

2025

با فرز لِنْتَرْفِيو

System Design?



التالي



مشاركة



حفظ

مختبراتيّاً مفهوم بازجِم تكون بمثابة v كُويس قبل إنترفيو System Design

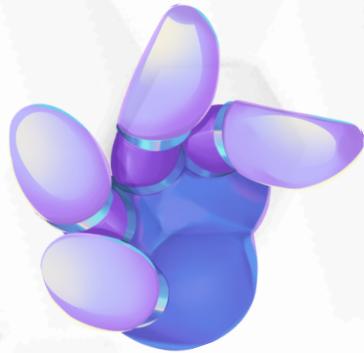
- مفهوم system design في ال interview
- اتعلّم بسرعة وتدخل ال design مثل هتسل عنهم.
- واثق وانت want من نفسك interview



مشاركة



حفظ



المعاملات الموزعة

Distributed Transactions

لما يكون عندك طلب بيعديل بيانات فى كذا مكان ، لازم كل العمليات تنجح مع بعض او كلها تتلغى علشان تحافظ على صحة البيانات.

الحلول

- sagas لتحقيق الاستقرار النهائي بشكل مرن
- 2Pc (2 phases commit) يضمن تنفيذ أو إلغاء العملية لتفادي خلاف البيانات، بس نادر لما بنستخدمه علشان معقد.

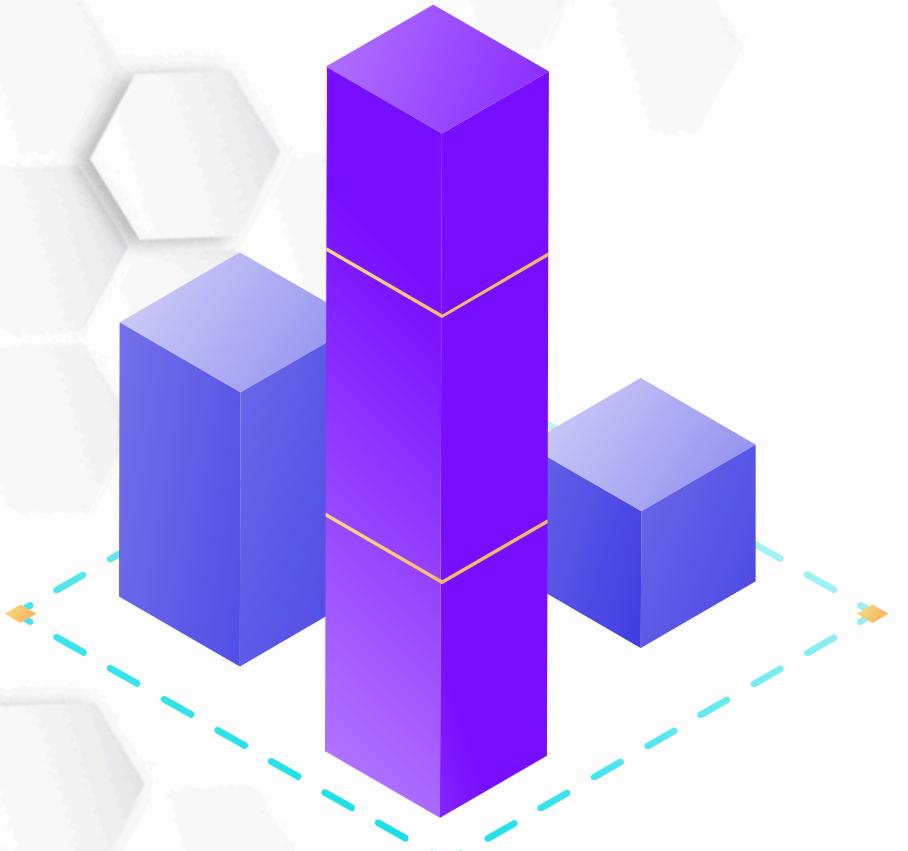


مشاركة



حفظ

مصدر الأحداث



- بدل متزن اخر حالة للبيانات ، خزن كل حدث(Event) حصل في النظام.
- لما تحتاج تعرف الحالة الحالية، ببساطه بتعيد تشغيل كل الاصدارات بالترتيب.
- كأنك بتكتب سجل تاريخي مفصل لكل حادث دخلت في النظام!



مشاركة



حفظ

إنتقاط تغيرات البيانات

Change Data Capture



- كل ما يحصل تغيير في قاعده البيانات،
CDC يرسله ك Event لأنظمه تانية.

الأنظمه التانية تقدر تعرف على التغيرات زي:

- تدريب قاعده بيانات تانية بنفس التغيرات
- تجديد محرك البحث تلقائي : مفيد جدا لما الأنظمه تحتاج
تتكلم مع بعض بطريقه مرنه و من غير متعتمد على
بعض بشكل مباشر



مشاركة



حفظ

إدراه العمليات الكبيره بين الخدمات

SAGAS

- لما يكون عندك مهمه كبيره متقسسه على خدمات كتير، بتضمن يا إما كل الخطوات تكمل بنجاح، يا اللي حصل منها يتلغي.
- حل آمن لإداره العمليات اللي بتم على اكتر من اعماليه مختلفه.

مثال عملي:

لو عميل اشتري منتج (دفع، تدبيث مخزون، ارسال طلب) بس المخزون طلع فاضي بنرجع الفلوس للعميل علاهول.



مشاركة



حفظ

تنظيم التعاون بين الخدمات

Choreography VS Orchestration

Choreography (التنسيق الحر)

- كل خدمه بتنصرف لوحدها بناء على الـ events اللي بتراحت. مفيش مدبر مركزى بيقولهم يعملو ايه.

مثال: نظام تسجيل الطلبات يبعث event, وخدمه المخزون تسمعه وتترك تلقائى.



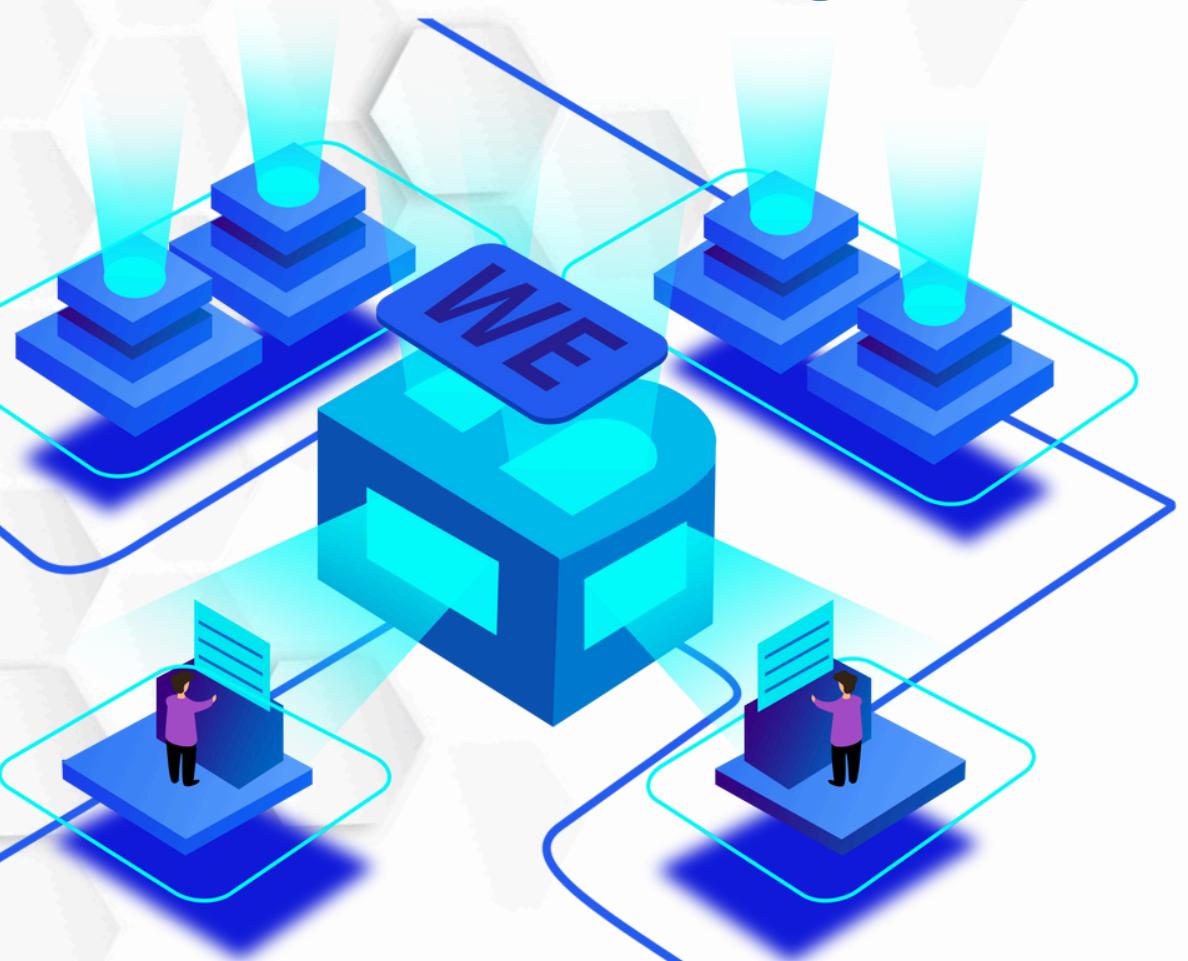
مشاركة



حفظ

تنظيم التعاون بين الخدمات

Choreography VS Orchestration



orchestration (تدكيم مركزي)

- فيه خدمة رئيسية بتندركم في كل حاجة و تقول لكل خدمة تعمل ايه و امتى. اسهل في التتبع و مراقبة الشغل.

مثال: خدمة إداره الطلبات بترتيب كل الخطوات من أول الدفع لحد الشحن.



مشاركة



حفظ

لازم تختار و تضمن بحاجه! CAP Theorem

نظريه CAP بتقول إن اي نظام كبير و مدحوط على اكتر من سيرفر
ميففعش يتحقق الثلاثه مع بعض. لازم تختار اتنين بس من:

- **تطابق البيانات (consistency):** كل السيرفرات تشفوف نفس البيانات في نفس الوقت.
- **استجابه سريعه (Availability):** اي طلب من المستخدم لازم النظام يرد عليه, حتى لو الرد خطأ او بيانات قديمه.
- **تحمل مشاكل الشبكة (Partition Tolerance):** لو حصل مشكله بين السيرفرات او الشبكه قطعه, النظام يفضل شغال و ما بيوقفش



مشاركة



حفظ

الكاش على أكثر من سيرفر

Distributes cache

- بدل ما كل مره تطلب البيانات من ال database (و ده بيأخذ وقت), بتخزن نسخه منها على سيرفرات اسرع.
- يقلل الحمل على ال database.
- يخلی البرنامج يستغل بكفاءة حتى لو عدد المستخدمين كبير.





جاھز تاں فٹ System Design interviews?

- ایہ اصعب سؤال واجھتے
فی interview قبل کدھ؟
- شارکنا تجربتک فی الکومونٹس!
- لو استفادت، اعمل لایک و شیر عشان
غیرک یستفید.



مشاركة



حفظ