

Multi-tenancy

Termenul “multi-tenancy” se referă la o arhitectură în care o singură instanță de software rulează pe un server și servește mai mulți chiriași. Un chiriaș este un grup de utilizatori care împărtășesc un acces comun cu privilegii specifice instanței software. Cu o arhitectură de acest fel, o aplicație software este concepută pentru a oferi fiecărui chiriaș o parte dedicată a instanței, inclusiv datele sale, configurația, gestionarea utilizatorilor, funcționalitatea individuală a chiriașului și proprietățile nefuncționale. Chiriasii pot avea posibilitatea de a personaliza anumite părți ale aplicației, cum ar fi culoarea interfeței cu utilizatorul (UI) sau regulile de afaceri , dar nu pot personaliza codul aplicației. Multi-tenancy contrastează cu arhitecturile cu mai multe instanțe, în care instanțele de software separate operează în numele unor chiriași diferiți.

Multi-tenancy permite economii de costuri în plus față de economiile de bază care pot fi obținute din consolidarea resurselor IT într-o singură operațiune. O instanță a unei aplicații generează de obicei o anumită cantitate de memorie și de procesare a cheltuielilor, care pot fi substanțiale atunci când sunt multiplicare de mulți clienți, în special dacă clienții sunt mici. Multi-tenancy reduce această cheltuială, amortizând-o pe mai mulți clienți. Economii de costuri suplimentare pot proveni din costurile de licențiere ale software-ului de bază (cum ar fi sistemele de operare și sistemele de gestionare a bazelor de date). În plus, dezvoltarea sistemelor multi-tenancy este mai complexă, iar testarea de securitate este mai strictă datorită faptului că datele multiple ale clienților sunt amestecate.

În cloud computing , înțelesul arhitecturii multi-tenancy s-a lărgit datorită noilor modele de servicii care beneficiază de virtualizare și acces la distanță . Un furnizor de software ca un serviciu (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.

Într-un cloud privat, clienții, care sunt numiți și chiriași, pot fi divizii de afaceri diferite în cadrul aceleiași companii.

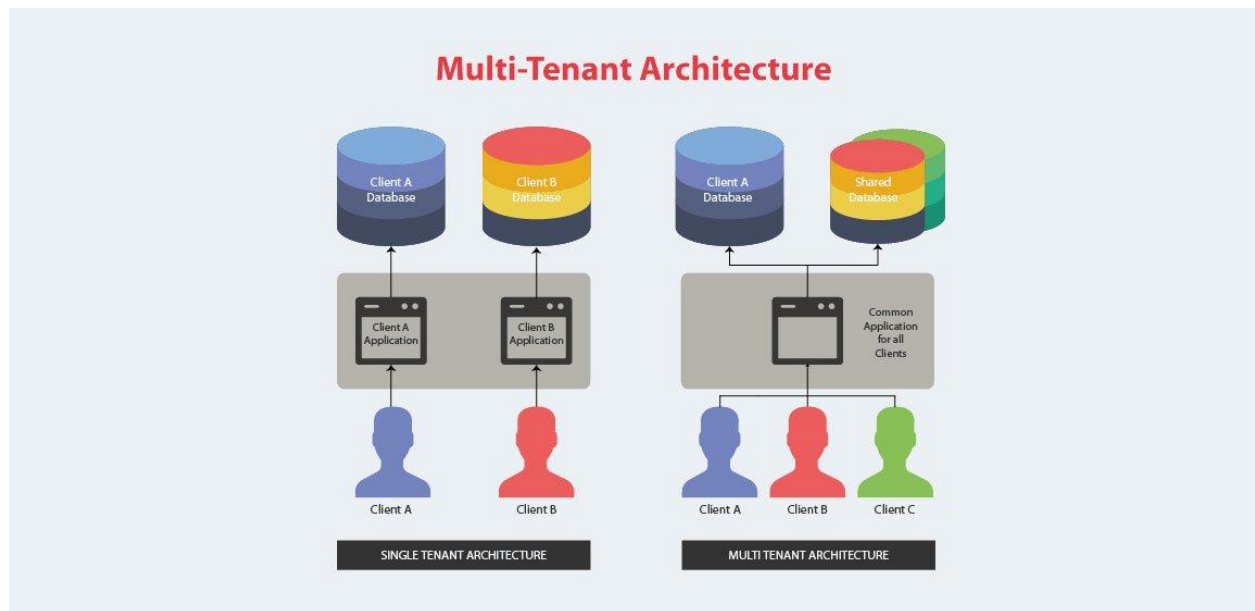
Într-un cloud public, clienții sunt deseori complet organizați. Majoritatea furnizorilor de cloud-uri publice folosesc modelul multi-tenancy. Acesta le permite să ruleze o instanță de server, care este mai puțin costisitoare și facilitează implementarea actualizărilor pentru un număr mare de clienți.

Mai departe aș vrea să fac o analiză comparativă a arhitecturilor de tip single-tenancy și multi-tenancy în aplicațiile SaaS.

Single-tenancy – o singură instanță a software-ului și a întregii infrastructuri de asistență servesc unui singur client. Cu o singură locație, fiecare client are o bază de date independentă și o instanță a software-ului. Cu această opțiune, în esență, nu există nicio împărțire.

Multi-tenancy - o singură instanță a software-ului și a întregii infrastructuri de asistență servește mai multor clienți. Fiecare client împarte aplicația software și împărtășește, de asemenea, o singură bază de date. Datele sunt etichetate în baza de date ca aparținând unui client sau altui client, iar software-ul este suficient de inteligent pentru a ști care date îi aparțin.

Din punct de vedere al utilizării resurselor, scopul utilizării optime a resurselor este mult mai mare într-o aplicație multi-tenancy în comparație cu cea aplicată într-o aplicație single-tenancy. Având în vedere că mai mulți chiriași utilizează aceeași infrastructură și resurse, utilizarea se obține automat în mod optim și menținerea ritmului de întreținere. De asemenea, dacă la un moment dat, un chiriaș nu utilizează resurse specifice, acesta poate fi utilizat de un alt chiriaș. Aceasta face ca utilitatea tuturor resurselor să fie proporțională între ele. Pentru a mă face mai bine înțeles, voi lăsa mai jos o imagine



O arhitectură multi-tenancy aduce beneficii pe termen lung furnizorilor de aplicații SaaS atât din punct de vedere al întreținerii, dezvoltării și investițiilor produselor. Cu toate acestea, nici nu este o operațiune foarte ușoară. Aceasta implică câteva provocări pe care trebuie să le identificați și să le câștigați pe termen lung. Unele provocări includ sarcina crescută a eforturilor de dezvoltare și confidențialitatea datelor. Aceste provocări pot fi depășite prin identificarea datelor, crearea unei arhitecturi de baze de date și proiectarea unei arhitecturi aprofundate a aplicațiilor.