

# DevOps Challenge Documentation

## ۱. هدف پروژه

این پروژه مربوط به چالش DevOps بود که باید عملکرد **RocksDB** را در مواجهه با داده‌های حجیم (میلیون‌ها هش ۱۲۸ بیتی) بررسی می‌کرد. تمرکز پروژه روی این بود که:

- داده‌ها هم روی دیسک (**SSD**) و هم در رم (**ramfs**) تست شوند،
- و چند پارامتر مختلف (مثل Bloom filter، نوع compaction، فرمت SST و اندازه دیتابیس) تغییر کنند تا تأثیرشان روی سرعت و مصرف منابع اندازه‌گیری شود.

---

فایل **devops\_challenge.pdf** خوانده شد تا اهداف و نیازها مشخص شوند. مفاهیمی که احتیاج بود راجع بهشون بخونم:

- LSM Tree و نقش MemTable / SST
- تفاوت تنظیمات RocksDB (BlockBasedTable، PlainTable، CuckooTable)
- مفهوم Bloom Filter و Compaction Style

## ساخت محیط develop

و یک پروژه کوچک ++C برای بنچمارک نوشته شد تا دو کار انجام دهد:

۱. نوشتن (**bulk load**) – تولید N هش تصادفی و نوشتن آن‌ها در RocksDB
۲. خواندن (**lookup**) – انجام چند هزار lookup تصادفی و اندازه‌گیری زمان

کد خروجی ذکر شده:

Inserted 100000 hashes in 0.0436046s

Read 100000 hashes in 0.0782298s

SSD

Inserted 100000 hashes in 0.044741s  
Read 100000 hashes in 0.06563s  
RAM

بعد از اون بخش‌های مختلفش رو بررسی کردم تا بهتر بفهمم که چطور کار می‌کنه و چطوری باید کدش رو گسترش بدم.  
به مرور ویژگی‌های مرتبط با متغیرهای مورد نیاز در پروژه با استفاده از ai اضافه کردم  
برای دو محیط:

1. روی SSD

2. روی RAM disk

و در نهایت با استفاده از Python و کتابخانه‌های **pandas** و **matplotlib** داده‌ها تجزیه و تحلیل شدند:

python3 analyze\_data.py

این اسکریپت میانگین **throughput** و **latency** هر ترکیب را محاسبه کرده و نمودار تولید کرد.

---

### ۳. استفاده از ChatGPT

در این پروژه از ChatGPT برای مراحل زیر استفاده شد:

کاربرد ChatGPT	مرحله
نوشتن فایل‌های <b>CMakeLists.txt</b> و ساختار <b>build</b>	تولید الگو و رفع خطاهای <b>build</b>
تحلیل داده‌ها با <b>pandas</b>	طراحی اسکریپت تحلیل و رسم نمودار
نوشتن مستندات و <b>README</b>	تولید قالب اولیه و بهبود ساختار نگارشی
راه‌اندازی <b>ramfs</b> روی <b>macOS</b>	دریافت دستورات سیستم برای تست <b>RAM-based setup</b>

---

#### ۴. نحوه‌ی اعتبارسنجی

برای اطمینان از درستی نتایج:

##### مقایسه اجراهای تکراری:

هر تست حداقل دو بار اجرا شد و نتایج میانگین گرفته شد.

##### مقایسه منطقی:

بررسی شد که با افزایش `read_batch`، انتظار می‌رود `throughput` بیشتر و `latency` کمتر شود. همین‌طور، فعال‌کردن Bloom Filter باید زمان `lookup` را کاهش دهد.

##### تست محیطی:

مقایسه بین اجرای روی `SSD` و `RAM disk` تا تفاوت رفتار حافظه و دیسک مشخص شود.

---

#### ۶. جمع‌بندی

در این پروژه یاد گرفتیم چطور:

- یک سیستم `benchmark` ساده طراحی کنم،
- تنظیمات `RocksDB` را تغییر بدهم.

نتایج نهایی رو `Disk`:





