Bortica Dumitru-Daniel
An IV, Grupa 1.1

Tema 2
Multi-tenancy

Multi-tenancy este o arhitectură în care o singură instanță a unei aplicații software servește mai multor clienți. Fiecare client este numit chiriaș. Chiriașii pot avea posibilitatea de a personaliza anumite părți ale aplicației, cum ar fi culoarea interfeței cu utilizatorul sau regulile de afaceri, dar nu pot personaliza codul aplicației.

În cloud computing, înțelesul arhitecturii multi-tenancy s-a mărit datorită noilor modele de servicii care beneficiază de virtualizare și acces la distanță. Multi-tenancy este atributul cheie al cloud-urilor publice si private și se aplică tuturor celor trei straturi ale cloud-ului: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) și Software-as-a-Service (SaaS). Un furnizor de software-as-a-service (SaaS), de exemplu, poate rula o instanță a aplicației sale pe o instanță a unei baze de date și oferă acces la web pentru mai mulți clienți. Într-un astfel de scenariu, datele fiecărui chiriaș sunt izolate și rămân invizibile pentru alți chiriași.

Majoritatea oamenilor se referă la stratul IaaS atunci când este vorba despre cloud. Chiar și așa, din punct de vedere arhitectural, cloud-urile publice, cât și cele private, depășesc trăsături tactice, cum ar fi virtualizarea, și se îndreaptă către implementarea conceptului de IT-as-a-Service (ITaaS) prin facturare - sau compensare în cazul unor cloud-uri private cu privire la utilizarea măsurată. Un IaaS are, de asemenea, o responsabilitate sporită prin utilizarea unor acorduri la nivel de serviciu (SLA), gestionarea identității pentru acces securizat, toleranța la erori, recuperarea în caz de distrugere, achizițiile dinamice și alte proprietăți cheie.

O arhitectură multi-tenancy oferă următoarele beneficii:

-Costuri mai reduse prin economii de scară: Cu ajutorul multi-tenancy, scalarea are mult mai puține implicații în infrastructură decât într-o soluție cu o arhitectură single-tenancy, deoarece noii utilizatori au acces la același software de bază.

-Infrastructura partajată conduce la costuri mai mici: SaaS permite companiilor de toate dimensiunile să împărtășească costurile operaționale ale infrastructurii și ale centrelor de date. Nu este nevoie de mai mult hardware sau instalare de noi aplicații. Companiile nu sunt nevoite să furnizeze sau să gestioneze nicio infrastructură sau software dincolo de resursele interne și astfel se pot concentra asupra sarcinilor zilnice.

-Întreținerea și actualizările în curs: Clienții nu trebuie să plătească taxe de întreținere costisitoare pentru a-și păstra software-ului actualizat. Furnizorii introduc noi funcționalități și actualizări, acestea fiind adesea incluse cu un abonament SaaS.

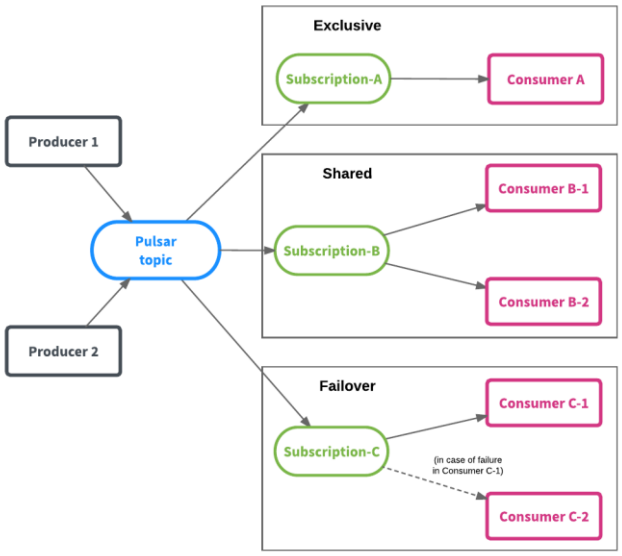
-Configurarea poate fi făcută fără a se modifica codul de bază: Soluțiile în cazul unei arhitecturi single-tanancy sunt adesea personalizate, necesitând modificări ale codului unei aplicații. Această particularizare poate fi costisitoare și poate face ca upgrade-urile să dureze mult timp, deoarece actualizarea ar putea să nu fie compatibilă cu noul mediu.

Soluțiile pentru multi-tenancy sunt proiectate pentru a fi foarte configurabile, astfel încât firmele să poată să își personalizeze aplicația. Nu există nici o schimbare a codului sau a structurii datelor, făcând procesul de actualizare ușor.

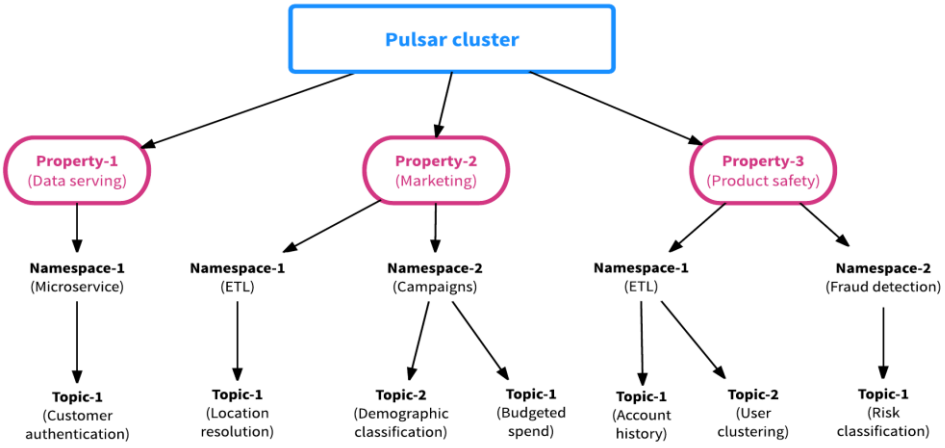
Apache Pulsar este un sistem de mesagerie distribuit open-source, creat inițial de Yahoo și care face parte acum din Apache Software Foundation. Este singurul sistem de mesagerie de tipul

publică-subscrie open-source disponibil care oferă multi-tenancy în totalitate. Pentru a înțelege modul în care Pulsar obține multi-tenancy, în continuare se prezintă modelul de mesagerie al Pulsar.

Ca și în multe alte sisteme publică-subscrie, o aplicație care alimentează date în Pulsar este numită producător, în timp ce o aplicație care consumă date de la Pulsar este un consumator. Aplicațiile pentru consumatori sunt, de asemenea, uneori denumite abonați. Urmând modelul publică-subscrie general, subiectul este construcția mesajului de bază în Pulsar. Un subiect reprezintă un canal în care producătorii adaugă date și din care consumatorii extrag datele. Un grup de consumatori formează un abonament pe un subiect. Diferite grupuri de consumatori își pot alege propriul mod preferat de a consuma mesaje din același subiect: exclusiv, partajat sau care nu au reușit. Diferitele moduri de abonare sunt prezentate în figura următoare.



Pulsar a fost construit de la început pentru a suporta multi-tenancy. Astfel, subiectele sunt organizate în cadrul a două resurse multi-tenancy specifice: proprietăți și namespace-uri. O proprietate reprezintă un chiriaș în sistem. Un chiriaș poate furniza mai multe namespace-uri în proprietatea sa. Și fiecare namespace poate conține apoi orice număr de subiecte. Namespace-ul este unitatea administrativă de bază pentru un chiriaș din Pulsar, pentru care se pot regla setările de replicare, gestiona geo-replicarea datelor în clustere, controla mesajele care expiră și efectua operații critice.



Primul pas pentru obținerea multi-tenancy este să se asigure că un chiriaș dat (a) nu are acces decât la subiectele pentru care are permisiuni și (b) nu are acces la subiectele pe care nu ar trebui să le vadă sau să le acceseze. Acest lucru se realizează printr-un mecanism de autentificare și autorizare.

În Pulsar, atunci când un client se conectează la un broker de mesaje, brokerul utilizează un plugin de autentificare pentru a stabili identitatea acestui client și apoi îi atribuie clientului un role token. Acest role token este un string, cum ar fi admin sau aplicația-1, care poate reprezenta un singur client sau mai mulți clienți. Role token-urile sunt folosite pentru a controla permisiunile clienților de a produce sau de a consuma anumite subiecte și de a administra configurația pentru proprietățile chiriașilor.

Pulsar suportă doi furnizori de autentificare: TLS Client Authentification și Athenz, un sistem de autentificare creat de Yahoo. De asemenea, clientul își poate implementa propriul furnizor de autentificare.

După ce role token-ul clientului este identificat de furnizorul de autentificare, brokerii Pulsar utilizează un furnizor de autorizație pentru a determina ce sunt autorizați să facă clienții. Autorizarea este administrată la nivel de proprietate, ceea ce înseamnă că pot fi mai multe scheme de autorizare active într-un singur cluster Pulsar. De exemplu, se poate creea o proprietate de shopping care să aibă un set de roluri și să se aplice unei aplicații de shopping folosită de o companie, în timp ce o proprietate de inventar ar fi utilizată numai de o aplicație de inventar. Permisiunile sunt gestionate la nivel de namespace, adică în proprietăți. Se pot acorda permisiuni anumitor roluri pentru listele de operații, cum ar fi producerea și consumul, într-un namespace.

În cele din urmă, autentificarea și autorizarea izolează chiriașii de accesarea subiectelor și efectuarea de acțiuni pentru care nu au permisiuni. Pe lângă izolarea de dragul securității, se așteaptă ca aplicațiile multi-tenancy să respecte acorduri la nivel de serviciu de către Pulsar, asigurând izolarea robusteții și performanțelor.

Arhitectura multi-tenancy este profitabilă din punct de vedere economic, deoarece costurile de dezvoltare software și de întreținere sunt împărțite. Aceasta poate fi comparată cu single-tenancy, o arhitectură în care fiecare client are propriul exemplu de software și poate primi acces la cod. Cu o arhitectură multi-tenancy, furnizorul trebuie să facă actualizări doar o singură dată. În cazul arhitecturii single-tenancy, furnizorul trebuie să modifice mai multe instanțe ale software-ului pentru a face actualizări.