASSWORD)
   * mc admin user add alias minio1 minio123
   * Kredencijali MinIO korisnika su ekvivalentni AWS S3 kredencijalima:
     + MinIO username ≈ AWS access key ID
     + MinIO password ≈ AWS secret key
2. Kreiramo pristupnu politiku koja omogućava pun pristup influxdb3 bucket-u.
   * MinIO koristi S3 kompatibilne pristupne politike. Kreiramo JSON fajl influxdb3-policy.json sa sledećim sadržajem:

{

"Version": "2012-10-17",

"Statement": [

{

"Action": [

"s3:GetBucketLocation",

"s3:ListBucket"

],

"Effect": "Allow",

"Resource": ["arn:aws:s3:::influxdb3"]

},

{

"Action": [

"s3:PutObject",

"s3:GetObject",

"s3:DeleteObject"

],

"Effect": "Allow",

"Resource": ["arn:aws:s3:::influxdb3/\*"]

}

]

}

* + Koristimo mc admin policy create komandu za kreiranje pristupne politike:

mc admin policy create \

alias \

policy\_name \

/path/to/influxdb3-policy.json

1. Pristupnu politiku dodeljujemo korisniku.
   * Koristimo mc admin policy attach komandu:

mc admin policy attach alias policy\_name --user minio1

* + MinIO podržava dodeljivanje politika korisnicima i grupama. Svi korisnici u grupi nasleđuju politike grupe.

Naš MinIO server ili klaster je sada postavljen i spreman za korišćenje sa InfluxDB 3 Enterprise.

**Konfigurisanje InfluxDB za povezivanje sa MinIO**

Da bismo koristili MinIO kao objektno skladište za InfluxDB 3 Enterprise instancu, pružamo sledeće opcije ili promenljive okruženja prilikom pokretanja influxdb3 serve komande:

* --cluster-id: ID našeg InfluxDB 3 Enterprise klastera (INFLUXDB\_CLUSTER\_ID)
* --node-id: ID našeg InfluxDB 3 Enterprise čvora (INFLUXDB\_NODE\_ID)
* --object-store: s3
* --bucket: influxdb3
* --aws-endpoint: URL našeg MinIO servera (<http://localhost:9000> ako se pokreće lokalno)
* --aws-access-key-id: MinIO korisničko ime (MINIO\_USERNAME)
* --aws-secret-access-key: MinIO lozinka (MINIO\_PASSWORD)
* --aws-allow-http: (opciono) uključiti ako ne koristimo HTTPS

influxdb3 serve \

--cluster-id cluster01 \

--node-id host01 \

--object-store s3 \

--bucket influxdb3 \

--aws-endpoint http://localhost:9000 \

--aws-access-key-id minio1 \

--aws-secret-access-key minio123 \

--aws-allow-http

**Provera funkcionalnosti objektog skladišta**

* Kada InfluxDB 3 Enterprise startuje, inicijalizuje naš MinIO objektni store sa potrebnom strukturom direktorijuma i počinje da skladišti podatke.
* Potvrđujemo funkcionalnost objektog skladišta:
  + Pregledamo influxdb3 serve log izlaz da potvrdimo da server radi ispravno.
  + Inspektujemo sadržaj našeg MinIO influxdb3 bucket-a da potvrdimo kreiranje neophodne strukture direktorijuma. Možemo koristiti MinIO Console ili mc ls komandu:

mc ls alias/influxdb3

**2.2.8. Kreiranje multi-node klastera**

Kreiramo **multi-node InfluxDB 3 Enterprise klaster** kako bismo obezbedili **high availability, performanse i izolaciju radnog opterećenja**. Konfigurišemo čvorove sa specifičnim modovima (ingest, query, process, compact) kako bismo optimizovali klaster prema našem scenariju korišćenja.

**Preduslovi**

* Potrebno je imati **shared object store** koji svi čvorovi u klasteru mogu koristiti za skladištenje podataka i koordinaciju shard-ova.

**Server modovi**

U InfluxDB 3 Enterprise, **modovi servera** određuju koje **subprocese čvor izvršava**. Ovi subprocessi obavljaju neophodne zadatke, uključujući prikupljanje **podataka, obradu upita, kompakciju podataka** i izvršavanje **processing engine** plugin-a.

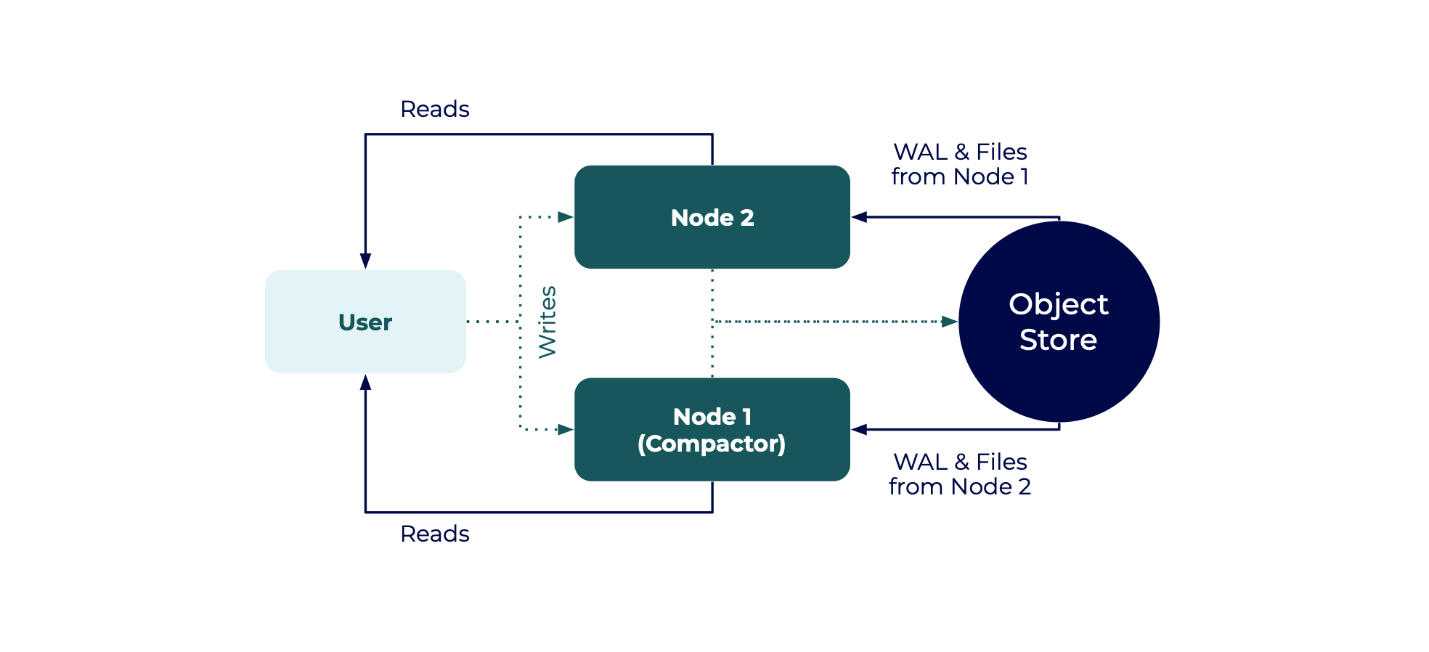
Opcija influxdb3 serve --mode definiše koje subprocess-e čvor pokreće. Svaki čvor može raditi u jednom ili više sledećih modova:

* **all (podrazumevano):** Pokreće sve neophodne subprocess-e.
* **ingest:** Pokreće subprocess za prikupljanje (ingest) podataka i obradu upisa.
* **query:** Pokreće subprocess za obradu upita.
* **process:** Pokreće subprocess processing engine-a za aktiviranje i izvršavanje plugin-a.
* **compact:** Pokreće subprocess kompaktora koji optimizuje podatke u objektno skladište.

### 2.2.9. Primeri konfiguracije klastera

#### 2.2.9.1. High availability klaster

Za postizanje osnovne **high availability (HA)** konfiguracije, potreban je minimum dva čvora. U ovakvom klasteru oba čvora istovremeno čitaju i upisuju podatke, čime obezbeđujemo kontinuitet rada i otpornost sistema na otkaze pojedinačnih čvorova.



Slika 2.3. High availability cluster

**Osnovna HA konfiguracija**

U osnovnom **HA (High Availability)** podešavanju:

* Dva čvora istovremeno upisuju podatke u isto objektno skladište i oba obrađuju upite.
* Čvor 1 i Čvor 2 funkcionišu kao **read replica** instanci, pri čemu čitaju iz direktorijuma objektnog skladišta jedan drugoga.
* Jedan od čvorova se definiše kao **Compactor čvor**.
* Samo jedan čvor može biti određen kao Compactor, jer je kompaktovani skup podataka namenjen jednom piscu i više čitaoca.

Slede primeri konfiguracije i pokretanja dva čvora u osnovnoj HA postavci:

**Čvor 1 (sa kompaktovanjem)**

# Primer promenljivih

# node-id: 'host01'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host01 \

--cluster-id cluster01 \

--mode ingest,query,compact \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

--http-bind localhost:8181 \

--aws-access-key-id <AWS\_ACCESS\_KEY\_ID> \

--aws-secret-access-key <AWS\_SECRET\_ACCESS\_KEY>

**Čvor 2 (ingest i query)**

# Primer promenljivih

# node-id: 'host02'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host02 \

--cluster-id cluster01 \

--mode ingest,query \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

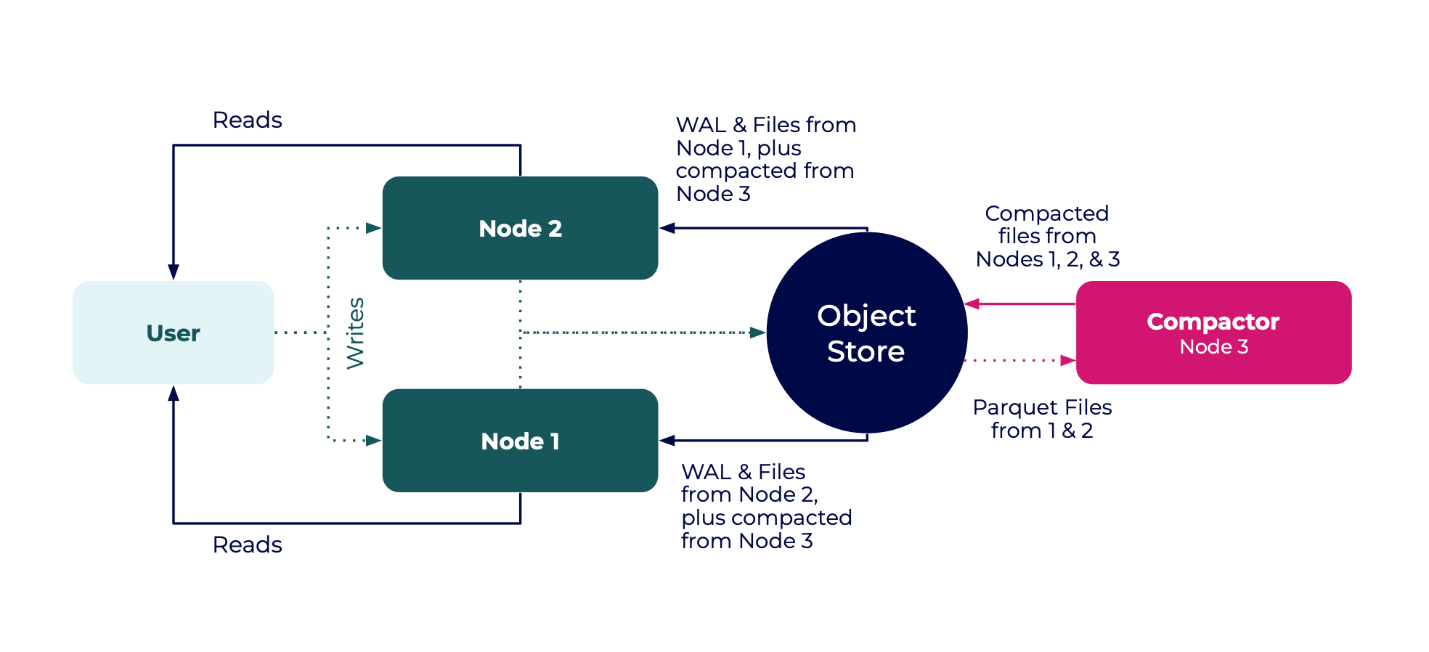
--aws-access-key-id AWS\_ACCESS\_KEY\_ID \

--aws-secret-access-key AWS\_SECRET\_ACCESS\_KEY

Nakon pokretanja čvorova, upiti prema bilo kom čvoru vraćaju podatke iz oba čvora, dok Čvor 1 obavlja kompakciju. Da bismo dodali nove čvorove u ovu postavku, pokrećemo dodatne **read replica** čvorove sa istim ID-jem klastera.

#### 2.2.9.2. High availability sa dedicated Compactor čvorom

**Kompakcija podataka** u InfluxDB 3 Enterprise predstavlja jednu od računarski najzahtevnijih operacija. Da bismo obezbedili stabilne performanse čvorova koji se koriste za **ingest i upite**, postavljamo poseban čvor koji služi isključivo kao **Compactor**. Na taj način izolujemo proces kompakcije od ostalih radnih opterećenja.



Slika 2.4. High availability with a dedicated Compactor

**Konfiguracija dedicated Compactor-a**

Za ovu postavku pokrećemo dva čvora za čitanje i pisanje (read-write), koji funkcionišu kao **read replica** instanci – slično kao u prethodnom primeru. Pored njih, dodajemo treći čvor koji je posvećen isključivo kompakciji.

**Čvor 1 — Writer/Reader Node #1**

# Primer promenljivih

# node-id: 'host01'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host01 \

--cluster-id cluster01 \

--mode ingest,query \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

--http-bind localhost:8181 \

--aws-access-key-id <AWS\_ACCESS\_KEY\_ID> \

--aws-secret-access-key <AWS\_SECRET\_ACCESS\_KEY

**Čvor 2 — Writer/Reader Node #2**

# Primer promenljivih

# node-id: 'host02'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host02 \

--cluster-id cluster01 \

--mode ingest,query \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

--aws-access-key-id <AWS\_ACCESS\_KEY\_ID> \

--aws-secret-access-key <AWS\_SECRET\_ACCESS\_KEY>

**Čvor 3 — Compactor Node**

Da bismo obezbedili da se ovaj čvor koristi isključivo za kompakciju, pokrećemo ga sa opcijom --mode=compact.

# Primer promenljivih

# node-id: 'host03'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host03 \

--cluster-id cluster01 \

--mode compact \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

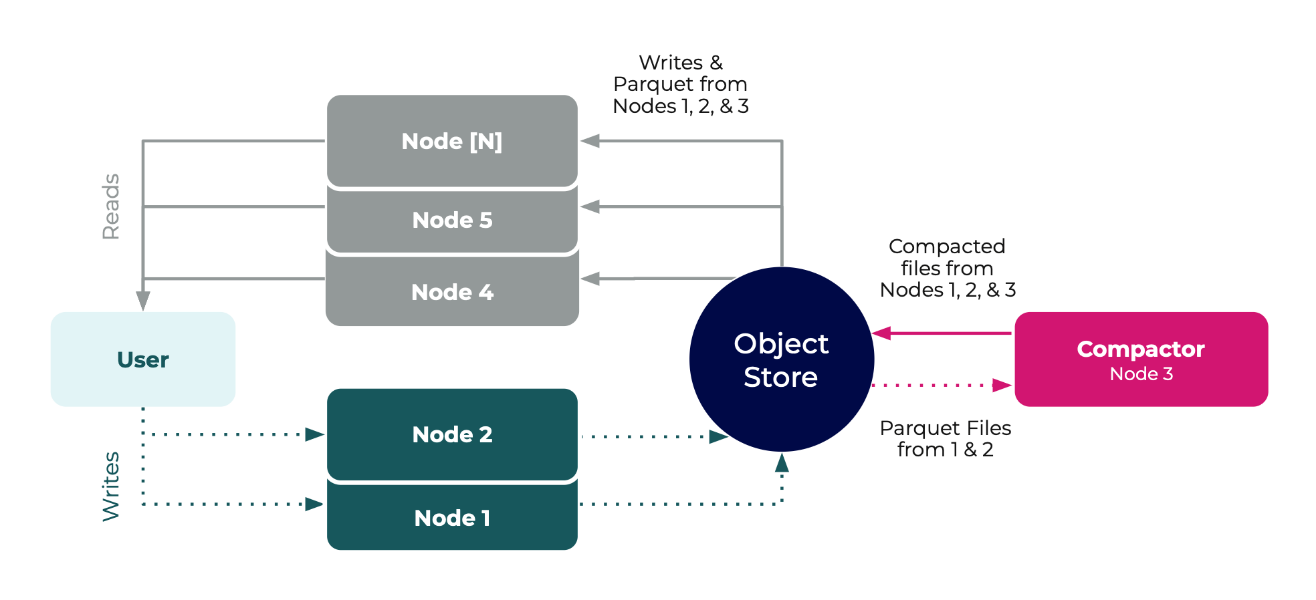
--aws-access-key-id <AWS\_ACCESS\_KEY\_ID> \

--aws-secret-access-key <AWS\_SECRET\_ACCESS\_KEY>

Ova konfiguracija omogućava da čvorovi namenjeni za prikupljanje podataka (ingest) i obradu upita rade bez dodatnog opterećenja, dok se proces kompakcije centralizuje na dedicated čvor, čime se poboljšava **stabilnost i raspodela opterećenja** u HA okruženju.

#### 2.2.9.3. High availability sa read replicas i dedicated Compactor čvorom

Za **robustno i efikasno okruženje** za upravljanje vremenskim serijama podataka, možemo kombinovati čvorove za ingest sa čvorovima za upite i posebnim **dedicated Compactor** čvorom. Ovakva konfiguracija omogućava **potpunu izolaciju radnog opterećenja** i optimalnu raspodelu resursa.



Slika 2.5.High availability with read replicas and a dedicated Compactor

**Workload Isolation Setup**

* Pokrećemo **ingest čvorove** u ingest modu.
* Svi **zahtevi za upis podataka** šalju se isključivo na ingest čvorove.
* **Compactor čvor** pokrećemo u compact modu.
* **Query čvorovi** se pokreću u query modu, i svi **zahtevi za čitanje/analizu podataka** šalju se samo njima.

**Čvor 1 — Writer Node #1**

# Primer promenljivih

# node-id: 'host01'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host01 \

--cluster-id cluster01 \

--mode ingest \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

--http-bind localhost:8181 \

--aws-access-key-id <AWS\_ACCESS\_KEY\_ID> \

--aws-secret-access-key <AWS\_SECRET\_ACCESS\_KEY>

**Čvor 2 — Writer Node #2**

# Primer promenljivih

# node-id: 'host02'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host02 \

--cluster-id cluster01 \

--mode ingest \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

--aws-access-key-id <AWS\_ACCESS\_KEY\_ID> \

--aws-secret-access-key <AWS\_SECRET\_ACCESS\_KEY>

**Čvor 3 — Compactor Node**

# Primer promenljivih

# node-id: 'host03'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host03 \

--cluster-id cluster01 \

--mode compact \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

--aws-access-key-id <AWS\_ACCESS\_KEY\_ID> \

--aws-secret-access-key <AWS\_SECRET\_ACCESS\_KEY>

**Čvor 4 — Read Node #1**

# Primer promenljivih

# node-id: 'host04'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host04 \

--cluster-id cluster01 \

--mode query \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

--aws-access-key-id <AWS\_ACCESS\_KEY\_ID> \

--aws-secret-access-key <AWS\_SECRET\_ACCESS\_KEY>

**Čvor 5 — Read Node #2**

# Primer promenljivih

# node-id: 'host05'

# cluster-id: 'cluster01'

# bucket: 'influxdb-3-enterprise-storage'

influxdb3 serve \

--node-id host05 \

--cluster-id cluster01 \

--mode query \

--object-store s3 \

--bucket influxdb-3-enterprise-storage \

--aws-access-key-id <AWS\_ACCESS\_KEY\_ID> \

--aws-secret-access-key <AWS\_SECRET\_ACCESS\_KEY>

Ovom konfiguracijom postižemo **jasnu podelu uloga**: ingest čvorovi obrađuju samo upise, query čvorovi obrađuju samo upite, a dedicated Compactor preuzima najzahtevniji proces kompakcije. Na taj način povećavamo **stabilnost, performanse i otpornost** InfluxDB Enterprise klastera u okruženju sa visokom dostupnošću.

### 2.2.10. Upisivanje i izvršavanje upita u multi-node klasterima

U multi-node klasterima InfluxDB 3 Enterprise možemo koristiti podrazumevani port 8181 za bilo koji **write** ili **query** zahtev, bez potrebe za promenom komandi.

**Specifikacija hostova za upis i izvršavanje upita**

Da bismo u potpunosti iskoristili prednosti **multi-node izolovane arhitekture**, potrebno je da pravilno usmeravamo zahteve:

* **Zahteve za upis (write requests)** šaljemo ka čvoru koji je određen kao **ingester**.
* **Zahteve za izvršavanje upita (query requests)** šaljemo ka čvoru koji je određen kao **querier**.

U situacijama kada pokrećemo više lokalnih instanci za potrebe testiranja ili kada koristimo odvojene čvorove u produkcionom okruženju, precizno navođenje hosta omogućava da upiti i upisi budu usmereni na ispravnu instancu.

**Primer upita ka određenom hostu**

# Primer kada čvor koristi HTTP port: 8585

influxdb3 query \

--host http://localhost:8585 \

--token AUTH\_TOKEN \

--database DATABASE\_NAME \

"QUERY"

**Objašnjenje parametara**

* [**http://localhost:8585**](http://localhost:8585) – host i port čvora kojem se upit šalje.
* **AUTH\_TOKEN** – token baze podataka koji ima dozvole za izvršavanje upita nad specificiranom bazom.
* **DATABASE\_NAME** – naziv baze nad kojom se upit izvršava.
* **QUERY** – SQL ili InfluxQL upit koji želimo da izvršimo.

Ovim pristupom obezbeđujemo da **svaka instanca čvora u klasteru obrađuje samo zahteve za koje je optimizovana**, što je ključni preduslov za stabilnost i visoke performanse u distribuiranom okruženju sa visokom dostupnošću.

### 2.2.11. Upis podataka u InfluxDB 3 Enterprise

InfluxDB 3 Enterprise je dizajniran za visok **propis podataka (high write-throughput)** i koristi efikasnu, čoveku čitljivu sintaksu za upis podataka, poznatu kao **line protocol**. InfluxDB funkcioniše kao **schema-on-write** baza podataka, što znači da možemo odmah započeti upis podataka, a InfluxDB automatski kreira logičku bazu podataka, tabele i njihove šeme bez potrebe za dodatnom intervencijom. Nakon kreiranja šeme, sistem validira sve buduće zahteve za upis podataka u skladu sa postojećom šemom, pre nego što prihvati nove podatke. Po potrebi, moguće je dodavati nove **tagove** i **polja (fields)** kada dođe do promene šeme.

**Line protocol**

InfluxDB 3 Enterprise prihvata podatke u formatu **line protocol**. Ova sintaksa sastoji se od sledećih elemenata:

* **table (obavezno):** String koji identifikuje tabelu u kojoj se podaci čuvaju.
* **tag set:** Lista parova ključ-vrednost odvojenih zarezima, pri čemu svaki par predstavlja tag. Ključevi i vrednosti tagova se navode kao neoznačeni stringovi. Karakteri kao što su razmak, zarez i znak jednakosti moraju biti „escape-ovani“.
* **field set:** Lista parova ključ-vrednost odvojenih zarezima, pri čemu svaki par predstavlja polje. Ključevi polja su neoznačeni stringovi, pri čemu razmaci i zarezi moraju biti „escape-ovani“. Vrednosti polja mogu biti sledećih tipova:
  + stringovi (u navodnicima)
  + brojevi sa pokretnim zarezom (floats)
  + celobrojne vrednosti (integers)
  + neoznačeni celobrojni tipovi (unsigned integers)
  + booleove vrednosti (true/false)
* **timestamp:** Unix vremenska oznaka povezana sa podacima. InfluxDB podržava preciznost do nivoa nanosekundi.

**Kako se parsiraju elementi line protocol-a?**

* **table:** Sve što se nalazi pre prvog ne-escape-ovanog zareza pre prvog praznog znaka (whitespace).
* **tag set:** Parovi ključ-vrednost između prvog ne-escape-ovanog zareza i prvog praznog znaka.
* **field set:** Parovi ključ-vrednost između prvog i drugog praznog znaka.
* **timestamp:** Celobrojna vrednost koja dolazi posle drugog praznog znaka.

Svaka linija se završava znakom za novi red (\n). Sintaksa line protocol-a je **osetljiva na razmake**, što znači da je precizno formatiranje obavezno.

A text on a white background

AI-generated content may be incorrect.

Slika 2.6. Line protocol

**Konstruisanje line protocol-a**

Sa osnovnim razumevanjem line protocol-a, možemo da konstrušemo sopstvene izraze i upisujemo podatke u **InfluxDB 3 Enterprise**.

Razmotrimo primer korišćenja u kojem prikupljamo podatke sa senzora u domaćinstvu. Svaki senzor meri **temperaturu**, **vlažnost vazduha** i **nivo ugljen-monoksida**. Za prikupljanje ovih podataka koristi se sledeća šema:

* **tabela:** home
* **tagovi:**
  + room: Living Room ili Kitchen
* **polja (fields):**
  + temp: temperatura u °C (*float*)
  + hum: relativna vlažnost u procentima (*float*)
  + co: ugljen-monoksid u delovima po milionu (*integer*)
* **timestamp:** Unix vremenska oznaka sa preciznošću u sekundama

Primer **line protocol** podataka prikupljenih na satnom nivou, od 2025-09-14T08:00:00Z do 2025-09-14T20:00:00Z (UTC):

home,room=Living\ Room temp=21.1,hum=35.9,co=0i 1757836800

home,room=Kitchen temp=21.0,hum=35.9,co=0i 1757836800

home,room=Living\ Room temp=21.4,hum=35.9,co=0i 1757840400

home,room=Kitchen temp=23.0,hum=36.2,co=0i 1757840400

home,room=Living\ Room temp=21.8,hum=36.0,co=0i 1757844000

home,room=Kitchen temp=22.7,hum=36.1,co=0i 1757844000

home,room=Living\ Room temp=22.2,hum=36.0,co=0i 1757847600

home,room=Kitchen temp=22.4,hum=36.0,co=0i 1757847600

home,room=Living\ Room temp=22.2,hum=35.9,co=0i 1757851200

home,room=Kitchen temp=22.5,hum=36.0,co=0i 1757851200

home,room=Living\ Room temp=22.4,hum=36.0,co=0i 1757854800

home,room=Kitchen temp=22.8,hum=36.5,co=1i 1757854800

home,room=Living\ Room temp=22.3,hum=36.1,co=0i 1757858400

home,room=Kitchen temp=22.8,hum=36.3,co=1i 1757858400

home,room=Living\ Room temp=22.3,hum=36.1,co=1i 1757862000

home,room=Kitchen temp=22.7,hum=36.2,co=3i 1757862000

home,room=Living\ Room temp=22.4,hum=36.0,co=4i 1757865600

home,room=Kitchen temp=22.4,hum=36.0,co=7i 1757865600

home,room=Living\ Room temp=22.6,hum=35.9,co=5i 1757869200

home,room=Kitchen temp=22.7,hum=36.0,co=9i 1757869200

home,room=Living\ Room temp=22.8,hum=36.2,co=9i 1757872800

**Pisanje podataka pomoću CLI alata**

Za brz početak upisa podataka koristi se komanda **influxdb3 write**. Potrebno je navesti sledeće opcije:

* --database – ime baze podataka u koju se piše
* --token – autentifikacioni token sa pravima za pisanje (ako nije već podešen preko promenljive okruženja INFLUXDB3\_AUTH\_TOKEN)
* podatke u formatu line protocol unete preko standardnog ulaza (*stdin*)

Primer:

influxdb3 write \

--database DATABASE\_NAME \

--token AUTH\_TOKEN \

--precision s \

'home,room=Living\ Room temp=21.1,hum=35.9,co=0i 1641024000

home,room=Kitchen temp=21.0,hum=35.9,co=0i 1641024000

home,room=Living\ Room temp=21.4,hum=35.9,co=0i 1641027600

home,room=Kitchen temp=23.0,hum=36.2,co=0i 1641027600

home,room=Living\ Room temp=21.8,hum=36.0,co=0i 1641031200

home,room=Kitchen temp=22.7,hum=36.1,co=0i 1641031200

home,room=Living\ Room temp=22.2,hum=36.0,co=0i 1641034800

home,room=Kitchen temp=22.4,hum=36.0,co=0i 1641034800

home,room=Living\ Room temp=22.2,hum=35.9,co=0i 1641038400

home,room=Kitchen temp=22.5,hum=36.0,co=0i 1641038400

home,room=Living\ Room temp=22.4,hum=36.0,co=0i 1641042000

home,room=Kitchen temp=22.8,hum=36.5,co=1i 1641042000

home,room=Living\ Room temp=22.3,hum=36.1,co=0i 1641045600

home,room=Kitchen temp=22.8,hum=36.3,co=1i 1641045600

home,room=Living\ Room temp=22.3,hum=36.1,co=1i 1641049200

home,room=Kitchen temp=22.7,hum=36.2,co=3i 1641049200

home,room=Living\ Room temp=22.4,hum=36.0,co=4i 1641052800

home,room=Kitchen temp=22.4,hum=36.0,co=7i 1641052800

home,room=Living\ Room temp=22.6,hum=35.9,co=5i 1641056400

home,room=Kitchen temp=22.7,hum=36.0,co=9i 1641056400

home,room=Living\ Room temp=22.8,hum=36.2,co=9i 1641060000

home,room=Kitchen temp=23.3,hum=36.9,co=18i 1641060000

home,room=Living\ Room temp=22.5,hum=36.3,co=14i 1641063600

home,room=Kitchen temp=23.1,hum=36.6,co=22i 1641063600

home,room=Living\ Room temp=22.2,hum=36.4,co=17i 1641067200

home,room=Kitchen temp=22.7,hum=36.5,co=26i 1641067200'

### 2.2.12. Demonstracija High availability sa dedicated Compactor čvorom

Pokretanje MinIO Object Storage Server-a:

.\minio.exe server ./

MinIO se pokrece lokalno na portu 9000. Zatim pokrećemo InfluxDB instance tako da koriste MinIO Object Storage Server:

// Pokretanje W/R node #1 na portu 8181  
.\influxdb3.exe serve --node-id host01 --cluster-id cluster01 --mode ingest,query --object-store s3 --bucket influxdb3 --http-bind localhost:8181 --aws-endpoint http://localhost:9000 --aws-access-key-id minio1 --aws-secret-access-key minio123 --aws-allow-http

//Pokretanje W/R node #2 na portu 8182

.\influxdb3.exe serve --node-id host02 --cluster-id cluster01 --mode ingest,query --object-store s3 --bucket influxdb3 --http-bind localhost:8182 --aws-endpoint http://localhost:9000 --aws-access-key-id minio1 --aws-secret-access-key minio123 --aws-allow-http  
  
//Pokretanje Compactor Node na portu 8183   
.\influxdb3.exe serve --node-id host03 --cluster-id cluster01 --mode compact --object-store s3 --bucket influxdb3 --http-bind localhost:8183 --aws-endpoint http://localhost:9000 --aws-access-key-id minio1 --aws-secret-access-key minio123 --aws-allow-http

Kreiramo token I vršimo upis na jednom od R-W nodova:  
  
.\influxdb3.exe create token --admin --host http://localhost:8181

New token created successfully!

Token: apiv3\_3-05k47hbXYeHEfI2hWdCq3xcPREIyZnR3z3cIG5qBzgXfUscN5Kw6pDqe8uskyCryUP\_N1LWAXlW4IzB-jHxw

HTTP Requests Header: Authorization: Bearer apiv3\_3-05k47hbXYeHEfI2hWdCq3xcPREIyZnR3z3cIG5qBzgXfUscN5Kw6pDqe8uskyCryUP\_N1LWAXlW4IzB-jHxw

IMPORTANT: Store this token securely, as it will not be shown again.

.\influxdb3.exe write --host http://localhost:8181 --database home --token apiv3\_3-05k47hbXYeHEfI2hWdCq3xcPREIyZnR3z3cIG5qBzgXfUscN5Kw6pDqe8uskyCryUP\_N1LWAXlW4IzB-jHxw "home,room=Living\ Room temp=22.5,hum=36,co=5i 1641024000"

success

Simuliramo pad R-W noda na portu 8181 I pokusavamo citanje na nodu sa portom 8182:  
  
.\influxdb3.exe query --database home --host http://localhost:8182 --token apiv3\_3-05k47hbXYeHEfI2hWdCq3xcPREIyZnR3z3cIG5qBzgXfUscN5Kw6pDqe8uskyCryUP\_N1LWAXlW4IzB-jHxw "SELECT \* FROM home"

+----+------+-------------+------+---------------------+

| co | hum | room | temp | time |

+----+------+-------------+------+---------------------+

| 5 | 36.0 | Living Room | 22.5 | 2022-01-01T08:00:00 |

+----+------+-------------+------+---------------------+

# 5. Zaključak

Na osnovu istraženih aspekata i primera implementacije, može se zaključiti da InfluxDB 3 Enterprise predstavlja snažno rešenje za upravljanje vremenskim serijama podataka, sa posebnim naglaskom na visoku dostupnost (High Availability). Uvođenjem podrške za multi-node klastere, read replike i različite režime rada čvorova (ingest, query, process, compact), InfluxDB omogućava otpornost sistema na otkaze, balansiranje opterećenja i optimizaciju performansi.

Upotreba objektnog skladišta (lokalnog ili cloud) dodatno povećava fleksibilnost i skalabilnost, dok integracija sa alatima poput MinIO omogućava prilagodljivost i u okruženjima gde je potreban open-source pristup. Token-based autorizacija i administrativni interfejs doprinose sigurnosti i olakšanoj kontroli pristupa.

High availability rešenja kod InfluxDB-a nisu samo tehnički mehanizam za sprečavanje prekida rada, već predstavljaju osnovu za stabilan, skalabilan i siguran sistem koji može da odgovori na zahteve modernih aplikacija u oblastima nadzora, analitike i automatizacije.

Upravo zbog toga, InfluxDB 3 Enterprise se pokazuje kao optimalan izbor za organizacije koje žele da obezbede kontinuiran rad sa kritičnim vremenskim serijama podataka, pri čemu se minimiziraju rizici od gubitka informacija i prekida u radu sistema.

# 6. Literatura

1. InfluxData Documentation: <https://docs.influxdata.com/influxdb3/core/>
2. <https://docs.influxdata.com/influxdb3/core/get-started/>
3. [Inside InfluxDB 3 Core from the Creator Himself: Rust Rewrite, Object Storage, and More | GrafanaCON - YouTube](https://www.youtube.com/watch?v=mKu1PIhl2Ms)
4. <https://docs.influxdata.com/influxdb3/core/get-started/setup/>
5. <https://docs.influxdata.com/influxdb3/core/reference/internals/>
6. <https://docs.influxdata.com/influxdb3/enterprise/>
7. <https://docs.influxdata.com/influxdb3/enterprise/install/>
8. <https://docs.influxdata.com/influxdb3/enterprise/get-started/>
9. <https://docs.influxdata.com/influxdb3/enterprise/get-started/setup/>
10. <https://docs.influxdata.com/influxdb3/enterprise/get-started/multi-server/>
11. <https://docs.influxdata.com/influxdb3/enterprise/get-started/write/>
12. <https://docs.influxdata.com/influxdb3/enterprise/admin/object-storage/>
13. <https://docs.influxdata.com/influxdb3/enterprise/admin/object-storage/minio/>