

Cloud5.pdf



onafolch



Computació en Entorns al Núvol



3º Grado en Ingeniería de Datos



Escuela de Ingeniería
Universidad Autónoma de Barcelona

TELECOGRESCA



29 DE MARÇ 2025



"LA TELECOGRESCA ÉS EL MILLOR FESTIVAL UNIVERSITARI"
- SÒCRATES

LA RAÍZ

FIGA FLAWAS · LA FÚMIGA

MARIA JAUME · MAIG · LO PUTO CAT

DJ AKSIS · IRA DIGITAL · DJ MAKI

PARC DEL FÒRUM · BARCELONA



Silence BCN Norte: tu moto eléctrica desde 3200€



Hasta 133 km de autonomía y 100 km/h. Solicita el plan Move (-1.300€) y llévate 200€ en accesorios. Silence BCN Norte.

CLOUD

CLOUD DATABASES

1. CLOUD DATABASES

És el lloc principal per a les aplicacions per emmagatzemar, recuperar i organitzar dades. Tradicionalment tenies el teu propi servidor de bases de dades en una instància, però el cloud et permet crear i executar el teu propi servidor de bases de dades al núvol (inicies la instància, instal·les la BD, carregues les dades i et connectes a l'aplicació) i et proporciona serveis de BD gestionades, de tal manera que AWS s'encarrega de la instal·lació i el manteniment del programari de base de dades.

Hi ha les BD gestionades, que l'escalat, la tolerància a errors i la disponibilitat s'incorporen al servei, i les no gestionades, que l'escalat, la tolerància a errors i la disponibilitat ho gestiones tu.

El problema de les BD relacionals és que té límits de cada a l'escalabilitat, la seguretat de les dades, les còpies de seguretat i l'alta disponibilitat.

2. AMAZON RDS

És un servei gestionat que configura i opera una BD relacional al núvol. És una BD relacional de classe empresarial.

En les BD locals, t'encarregues tu de tot. Les bases de dades a EC2, amazon et proporciona una part, i en canvi, amb RDS o Amazon Aurora s'encarrega de tot menys l'optimització de l'aplicació.

Tindé la BD d'amazon RDS dintre una subxarxa privada en el meu VPC, de tal manera que els usuaris no hi podran accedir. S'utilitza per webs i aplicacions de mòbils, aplicacions del comerç electrònic i jocs.

Quan la meua aplicació necessiti transaccions o consultes complexes, quantitat de consultes d'escriptures mitjana o alta (fins 30.000 IOPS), no més d'un worker node i alta durabilitat utilitzaré RDS. Mentre que no l'utilitzare quan tingui moltes escriptures i lectures, fragmentació a causa de la gran mida de les dades, consultes simples de GET o PUT que pot gestionar una BD NoSQL o personalització del sistema de gestió de bases de dades relacionals.

Escalabilitat vertical

Passar a una instància més potent, amb més memòria, CPU i ample de banda.

Escalabilitat horitzontal (replicació)

Tenir una còpia de les dades en múltiples màquines connectades a través de la xarxa. Manté les dades geogràficament a prop dels usuaris (menys latència), el sistema pot funcionar si algunes peces han fallat (disponibilitat) i augmenta el nombre de màquines que poden atendre consultes de lectura (rendiment de lectura).

Si tinc dades replicades en diverses màquines, m'he d'assegurar que totes les dades estiguin igual. S'utilitza la replica basada en un leader. Una de les rèpliques serà el leader, que serà qui s'encarregarà de les escriptures. Les altres rèpliques seran els followers, que quan el leader escrigui dades, l'enviarà a tots els followers. Cada follower té un document (replication log) on té totes les operacions que ha de fer per tal de tenir totes les dades actualitzades. Els clients poden llegir tant de les rèpliques com del leader, però només poden escriure al leader.

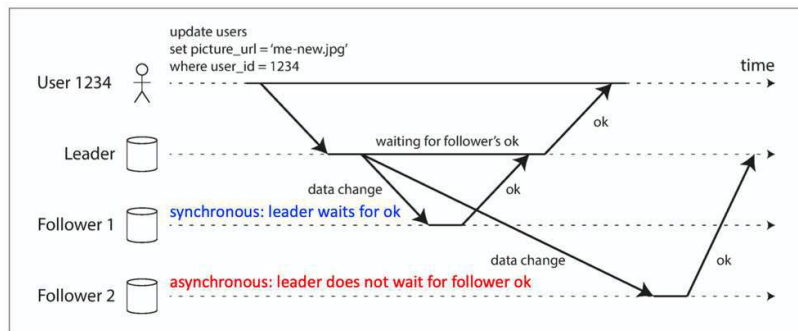
Tu moto eléctrica ideal en Sant Andreu: www.silencebcn-norte.com, 933128000



Ona Folch

WUOLAH

Hi ha la replicació síncrona i asíncrona:



Al ser síncrona, tots els followers tenen les dades al dia, però si per qualsevol raó el follower no respon, l'escriptura no es pot fer i el leader ha de para totes les escriptures pendents fins que la rèplica torni a estar disponible, la qual cosa fa parar el sistema i no és pràctic. Per aquesta raó, la majoria de sistemes utilitzen la replicació asíncrona.

Necessitem que l'aplicació/base de dades tingui una **alta disponibilitat**, és a dir, que segueixi funcionant si una màquina, o la xarxa o el datacenter cau.

Si una replica (follower) falla, quan es pugui es tornarà a aixecar la màquina. Quan es fagi, recuperarà el log, i obtindrà l'última transacció que es va processar abans que falles. Un cop tingui això, es comunicarà amb el leader i li demanarà les dades que s'han canviat després de l'última transacció, i aplicarà els canvis.

Quan cau el leader, s'escull un nou leader entre les replicues (normalment és la que està més al dia). Es reconfigura el sistema per tal que les escriptures es fagin al nou leader, i es configura el leader antic per ser una replica.

Tenir aquest sistema de replicació té certs problemes. Un és que si llegeixes la mateixa dada del leader i d'un dels followers, podria ser que obtinguis resultats diferents, ja que es basa en la consistència eventual. Pot ser un problema si el replication lag tarda minuts.

Si un usuari llegeix les dades que ha escrit, podria ser que al llegir-les ho fes d'un follower que encara no té les dades, i per tant que no obtingui el resultat esperat.

Per tenir una consistència de read-after-write, podries només permetre les lectures al leader, o que les escriptures recents només es llegeixin del leader. Es podria utilitzar un timestamp per llegir les rèpliques actualitzades.

Les rèpliques de només lectura serveixen per millor el rendiment de la BD. Serveixen per quan tenim una gran nombre de lectures o per informes mensuals amb consultes intensives.

Per tenir alta disponibilitat quan el master de la BD cau, podríem tenir una instància RDS addicional. La primària replica les dades a la segona, de la tal manera que si falla la primària, RDS automàticament passarà a la secundària en menys de 2 minuts, i les dades no es perdran. Aquesta secundària està a una zona diferent.

RDS pot fer manual o automàticament EBS snapshots de les instàncies, que es guarden en diverses zones de disponibilitat. Pot restaurar instàncies a partir dels snapshots, la qual cosa va bé per còpies de seguretat i testing.

Reconvery Point Objective (RPO): els snapshots em permeten restaurar instàncies d'una BD, és barat però lent. Si tinc les instàncies duplicades en diverses AZ, va bé per mantenir la BD funcionant si una instància falla, cosa que és ràpid, però car. Si puc tolerar perdre 1 hora les meves dades, utilitzaré snapshots, mentre que si puc tolerar fins a 5 minuts màxim, utilitzaré multi-AZ.

2. DYNAMO DB

Les bases de dades relacionals es guarden el files i columnes, i l'esquema és fixat, mentre que les no relacionals es guarden en documents, grafs o clau-valor, i l'esquema és dinàmic. L'escalabilitat de les relacionals és vertical i no relacional és horitzontal.

DynamoDB és un servei de base de dades NoSQL ràpid i flexible que només s'executa en SDD. Utilitza taules NoSQL, té emmagatzematge il·limitat, els ítems poden tenir atributs diferents, la latència és baixa i és escalable.

S'emmagatzema en ítems, que són similars a les files d'una BD relacional. Aquests ítems s'emmagatzemen en taules, i cada taula es guarda en una o més particions, les quals es repliquen en múltiples zones d'una regió.

Els ítems tenen una primary key, que és un valor únic de la taula, i els atributs són parells de clau-valor. Pots tenir la quantitat d'atributs que tu vulguis en un ítem, i es pot guardar fins un total de 400KB de dades per ítem.

Les particions són una assignació d'emmagatzematge per una taula, i des d'un principi hi ha un nombre inicial de particions per proporcionar el mínim rendiment. Es creen més particions (per clau primària), quan la partició existent s'omple al màxim i necessites més espai d'emmagatzematge. Aquesta gestió de particions és un procés automàtic en segon pla.

Quan faig una query, agafaré la clau primària i faré el seu hash, que em donarà un nombre enter entre el nombre de particions que tens, i buscaré el valor dintre aquella partició. Si jo vull escriure un ítem, faré el hash de la clau i em portarà a la partició on he d'escriure el valor (sense ordenar).

Fer consultes per clau és eficient, mentre que fer una scan dels atributs és lent, ja que s'han de llegir tots els valors de totes les particions (en paral·lel).

Consistència de lectura: l'aplicació escriu dades a una taula DynamoDB, i si rep un OK és que s'ha produït correctament l'escriptura i és duradora. Les dades tenen una consistència eventual en totes les ubicacions d'emmagatzematge de menys d'un segon.

Si es necessita strong consistency, es pot fer, però pot generar errors del servidor, i pot tenir una latència més elevada que la consistència eventual.