ISTITUTO NAZIONALE DI FISICA NUCLEARE

Direzione Sistemi Informativi

INFN-24-09-DSI
16 ottobre 2024

Gestione e operatività dei database per le nuove applicazioni della Direzione Sistemi Informativi

Stefano Bovina[1], Guido Guizzunti[1], Giuseppe Misurelli[1]

[1)] *INFN, Direzione Sistemi Informativi, I-00044 Frascati (Roma), Italy*

Abstract

La modernizzazione dei database, necessaria all'evoluzione di alcuni applicativi verso l'architettura a microservizi, è stata un momento di svolta per il servizio Database dell'Ufficio Sviluppo e Gestione Strategica dell'Infrastruttura. Una svolta tecnologica, organizzativa e di sostenibilità economica basata sulla possibilità di esplorare soluzioni Open Source con cui superare alcuni limiti del modello commerciale, come ad esempio l'indipendenza da una soluzione Oracle, e allo stesso tempo validare l'inclusione nella piattaforma, alla base di sviluppo e gestione dell'Infrastruttura della Direzione Sistemi Informativi (con i suoi servizi e le buone pratiche), di un ulteriore tassello con cui gestire i dati strutturati e non strutturati generati all'interno dell'ecosistema degli applicativi.
In questa nota interna viene descritto, evidenziandone architettura e operatività, come le due tecnologie di database pensate rispettivamente per gestire dati di tipo SQL e NoSQL siano state introdotte all'interno della piattaforma e integrate nel contesto di efficienza operativa della piattaforma stessa.

DOI n. 10.15161/oar.it/211818

Published by
Laboratori Nazionali di Frascati

1 Introduzione

L'introduzione di nuovi applicativi dell'Istituto, come "Booking", così come l'evoluzione di quelle esistenti, come "Concorsi", verso un'architettura a microservizi, ha reso necessario una profonda revisione delle modalità con cui la persistenza dei dati, le relazioni fra di essi e l'operatività dei vari database vengono forniti, creati e gestiti dal servizio Database dell'Ufficio Sviluppo e Gestione Strategica dell'Infrastruttura (nel seguito gruppo Sysinfo-Ops).

Nei fatti, la modernizzazione delle applicazioni ha altresì richiesto un ammodernamento dei database ed in particolare una migrazione verso tecnologie e soluzioni diverse rispetto a quella Oracle utilizzata per le cosiddette applicazioni legacy. Migrazione basata sulla volontà di passare da soluzioni commerciali ad altre tipicamente Open Source, utilizzate da una vasta comunità e riconosciute come standard de facto, che meglio supportano alcune delle metodologie e pratiche tipiche di un approccio DevOps, come l'efficienza operativa, elementi centrali della piattaforma alla base di sviluppo e gestione dell'Infrastruttura della Direzione Servizi Informativi (DSI) [1].

In questa nota interna viene descritto come il gruppo Sysinfo-Ops gestisce i due principali database (PostgreSQL [2] e MongoDB [3]) in uso dalle nuove applicazioni, dettagliandone l'operatività dal punto di vista di sicurezza, affidabilità, resilienza ed efficienza delle prestazioni.

2 Modernizzazione dei database della DSI

L'adozione dell'architettura a microservizi per alcuni dei sistemi gestionali usati dagli utenti dell'Istituto è stato un momento di riflessione sull'evoluzione delle modalità con cui i dati vengono gestiti, strutturati, acceduti e memorizzati all'interno di database relazionali (tipo SQL) e non relazionali (tipo NoSQL), da parte di tali servizi.

Una delle peculiarità proprie dei microservizi è quella che ogni servizio gestisce i propri dati in maniera esclusiva rispetto ad altri servizi della medesima applicazione. Il motivo di questo approccio è di garantire il disaccoppiamento tra i vari servizi al fine di evitare situazioni di condivisione degli stessi schemi di dati sottostanti fra servizi diversi, nascondendo di conseguenza i dettagli implementativi (es: database utilizzato, la progettazione dello schema dei dati) e l'accesso al dato grezzo in favore di interfacce di alto livello.

Tutto ciò, dal punto di vista del gruppo Sysinfo-Ops che fornisce il servizio Database, si è tradotto nello scouting di tecnologie e soluzioni SQL e NoSQL che potessero essere gestite con dinamiche DevOps (es. installazione e configurazione automatizzata, integrazione con scenari di build, deploy, delivery continuo) e che facilitassero la concretizzazione di una serie di principi di efficienza operativa come il supporto all'interazione programmatica attraverso API di tipo RESTful, il supporto a modalità di clustering in grado di realizzare scalabilità orizzontale e verticale, la tolleranza ai

fallimenti, la gestione dei backup, il monitoraggio e la replicazione geografica per assicurare la disponibilità delle applicazioni in scenari di disaster recovery.

Tecnologie e soluzioni da cercare nell'ambito Open Source, in opposizione a quello commerciale, privilegiando quelle utilizzate da una vasta comunità e riconosciute come standard de facto con l'intento di beneficiare di una conoscenza diffusa in fatto di risoluzione di problematiche e ottimizzazioni da apportare.

Soluzioni che sono state individuate in PostgreSQL e MongoDB rispettivamente come database SQL e NoSQL.

3 La soluzione PostgreSQL in DSI

PostgreSQL è un Database Management System (DBMS) che da 25 anni, nel panorama Open Source, evolve costantemente e che da qualche tempo a questa parte è diventato un riferimento in fatto di affidabilità e sicurezza dei dati, suite di funzionalità e di utilità disponibili tanto da diventare una sorta di data platform più che un database.

L'adozione di tale DBMS, in occasione della modernizzazione dei database di tipo relazionali, ha consentito al gruppo Sysinfo-Ops di realizzare una soluzione capace di offrire alta affidabilità, resilienza ai guasti e disaster recovery geografico attraverso la disponibilità di un cluster di produzione basato sull'architettura mostrata in figura 1.

Tale architettura consente di usare la clusterizzazione di PostgreSQL per avere scenari di alta affidabilità in cui il single point of failure viene eliminato attraverso la promozione, sia a livello intra cluster che a livello geografico, di un endpoint primario grazie alla replicazione dei dati dal primario (su cui sono consentite operazioni di scrittura e lettura) alle varie repliche (su cui sono consentite solo operazioni di lettura). La replicazione viene fatta in maniera asincrona con i seguenti scenari:

1. dal primario verso le repliche all'interno dello stesso cluster di Business Continuity;
2. dal primario su Business Continuity allo standby primario posto nel sito di Disaster Recovery;
3. dallo standby primario alle sue repliche nel sito di Disaster Recovery.

Replicazione e meccanismo di alta affidabilità realizzato utilizzando Patroni [4], tecnologia di high availability (HA) disponibile nell'ecosistema delle utilità di PostgreSQL che, avvalendosi di un file YAML di configurazione e di un gestore distribuito di tali configurazioni (nel nostro caso ETCD [5]), riesce a gestire scenari di fallimenti come i seguenti:

1. perdita di connettività di rete di una replica. Patroni non include più il nodo nella lista dei nodi a cui inviare una replica;
2. perdita di connettività di rete del primario. Patroni non include più il nodo nella lista dei nodi disponibili ed elegge una delle repliche a primario;

3. problemi al processo PostgreSQL nel primario. Patroni, che ha il controllo del processo nel nodo, lo riavvia. Dopo il quale, il nodo continuerà ad agire come primario.

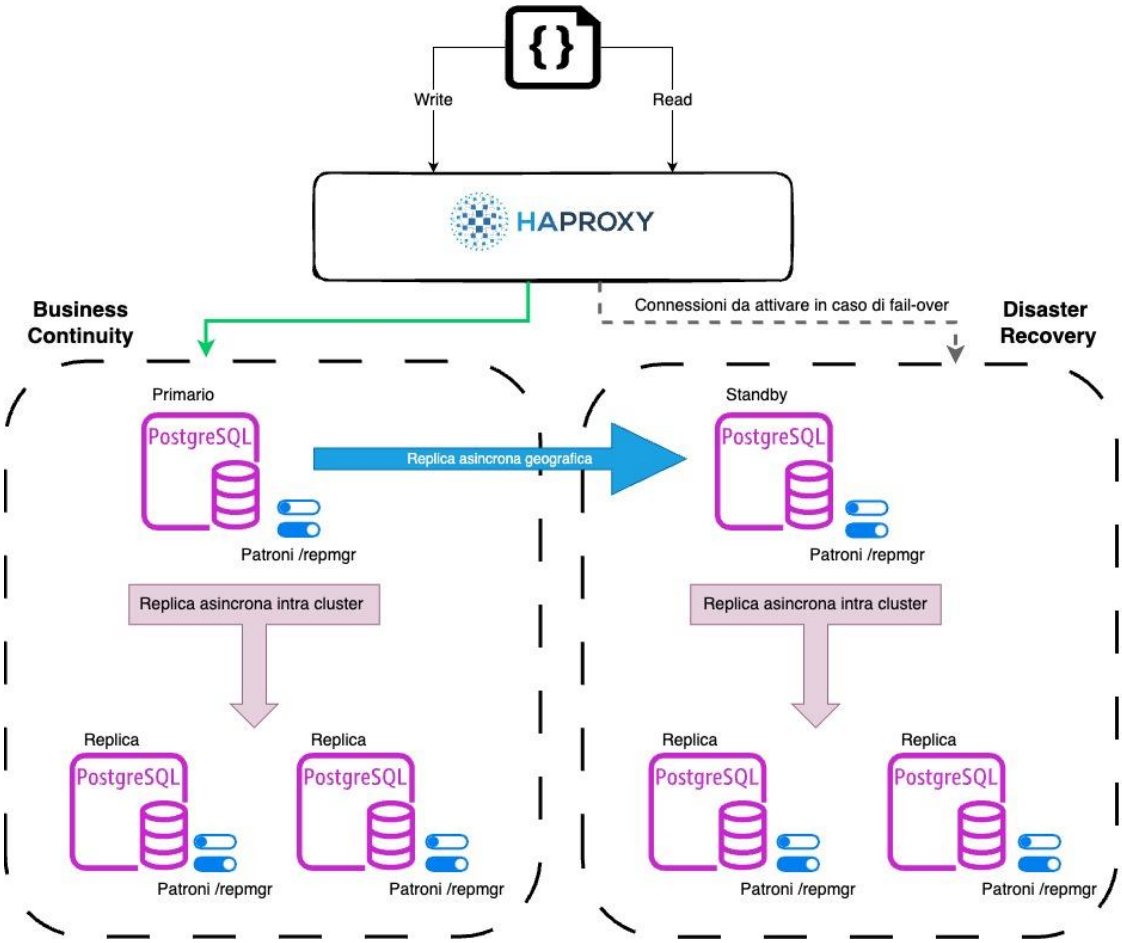

Figura 1: Architettura della soluzione PostgreSQL disponibile all'interno del servizio Database offerto dal gruppo Sysinfo-Ops.

In riferimento alle questioni operative, la gestione del servizio viene declinata nelle attività relative a: setup automatizzato, monitoraggio e analisi dei log, backup e ripristino e aggiornamenti software. Attività che sono nel seguito descritte.

3.1 Setup automatizzato

Il setup automatizzato si avvale del servizio del Provisioning e Gestione delle Configurazioni [1] per creare le macchine virtuali e configurarle come appartenenti ai diversi cluster predisposti per gli ambienti di test e produzione e nelle due infrastrutture di virtualizzazione di Business Continuity e Disaster Recovery.

Ogni macchina viene creata all'interno di un hostgroup Foreman in modo da ereditare tutta una serie di manifesti dichiarativi delle configurazioni Puppet comprensivi di tutti i parametri con i valori necessari nei diversi contesti operativi e funzionali (come mostrato in figura 2 per un cluster di produzione con la funzionalità di backup verso un sistema di object storage).

```
class {'sysinfo_postgresql':
    environment                => 'prod',
    cluster_id                 => 'cls1',
    replica_password           => '<resplica_pwd>',
    superuser_password         => '<su_pwd>',
    rewind_password            => '<re_pwd>',
    raft_encrypt_password      => '<raft_encrypt_password>',
    db_monitoring_pass         => '<db_monitoring_pass>',
    minio_ro_user              => '<minio_ro_user>',
    minio_ro_password          => '<minio_ro_pwd>',
    minio_rw_user              => '<minio_rw_user>',
    minio_rw_password          => '<minio_rw_pwd>',
    pgbackrest_repo_cipher_pass => '<<pgbackrest_repo_cipher_pass>',
}

class {'sysinfo_postgresql':
    context                    => 'backup',
    minio_ro_user              => '<minio_ro_user>',
    minio_ro_password          => '<minio_ro_pwd>',
    minio_rw_user              => '<minio_rw_user>',
    minio_rw_password          => '<minio_rw_pwd>',
    pgbackrest_repo_cipher_pass => '<pgbackrest_repo_cipher_pass>',
}
```

Figura 2: Esempio di manifesto di configurazione Puppet per un cluster di produzione con la funzionalità di backup verso un sistema di object storage.

In maniera del tutto simile, altri manifesti dichiarativi verranno usati, durante il setup dei cluster, per configurare tutta una serie di utilità come quelle di monitoraggio e analisi dei log. Tali configurazioni verranno inoltre mantenute nel tempo e ripristinate tutte le volte che qualcosa, come un intervento manuale direttamente sul server, agisca al di fuori del codice con cui vengono generati i manifesti Puppet.

3.2 Monitoraggio e analisi dei log

Tutti i database della DSI, unitamente ai vari cluster in cui vengono ospitati, recepiscono le configurazioni di monitoraggio e allarmistica, così come quelle per l'analisi dei log, rispettivamente dai servizi del gruppo Sysinfo-Ops "Monitoraggio e Allarmistica" e "Centralizzazione e Analisi dei Log".

In questo modo, per quel che riguarda il monitoraggio è possibile generare metriche, generiche di sistema e specifiche di database, aggregate in dashboard come quella mostrata in figura 3.

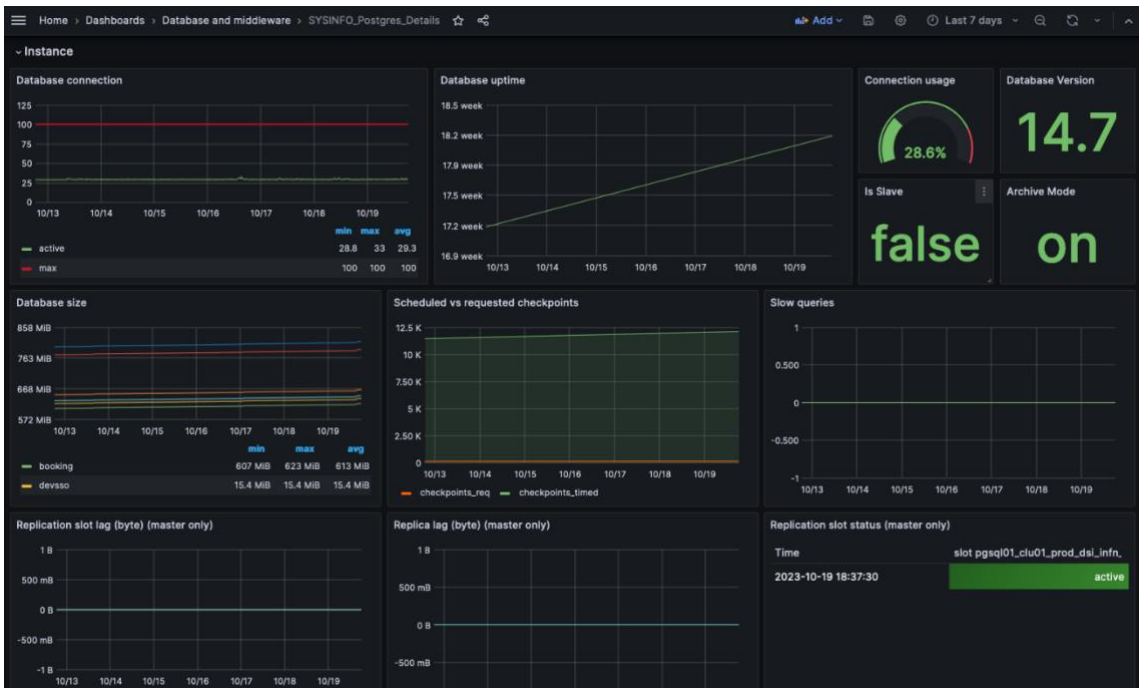

Figura 3: Dashboard di aggregazione per grafici e informazioni sulle metriche di database per un cluster PostgreSQL della DSI.

Unitamente a tali dashboard, utilizzate in fase di consultazione e analisi sugli andamenti storici e su possibili tendenze future, le metriche di monitoraggio vengono utilizzate per la generazione di allarmi relativi al superamento di soglie (es. check_replica_lag per avvisare quando il tempo di duplicazione dei dati del database da un nodo principale alla sua replica avviene con un ritardo superiore ai due minuti).

Per l'analisi dei log è possibile far riferimento, nelle fasi di indicizzazione e ricerca nel sistema di centralizzazione utilizzato, a metadati (es. `app_id:postgresql`, `environment:prod`) per, rispettivamente, aggregare e consultare i log in occasioni di risoluzione di problemi o quando, in fase di rilascio di un nuovo applicativo, si vuole da subito evidenziare un drastico aumento di messaggi di log relativi a possibili problemi.

Tali log vengono messi a disposizione alle parti interessate (es. personale dell'ufficio Servizio Sviluppo e Gestione Applicativi) che possono quindi consultarli attraverso delle ricerche da poter fare in tutta autonomia.

3.3 Backup e ripristino

Le funzionalità di backup e ripristino dei database PostgreSQL della DSI vengono realizzate avvalendosi dello strumento pgBackRest [6] che offre una serie di soluzioni (es. backup totale/incrementale, controlli d'integrità, interfacciamento con sistemi di object storage) utilizzate dal gruppo Sysinfo-Ops per garantire la conservazione dei dati ed il loro ripristino.

La strategia di backup adottata consiste nell'esecuzione di un backup totale (dati e registro transazioni) ogni sette giorni e di uno incrementale (dati e registro transazioni) una volta al giorno. Tale strategia consente il giusto bilanciamento di vari aspetti, come la quantità di dati da salvare, le tempistiche di backup (più lunghe per i backup di tipo totale) e quelle di ripristino (più veloci nel caso di backup incrementali purché i dati da recuperare siano relativi alle 24 ore precedenti all'ultimo backup incrementale).

L'archiviazione dei dati di backup viene effettuata nel sistema di object storage gestito dal gruppo Sysinfo-Ops, grazie al quale è possibile far eseguire a pgBackRest, debitamente autenticato e autorizzato, le tipiche operazioni di upload dei nuovi backup e di cancellazione di quelli più vecchi, in accordo con le policy di versionamento e conservazione degli oggetti. Facendo leva sulla funzionalità di object lock, l'archiviazione dei dati avviene adottando il modello Write Once Read Many (WORM) [7], garantendo che i dati di backup non vengano cambiati o rimossi dopo la loro scrittura nel sistema di storage, dove restano per 30 giorni. Inoltre, i backup vengono inoltrati su un sito di disaster recovery per garantire una maggiore resilienza e protezione dei dati in caso di disastro.

Lo stesso PMB è il principale vettore per il ripristino dei dati da un backup con un flusso d'interazione simile, ma contrario, a quello visto per il backup. PMB interroga il servizio di object storage per scaricare il backup da ripristinare e interagisce con il cluster PostgreSQL per il ripristino consistente di dati e metadati (es. indici).

3.4 Aggiornamenti software

Gli aggiornamenti dei cluster PostgreSQL vengono fatti in maniera periodica e ogni qual volta si rendano necessari, ad esempio per l'applicazione di patch di sicurezza o l'implementazione di nuova funzionalità. Si applicano sia al sistema operativo ospitante il database, sia al software PostgreSQL stesso.

In accordo con l'efficienza operativa perseguita dal gruppo Sysinfo-Ops, le operazioni di aggiornamento seguono procedure ben definite, che formalizzano tutte le attività preliminari (es. disabilitazione monitoraggio per evitare falsi positivi, adeguamento del codice di configurazione automatizzata), quelle in corso (es. configurazione automatizzata) e le verifiche finali per assicurare la completa operatività dei cluster dopo l'aggiornamento.

4 La soluzione MongoDB in DSI

Ci sono applicazioni INFN come Concorsi, PostOrdine e Prodotti che, in fase di ingegnerizzazione, sono state progettate per interagire con dati non strutturati, come per esempio documenti contenenti dati rappresentati sotto forma di collezioni di coppie chiave/valore. È proprio per gestire questo tipo di dati che il gruppo Sysinfo-Ops ha scelto di adottare MongoDB come DBMS non relazionale. Perché offre, tra l'altro, il supporto a documenti di tipo json [8], la giusta flessibilità nella modellizzazione dei dati non strutturati, l'indicizzazione dei documenti di tipo json, e una suite di utilità operative (es. replica dei dati, scalabilità orizzontale) che facilitano l'erogazione di un servizio in alta disponibilità e affidabilità.

Dal 2009 MongoDB, così come menzionato per PostgreSQL, ha continuato a evolversi costantemente, affermandosi ormai come un riferimento in fatto di affidabilità e sicurezza dei dati, grazie anche alla sua ricca suite di funzionalità e di utilità, tanto da diventare un elemento chiave per i microservizi che devono interagire con grandi quantità di dati non strutturati.

L'architettura della soluzione MongoDB nella DSI, rappresentata in figura 4, è stata pensata sulla base della clusterizzazione di tipo replica set [9] per garantire alta affidabilità e disponilità del servizio.

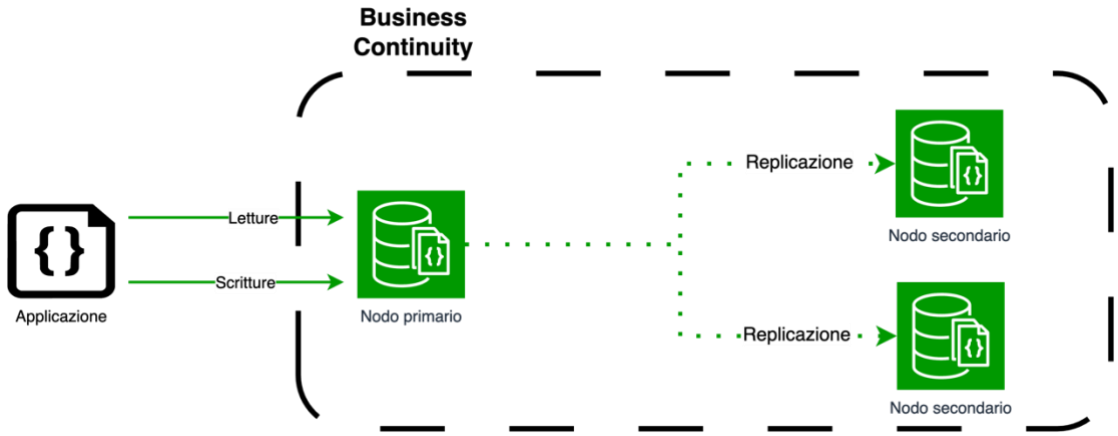

Figura 4: Cluster MongoDB in replica set ospitato all'interno del sito di Business Continuity.

Un replica set rappresenta un gruppo di più server MongoDB contenenti ciascuno un'esatta copia dei dati applicativi inseriti nel database. I nodi del cluster sono tre, nel sito di Business Continuity (CNAF-LNL), più uno di stand by, non partecipante attivamente al cluster, nel sito di Disaster Recovery (LNF). Focalizzandoci sui tre nodi di Business Continuity, di questi uno viene eletto come primario, ed è responsabile delle operazioni di lettura e scrittura, mentre gli altri due, denominati secondari, contengono le repliche dei dati.

Nel caso in cui il nodo primario non dovesse essere più disponibile, viene eletto a nuovo primario uno dei due secondari, come mostrato in figura 5, e il traffico proveniente dalle applicazioni viene rediretto verso quest'ultimo in maniera del tutto trasparente e senza alcun intervento manuale. Il vecchio primario, una volta ripristinato, viene automaticamente riunito al cluster con il ruolo di secondario.

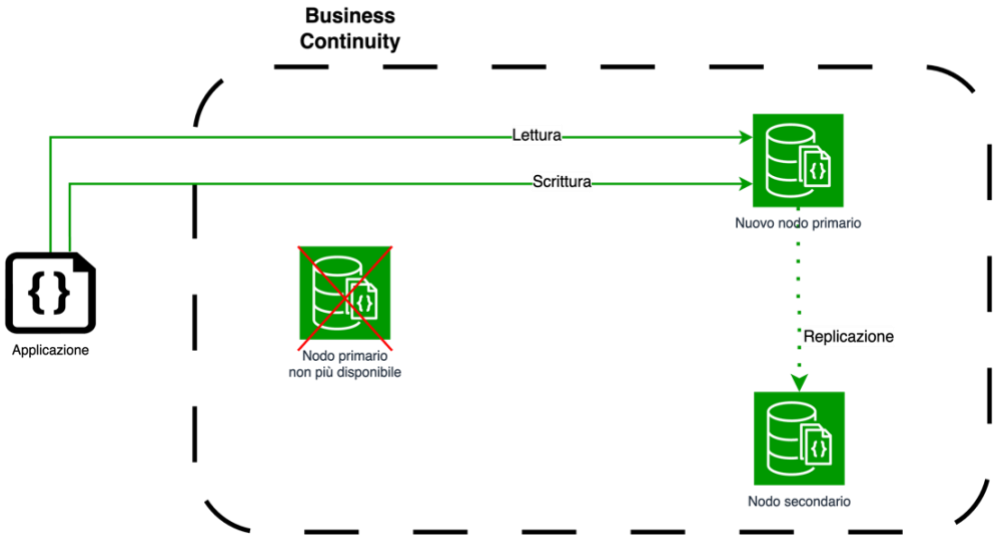

Figura 5: Cluster MongoDB in replica set adattivo al caso del nodo primario non più disponibile.

In riferimento alle questioni operative, analogamente a quanto fatto per la soluzione PostgreSQL, la gestione del servizio viene declinata nelle attività relative a: setup automatizzato, monitoraggio e analisi dei log, backup e ripristino dei dati, e aggiornamenti software. Attività che sono nel seguito descritte.

4.1 Set up automatizzato

Il provisioning delle machine virtuali e la loro configurazione come nodi di un cluster MongoDB ricadono, anche in questo caso, nel modus operandi proprio del servizio di Provisioning e Gestione delle Configurazioni.

In questo modo è possibile garantire controllo, coerenza e riproducibilità del set up attraverso una serie di manifesti dichiarativi Puppet che vengono applicati ai nodi di un cluster MongoDB utilizzando Foreman come "Puppet External Node Classifier" [10] capace quindi di indirizzare Puppet verso la giusta configurazione di un cluster, ad esempio di produzione, rispetto ad un altro.

A completare gli aspetti di set up si aggiungono inoltre una serie di procedure semi-automatiche eseguite dai membri del gruppo Sysinfo-Ops, in tutte quelle situazioni che vanno dal rinnovo dei certificati, al restart di un nodo per manutenzione straordinaria,

fino alla reinstallazione di un nodo del cluster, senza pregiudicare l'operatività generale del database.

4.2 Monitoraggio e analisi dei log

Così come per tutti i sistemi ed i servizi gestiti dal gruppo Sysinfo-Ops, il servizio di "Monitoraggio e Allarmistica" e "Centralizzazione e Analisi dei Log" fa sì che i cluster MongoDB vengano monitorati da metriche e parametri di sistema specifici per MongoDB, come mostrato nelle due figure 6 e 7.

Figura 6: Estratto di una dashboard rappresentante Opscounters per un nodo del cluster (numero delle diverse tipologie di operazioni nel tempo) oltre ad una serie di grafici sui contatori (per collection, oggetti e indici) e dimensioni utilizzate di storage, dati e indici.

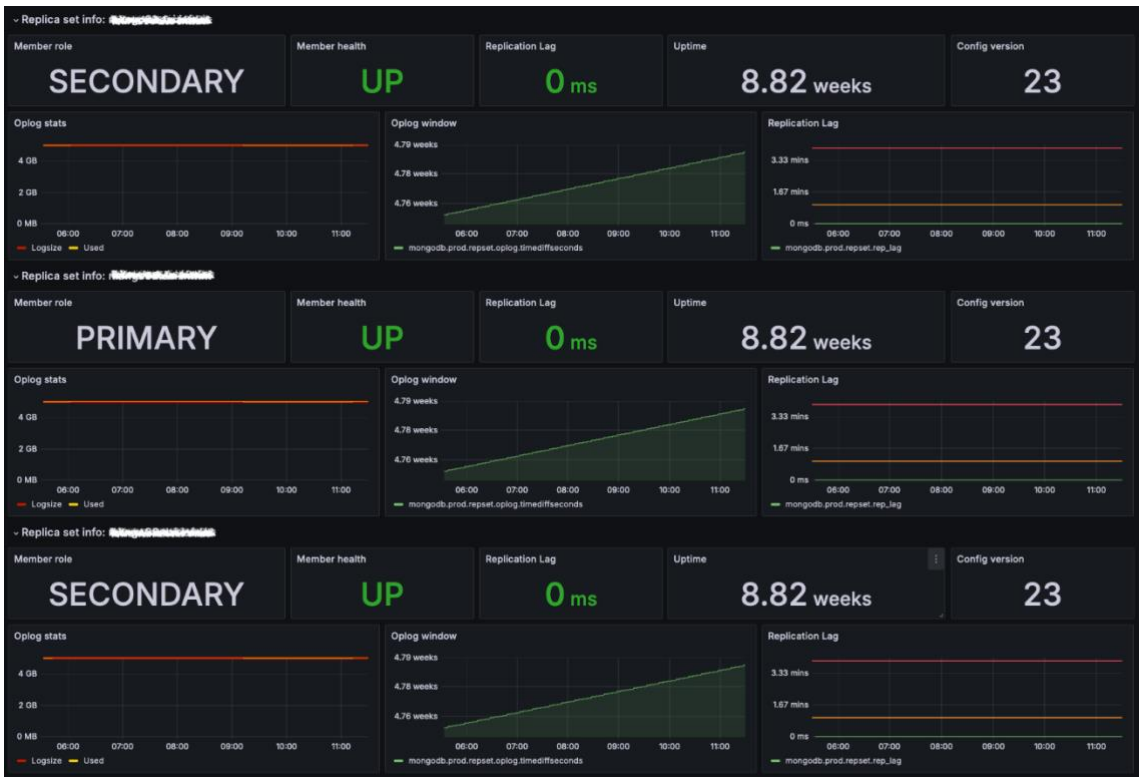

Figura 7: Colpo d'occhio sullo stato dei tre nodi di un cluster MongoDB. Per ognuno di essi vengono rappresentate informazioni cruciali come il ruolo (primario o secondario) e l'eventuale ritardo nella replica dei dati dal nodo primario ai secondari.

Per alcune di queste metriche, al superamento di determinate soglie, vengono attivati degli allarmi per notificare agli interessati situazioni di malfunzionamento o di rischio di indisponibilità.

L'analisi dei log dei cluster MongoDB viene demandata al servizio "Centralizzazione e Analisi dei Log". Analogamente al caso PostgreSQL, è possibile far riferimento, nelle fasi di indicizzazione e ricerca nel sistema di centralizzazione utilizzato, a metadati (es. `app_id:mongodb`, `environment:prod`) per rispettivamente aggregare e consultare i log, utile per esempio in occasione di risoluzione di problemi o quando, in fase di rilascio di un nuovo applicativo, si vuole da subito individuare un drastico aumento di messaggi di log relativi a potenziali problemi.

4.3 Backup e ripristino

Il backup dei dati viene effettuato avvalendosi dello strumento Percona Backup for MongoDB (PMB) [11], il quale, debitamente configurato, interagisce sia con i nodi del cluster per le operazioni di backup e ripristino, sia con il servizio di object storage per lo stoccaggio e la gestione dei dati di backup (flusso rappresentato in figura 9).

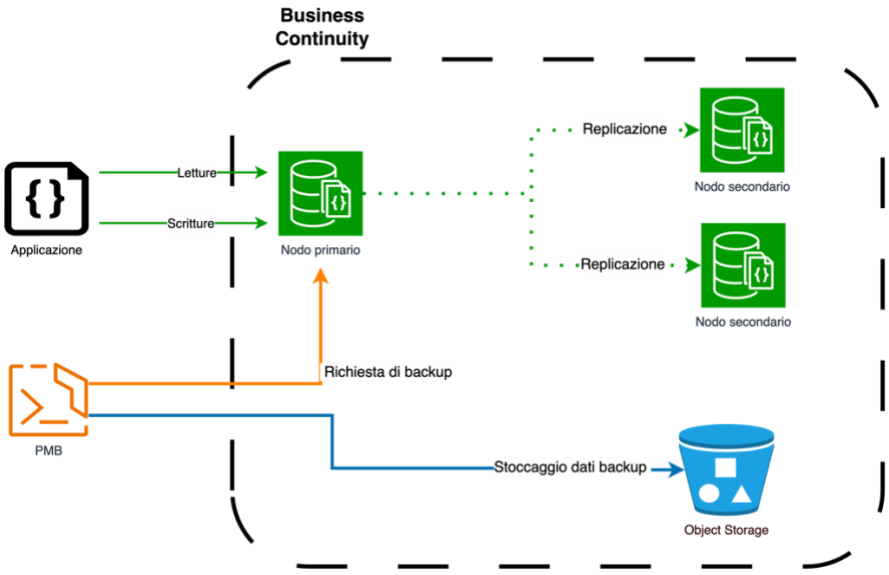

Figura 9: Funzionamento del backup di un cluster MongoDB. Il client PMB esegue una richiesta alla replica set connection string, ne controlla l'andamento e al termine invia i dati di backup verso il servizio di object storage per lo stoccaggio degli stessi. Il tutto avviene in maniera consistente senza dover fermare la normale funzionalità del cluster.

La strategia di backup si basa su un backup totale giornaliero (comprendente i dati e i log relativi alle operazioni su di essi) con un mantenimento di massimo trenta giorni rispetto all'ultimo backup effettuato. La cancellazione dei backup non più nella finestra di conservazione viene gestita attraverso PMB in accordo con le policy di versionamento e preservazione degli oggetti conservati nel servizio di object storage. Analogamente a quanto visto per PostgreSQL, anche in questo caso il modello di storage Write Once Read Many (WORM) evita che i dati di backup vengano modificati o rimossi dopo la loro scrittura nel sistema di storage.

Lo stesso PMB è il principale vettore per il ripristino dei dati da un backup, con un flusso d'interazione simile, ma contrario, a quello visto per il backup. PMB interroga il servizio di object storage per scaricare il backup da ripristinare e interagisce con il cluster MongoDB per il ripristino consistente di dati e metadati (es. indici).

Il ripristino può avvenire nei due contesti di Business Continuity (es. dati corrotti che necessitano di essere ripristinati) e di Disaster Recovery (es. quando il sito di Business Continuity non è più utilizzabile). In quest'ultimo caso, così come mostrato in figura 10, la movimentazione dei dati di backup viene delegata alla replicazione degli oggetti di object storage dal servizio in Business Continuity a quello in Disaster Recovery. In questo modo, PMB può richiedere e ottenere, a bassa latenza, i dati di backup e conseguentemente far partire le operazioni di ripristino del cluster MongoDB.

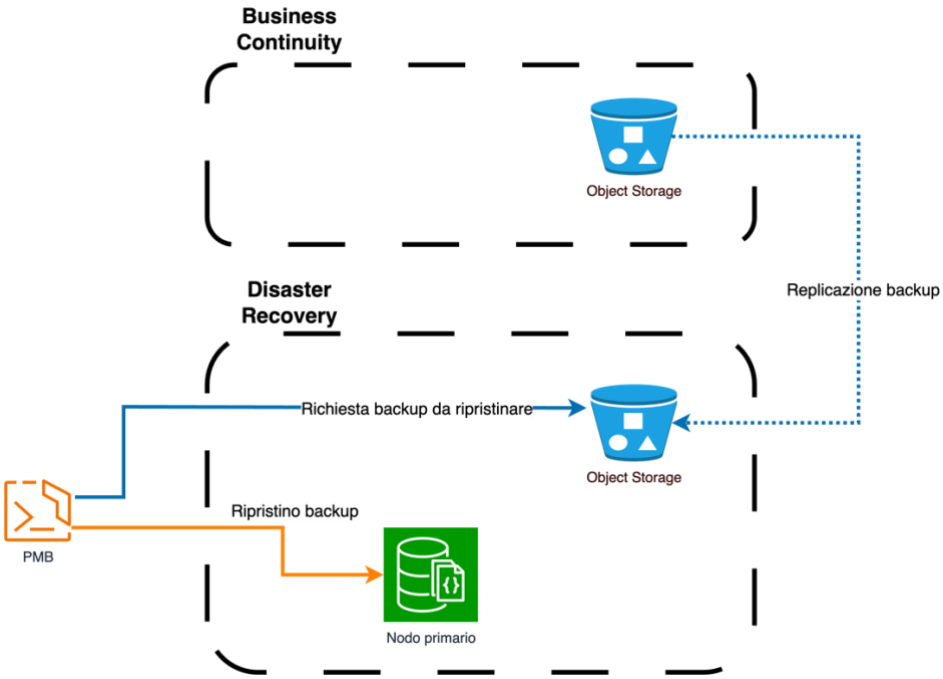

Figura 10: Flusso di ripristino di un cluster MongoDB nel contesto Disaster Recovery a partire dalla replicazione dei dati di backup dal contesto Business Continuity.

4.4 Aggiornamenti

Con cadenza simile a quanto visto per i cluster PostgreSQL, gli aggiornamenti dei cluster MongoDB vengono fatti sia in maniera periodica che all'occorrenza (es. patch di sicurezza, nuova funzionalità da utilizzare).

Anche per gli aggiornamenti di MongoDB, vengono seguite procedure semi-automatiche che includono l'analisi di fattibilità dell'aggiornamento, le modalità concrete di esecuzione e i test di validazione post-aggiornamento. Sebbene gli aggiornamenti dei cluster MongoDB vengano effettuati in continuità operativa, durante l'intero processo viene previsto, come misura del tutto precauzionale, il blocco di nuovi deployment, per garantire la massima stabilità al sistema.

5 Riferimenti

[1] La piattaforma alla base di sviluppo e gestione dell'Infrastruttura della Direzione Servizi Informativi

[2] https://www.postgresql.org/

[3] https://www.mongodb.com/

[4] https://patroni.readthedocs.io/

[5] https://etcd.io/

[6] https://pgbackrest.org/

[7] https://en.wikipedia.org/wiki/Write_once_read_many

[8] https://www.json.org

[9] https://www.mongodb.com/resources/products/capabilities/replication

[10] https://www.puppet.com/docs/puppet/7/puppet_index.html

[11] https://docs.percona.com/percona-backup-mongodb/index.html