

Subversion Sürüm Kontrol Sistemi Kurulumu ve Apache Yapılandırması

Çağrı Emer

Ağustos, 2011

İçindekiler

1	Giriş	2
2	Peki ne yapar bu Subversion?	3
3	Subversion benim için doğru araç mı?	4
4	Subversion Kurulumu	5
5	Peki bu kullanıcı izinlerini nasıl oluşturacağız?	7
6	Kullanıcılar tamam da, ya belli kullanıcıların belli dosyalara erişmesini istiyorsak ne yapacağız?	8
7	Notlar ve Kaynaklar	9

1 Giriş

2000 yılında CollabNet, Inc. tarafından başlatılan açık kaynak Subversion projesi, geçtiğimiz on yıl içerisinde büyük başarılarla imza atarak neredeyse piyasa standardı haline gelen bir sürüm kontrol sistemidir. Yakın geçmişte Apache Software Foundation çatısı altına girmiş olan Subversion, bir Apache projesi olarak hayatına devam etmektedir.

2 Peki ne yapar bu Subversion?

Subversion, zaman içinde dosya ve dizinlerinizde yaptığınız değişiklikleri takip eden ve gerektiğinde bir dosyanın ya da dizinin belli bir tarihteki versiyonuna ulaşmanıza imkan veren yazılımdır. Böylelikle verilerinizin zaman içinde nasıl değiştiğini görebilir, istenmeyen değişiklikleri ekstra bir iş yüküne gerek duymaksızın geri alabilirsiniz. Bu bağlamda Subversion, kullanıcı kılavuzunda da geçtiği üzere tabiri caizse bir zaman makinasıdır.

Ağ üzerinden çalışabildiği için, değişik bilgisayarlardaki insanların dosyalara erişebilmesine ve dosyaları düzenleyebilmesine olanak verir. Üstelik bunu yaparken birinin dosyayı bozma ihtimalinden korkmanız gerekmez çünkü yapılan değişiklikleri her zaman geri alabilirsiniz. Bunun değeri, çok kişinin aynı proje üzerinde çalışması gerektiği anlarda daha rahat olarak anlaşılacaktır.

3 Subversion benim için doğru araç mı?

Subversion kullanmayı düşünen bir kullanıcı ya da sistem yöneticisiyseniz kendinize sormanız gereken ilk soru Subversion'ın sizin kullanımınız için doğru bir araç olup olmadığıdır. Yine kullanıcı kılavuzundan aktaracak olursak, "Subversion bir çekiçtir, fakat her problemi çivi gibi görmemeye özen gösterin."

Eğer dosyaların ve izinlerin eski hallerini arşivlemek, muhtemelen bu hallere geri dönmek ve verinizin zaman içinde nasıl değiştiğini gösteren kayıtları incelemek istiyorsanız Subversion tam sizin aradığınız araçtır. Eğer çeşitli dosyalar üzerinde, genellikle ağ üzerinden, diğer insanlarla birlikte çalışmak durumundaysanız ve yapılan değişiklikleri kimin yaptığını görmek istiyorsanız Subversion kullanımı yine uygun olacaktır. Subversion'ın yazılım geliştirme ortamlarında kullanılmasının en büyük sebebi de budur.

Tabii ki Subversion kullanmanın bir de maliyeti vardır. Verilerinizi ve tarih içindeki hallerini saklamak için bir depo yönetmeniz ve bu deponun zaman içinde yedeklerini almanız gerekir. Dosyalarınızla çalışırken bildiğimiz kopyalama, taşıma, yeniden adlandırma ve silme işlemleri yerine bu işlemlerin Subversion'a özgü olanlarını kullanmanız gerekir.

Bütün bu ekstra iş yüküne -ki doğru araçları kullanırsanız zamanla aslında iş yükü olmadığını göreceksiniz- razıysanız, yine de Subversion'ı diğer araçların daha iyi şekilde çözebileceği bir problem için kullanmamalısınız. Örneğin Subversion, versiyonu değişmeyen fakat zaman içinde büyüyen verilerinizi dağıtabileceğiniz bir sistem değildir. Bu şekilde kullanabilirsiniz fakat bu işi çok daha iyi yapan araçlar varken Subversion kullanmak gereksiz olacaktır. Bu yazıda dav modülü ile Apache üzerinden kullanılacak şekilde Subversion yapılandırılması anlatılmıştır.

4 Subversion Kurulumu

İlk olarak aşağıdaki komutla Subversion ve Apache için gerekli olan Subversion modüllerini kurmalısınız.

```
1 sudo apt-get install subversion libapache2-svn
```

Şimdi deponuzu tutacağınız dizini yaratıyoruz. Dizin seçimi size kalmakla birlikte, genellikle /srv ya da /var altında tutulduğunu söylersek yanlış olmayacaktır. Bu yazıda ben, /srv/svn dizinini depo olarak kullanacağım.

```
1 sudo mkdir -p /srv/svn
```

Dizini oluşturduktan sonra ilk depoyu yaratmaya hazırsınız. Şu komutla devam edelim.

```
1 svnadmin create /srv/svn
```

Tebrikler ilk svn deponuzu oluşturmuş oldunuz. Şimdi deponuzu Apache üzerinden sunmak için yapmanız gereken temel ayarlara bakalım. Diyelim ki http://svn.ornek.com sitesi üzerinden bu işlemleri gerçekleştireceksiniz. örnek.com'un yapılandırma dosyasında svn'i bir VirtualHost olarak barındırdığınız varsayıldığında, yapılandırma dosyanız içine eklemeniz gereken kod şuna benzer olacaktır.

```
1 <VirtualHost IP:Port>
2     ServerAdmin yoneticisi@ornek.com
3     ServerName svn.ornek.com
4     DocumentRoot /var/www/ornek.com/svn.ornek.com/public_html
5     <Location /svn>
6         DAV svn
7         SVNPath /srv/svn
8     </Location>
9     CustomLog /var/www/ornek.com/svn.ornek.com/logs/svn_logfile "%t %u %{SVN-ACTION}e" env=SVN-ACTION
10 </VirtualHost>
```

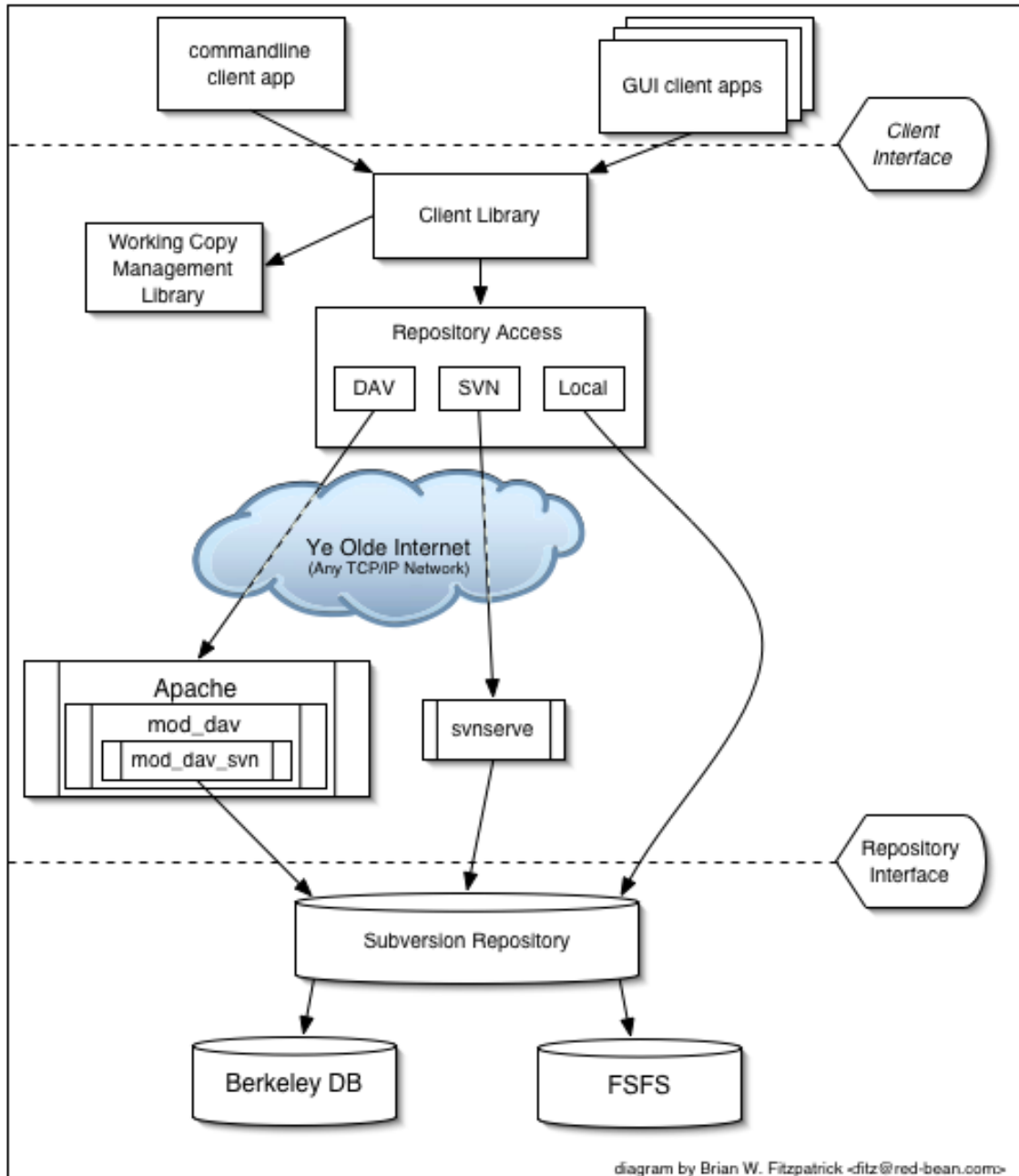
Apache'yi yeniden başlattığınızda http://svn.ornek.com adresi üzerinden deponuza erişebilirsiniz. Lakin şu anda içinde herhangi bir şey olmadığı için boş olarak gözükecektir. Bu yapılandırmada bir erişim izni ya da erişim kontrolü oluşturmadık. Bunlar daha üst düzeyde ayarlar olduğundan ve hepsini anlatmak mümkün olmadığından çok temel iki tanesi ile devam edeceğim. Dahası için mutlaka kullanıcı kılavuzunu okumalısınız.

Yukarıda oluşturduğumuz depo su anda herkese açık. svn.ornek.com adresine erişimi olan herkes depomuza da erişebilir. Dolayısıyla eğer sadece istediğimiz kişilerin erişmesini istiyorsak, erişim izinleri oluşturmalıyız.

```
1 AuthType Basic
2 AuthName "Kimsin Sen?"
3 AuthUserFile /etc/svn-auth-file
4 Require valid-user
```

Örnek.com sitenizin yapılandırma dosyasına ekleyeceğimiz yukarıdaki satırlar tam olarak istediğimiz işi yaparlar. Yani yeni VirtualHost'umuz aşağıdaki gibi gözükyorsa, svn.ornek.com adresine gelen kullanıcıların artık kullanıcı adı ve şifre girmeleri gerekecektir.

```
1 <VirtualHost IP:Port>
2     ServerAdmin yoneticisi@ornek.com
3     ServerName svn.ornek.com
4     DocumentRoot /var/www/ornek.com/svn.ornek.com/public_html
5     <Location /svn>
6         DAV svn
7         SVNPath /srv/svn
8         AuthType Basic
9         AuthName "Kimsin Sen?"
10        AuthUserFile /etc/svn-auth-file
11        Require valid-user
12    </Location>
13    CustomLog /var/www/ornek.com/svn.ornek.com/logs/svn_logfile "%t %u %{SVN-ACTION}e" env=SVN-ACTION
14 </VirtualHost>
```



Şekil 1:

5 Peki bu kullanıcı izinlerini nasıl oluşturacağız?

Neyse ki Apache bunu bizim için çok kolaylaştırmış. İlk seferde yapmamız gereken;

```
1 AuthUserFile /etc/svn-auth-file
```

satırında geçen /etc/svn-auth-file adresine izin dosyamızı oluşturmak. Bu dosya adresini kendi ihtiyaçlarınıza göre değiştirebilirsiniz. Yani /srv/permissions/svn-perm gibi bir dosyada da tutmanız herhangi bir sorun yaratmayacaktır. Unutmamanız gereken izinleri yarattığınız dosyanın adresini AuthUserFile değişkenine parametre olarak atamak. Şimdi ilk kullanıcıyı oluşturalım.

```
1 htpasswd -cs /etc/svn-auth-file yonetici
```

Bu komutla /etc dizini altına svn-auth-file diye bir dosya yarattık. Yönetici isminde bir kullanıcının şifresini SHA algoritmasıyla şifreledik ve bu dosyaya kaydettik. Bundan sonra oluşturacağımız kullanıcılarda artık create manasına gelen -c seçeneğini kullanmamıza gerek yok. Bir de böyle bir kullanıcı tanımlayalım.

```
1 htpasswd -s /etc/svn-auth-file kullanıcı1
```

-s seçeneğini kullanmazsak şifrelerimiz düz metinler halinde saklanır ki hiç hoş bir şey değildir. SHA algoritması dışında hangi algoritmaları kullanabileceğinizi görmek için htpasswd'nin man sayfalarına bakabilirsiniz.

6 Kullanıcılar tamam da, ya belli kullanıcıların belli dosyalara erişmesini istiyorsak ne yapacağız?

Bunun için erişim kontrol dosyası oluşturmalı ve yine örnek.com'un yapılandırma dosyasına birkaç satır eklemeliyiz. Erişim kontrol dosyası çok basit bir yapıya sahip. Diyelim yukarıda oluşturduğumuz yönetici isimli kullanıcının her dizine erişebilmesini fakat kullanıcı1 isimli kullanıcının sadece kullanıcı1 dizinine yazabilmesini fakat diğer dizinleri okuyabilmesini istiyoruz. O halde şöyle bir dosya oluşturmalıyız. Yine erişim izin dosyasında olduğu gibi bu dosyayı istediğimiz isimde ve istediğimiz dizinde oluşturabiliriz. Yeter ki adresini bilelim.

```
1 nano /etc/svn-access-file #favori metin editörü tartışmasına girmeyelim :) ben de istemez miyim bir vi
   kullanayım, emacs'le macro'dan macro'ya uçayım.
```

Bu dosyaya şu satırları yazacağız.

```
1 [/]
2 yönetici = rw
3 kullanıcı1 = r
4
5 [/kullanıcı1]
6 kullanıcı1 = rw
```

Tabii ki bu dosyaya ekleyeceğimiz satırlar kullanıcılarınıza ve deponuzun dizin yapısına göre değişecektir. Birden çok depo kullanmanız halinde ya da Apache yapılandırmasında SVNParentPath kullandıysanız bu gibi sebeplerden farklı bir yapılandırmaya ihtiyaç duyabilirsiniz. Bu noktada sizi tekrar kullanıcı kılavuzuna yönlendiriyorum. Şimdi erişim kontrolü için gereken değişkeni Örnek.com'un Apache yapılandırmasına ekleyelim.

```
1 <VirtualHost IP:Port>
2     ServerAdmin yönetici@ornek.com
3     ServerName svn.ornek.com
4     DocumentRoot /var/www/ornek.com/svn.ornek.com/public_html
5     <Location /svn>
6         DAV svn
7         SVNPath /srv/svn
8         AuthzSVNAccessFile /etc/svn-access-file
9         AuthType Basic
10        AuthName "Kimsin Sen?"
11        AuthUserFile /etc/svn-auth-file
12        Require valid-user
13    </Location>
14    CustomLog /var/www/ornek.com/svn.ornek.com/logs/svn_logfile "%t %u %{SVN-ACTION}e" env=SVN-ACTION
15 </VirtualHost>
```

Apache'yi yeniden başlattığımızda hem izin kontrolü hem de erişim kontrolü yapan ve Apache üzerinden çalışan bir Subversion sunucuya sahip olacağız. Kendimizi tebrik edebiliriz :)

Temel Subversion kullanımı ve Git ile arasındaki farkları gelecek ayki SUDO'da bulabilirsiniz.

7 Notlar ve Kaynaklar

VirtualHost yapılandırması için kullandığım CustomLog'un nasıl oluşturulduğunu Apache sayfalarında bulabilirsiniz. \ http://httpd.apache.org/docs/2.2/mod/mod_log_config.html#customlog

Diğer her şey için bakmanız gereken kaynak kullanıcı kılavuzudur. \ <http://svnbook.red-bean.com/>