Budapesti Műszaki és Gazdaságtudományi Egyetem
Méréstechnika és Információs Rendszerek Tanszék
Hibatűtő Rendszerek Kutatócsoport

# Finom szemcsézettségű hozzáférésvédelem Git tárolókhoz

Koltai Kadosa, BSc. Mérnökinformatikus szakos hallgató
Konzulens: Bergmann Gábor adjunktus
Rendszertervezés specializáció
Témalabor beszámoló
2018/19 1. félév

# Alapfogalmak

Git alapvetően egy verziókezelő szoftver. Pillanatképeket tárol a verziókezelt fájlokról, és lehetőséget biztosít arra, hogy ezeknek a pillanatképeknek megfelelő formába állítsuk vissza a verziókezet fájlokat. Ezt checkoutolásnak, a pillanatképet commitnak hívjuk.
Git képes ezen commitok közötti különbségeket felfedni, fájlnál kisebb méretű különbségeknél is.
A mappát amiben Git a commitokat tárolja a tárolónak, vagy repónak hivjuk. Lehetséges saját olyan saját repót létrehozni, ami hozzá van kötve egy másikhoz. Ezt clone-ozásnak hívjuk. A saját repó frissítését, tehát az eredeti tepóba újonnan bekerült információk fissítísét fetch-elésnek hívjuk.
Git nyújt lehetőséget arra, hogy személyre szabjuk. A munkafolyamata különböző lépcsőinél lefuttat általunk testreszabható függvényeket, amiket hookoknak hív.

# Szerver Implementációk

Több megoldás létezik arra, hogy a Git repóinkat megosszuk.
A legegyszerűbb az alap Git implementáció. Ezzel az a probléma, hogy nem különbözteti meg a felhasználókat. Akinek jogosultsága van elérni a unix felhasználót, amiben a repók vannak, mindegyiket tudja olvasni.
Elterjedt a GitLab, ami körbezárja az alap Gitet, és további funkciókat szolgáltat. Bár különbséget tesz felhasználó között, és ezeket a felhasználók jogosultságát korlátozza, de csak repo szintre, és a felhasználó kilétét nem adja át a hookoknak, így mi sem változtathatunk ezen.
Ezen felül létezik a Gitolite, ami szintén körbezárja az alap Gitet és további funkciókat szolgáltat. Vannak felhasználói, és kilétüket biztosítja számunkra a saját hook rendszerében, amit triggereknek hív.

Ebből egyszerűen látszik, hogy saját megoldásom lényeges része Gitolite triggerek testreszabásában fog megjelenni.

# Saját proof-of-concept létrehozása

Mivel a Gitolite csak repo szintű hozzáférésvédelmet nyújt, minden felhasználónak készítünk egy saját repót, valamint egy mester repót. Gitolite megoldja, hogy minden felhasználó csak a saját repóját tudja olvasni, ezért a mi munkánk már csak az, hogy minden repóban mindig a megfelelő információ legyen.
Minden nem-mester repóhoz tartozik egy trigger (ACCESS_1), ami akkor fut le, amikor a felhasználó feltöltene információt, és Gitolite már engedélyezte a felhasználó hozzáférését ehhez a repóhoz, de a változtatásokat Git még nem hajtotta végre. Ekkor ellenőrizzük, hogy a felhasználó csak olyat szerkesztett, amire jogosultsága volt. Ha minden rendben, akkor tovább engedjük.
Minden nem-mester repóhoz tartozik egy trigger (POST_GIT), ami akkor fut le, amikor Git új információt ad a repóhoz. Először ellenőrizzük, hogy ez az információ a mester repóból jött-e. Ha igen, nincs feladatunk, de ha nem akkor ezt az információt továbbítani kell a mester repóba. Checkoutoljuk a mester repót és ezt a repót, és átvisszük az információt ebből a mester repóba. A commit üzenet végére odaírjuk, melyik repóból jött az információ.
A mester repóhoz tartozik egy trigger (POST_GIT), ami akkor fut le, amikor Git új információt ad hozzá. Megnézzük a commit üzenetet, hogy honnan jött ez az információ. Ha nincs ilyen információ, akkor közvetlenül ide érkezett. Checkoutoljuk a mester repót . Az összes nem-mester repót, kivéve az információ eredeti helyét, ha van, egyesével checkoutoljuk (továbbiakba „az”). A mester repó tartalmát, ahhoz a repóhoz tartozó felhasználó jogosultsági alapján, átvisszük oda. A commit üzenet végére odaírjuk, hogy a mester repóból jött az információ.

# Kitekintés

Jelenleg azt, hogy hozzáadunk egy felhasználót:

- Gitolite config fájlba hozzáadjuk a felhasználót
- ACCESS_1 triggerbe beírjuk az írási jogosultságait
- POST_GIT triggerbe beírjuk az olvasási jogosultságait

Lépésekkel érjük el, ami sajnos nem automatikus. Ahhoz, hogy ez hasznos implementációja legyen a témának ezt a felhasználónak (szerver admin) deklaratívan kell, hogy tudja megadni.