

# RASPBERRY PI KULLANARAK WEB ÜZERİNDEN VERİ DEPOLAMA ve VERİ ALIŞVERİŞİ CİHAZI YAPIMI

## KİŞİSEL BULUTLAR

### Giriş

Günlük hayatımızda elimizden düşmeyen telefonlarımız sürekli yeni modelleri çıkmaktadır. Bu modeller ilgimizi çekmekte ve yeni telefonlar almaktayız. Biz bu telefonları alırken fiyatı daha ucuz olabilmesi adına düşük hafıza alanlılarını seçeriz. Biz sürekli yeni modelini düşünürüz fakat günlük yaşamda en büyük sıkıntımız hafıza alanları ve çeşitli yedeklemelerdir.

Günümüz 5G internet, mini bilgisayarlar, ultra hızlı depolama birimleri vb. teknolojiler yer alırken bizim telefon seçiminde sabit hafıza alanımızın derdine düşmemiz gerekmektedir. Bu durumda “verilerimiz neden telefona bağlı kalmalı?” sorusu aklıma gelmektedir. Günümüzde kullanılan NAS (Network Attached Storage) sistemleri, halk arasında uzak depolama veya bulutlar olarak geçer, ilham kaynağım olmaktadır. Ayrıca evlerimizde bulunan harici harddisk (SSD, HDD vb.) ve USB cihazlarımız vardır. Bunları veri depolama cihazlarıdır ve içlerinde verilerimizi saklarız fakat gerektiği zaman yanımızda olmazlar.

Bu olayların birleştirilmesi sonucu evimizde basit kurulumu yapılabilen uzak depolama alanları projesini yapıyorum. Raspberry Pi kullanarak web üzerinden veri depolama ve veri alışverişi cihazı yani “*kişisel bulutlar*” projemin detayları aşağıdadır:

### Kullanılan Donanım Bileşenleri:

1. Raspberry Pi 3 (Model B+),
2. Mikro SD card,
3. Statik IP (sabit IP) barındıran ağ sistemi ve modemi,
4. Harici harddisk veya USB,
5. Ethernet bağlantı kablosu ve +2A çıkış verebilen Micro USB adaptör.

### Kullanılan Yazılım Bileşenleri:

1. Raspbian Stretch (<https://www.raspberrypi.org/downloads/raspbian/>),
2. Linux Kernel yazılımı,
3. Ağ haberleşmesi ve ağ üzerinde port yönlendirmesi,
4. Owncloud.

### Kullanılan Bileşenlerin Özellikleri:

1. Raspberry Pi 3 (Model B+), son model raspberry pi olan B+ ile işlem hızlarımız son derece hızlı olacaktır. İstenirse 2
2. Micro SD card, raspberry pi'ımıza işletim sisteminin yüklenmesini ve veri depolama alanı sağlayan komponent.
3. Statik IP (sabit IP) barındıran ağ sistemi ve modemi, statik IP sabit bir dış IP'miz olmasını sağlar. Bu sayede ağ dışı bağlantılarımız kolaylaşır.
4. Harici harddisk veya USB, uzak depolama alanımız için istediğimiz sayısal değerde hafıza alanı sağlan komponent.

5. Ethernet bağlantı kablosu ve +2A çıkış verebilen Micro USB adaptör, Ethernet kablosu ile Pi'ımıza daha hızlı ağ erişimi sağlarız (wireless kullanılabilir\*), adaptör ile Pi'ımızın çalışması için gerekli elektriksel güç sağlanır.
6. Owncloud, bu arayüz ve programdır. Bunu sistemlerimize kurarak serverlarımıza bağlantı sırasında hoş bir arayüz ile karşılarız.

### **Yapım Aşamaları:**

Sistem 3 ana kurulum ve ayarlardan oluşmaktadır: (Raspberry Pi'in ilk kurulumu yapılmış varsaymaktayım. Raspbian Stretch işletim sistemi kuruldu.)

- A. Raspberry Pi'ı Owncloud server yapma
- B. Depolama alanlarına erişim
- C. Port yönlendirme ve harici erişim
- D. Owncloud'u ilk açma

### **A. Raspberry Pi'ı Owncloud Server Yapma:**

1. Öncelikle, Raspberry Pi'in komut satırı aracılığıyla Pi ve paketlerin güncellemesi kontrol edilip, güncellenir.

```
sudo apt-get update  
sudo apt-get upgrade
```

2. Pi'in çalışma alanının oluşturulması, bellek bölünmesi, verimli çalışma ve hız aşırma ayarı için bazı değişiklikler yapılır. Raspi config aracı açılır.

```
sudo raspi-config
```

Değiştirilecek kısımlar:

Internationalization options içerisinde Change local kısmında en\_US.UTF8 seçilmesi.

Advanced options içerisinde Memory split kısmında 16M seçilmesi.

Overclock medium olarak değiştirilir.

3. www-data grubunda www-data kullanıcısı oluşturulması.

```
sudo usermod -a -G www-data www-data
```

4. Raspbian Stretch kullanımına uyumlu olarak, gerekli tüm paketleri ve PHP7'in yüklenmesi.

```
sudo apt-get install nginx openssl ssl-cert php7.0-cli  
php7.0-sqlite php7.0-gd php7.0-common php7.0-cgi sqlite3 php-
```

pear curl libapr1 libtool curl libcurl4-openssl-dev php7.0-xml php7.0 php7.0-dev php7.0-curl php7.0-gd php7.0-fpm memcached php7.0-memcache php7.0-zip php7.0-intl php7.0-mbstring varnish

```
pi@raspberrypi:~$ sudo apt-get install nginx openssl ssl-cert php7.0-cli php7.0-sqlite php7.0-gd php7.0-common php7.0-cgi sqlite3 php-pear curl libapr1 libtool curl libcurl4-openssl-dev php7.0-xml php7.0 php7.0-dev php7.0-curl php7.0-gd php7.0-fpm memcached php7.0-memcache php7.0-zip php7.0-intl php7.0-mbstring varnish
Paket listeleri okunuyor... Bitli
Bağımlılık ağacı oluşturuluyor
Durum bilgisi okunuyor... Bitli
Not: 'php7.0-sqlite' düzenli ifadesi için 'php7.0-sqlite3' seçiliyor
Not: 'php7.0-memcache' yerine 'php-memcache' seçiliyor
curl zaten en yeni sürümde (7.51.5-2deb9).
openssl zaten en yeni sürümde (1.1.0j-1-deb5ul).
openssl elle kurulum olarak ayarlandı.
Aşağıdaki paket otomatik olarak kurulmuş ve artık bu pakete gerek duyulmuyor:
  readline
Bu paketi kaldırmak için 'sudo apt autoremove' komutunu kullanın.
Aşağıdaki ek paketler kurulacak:
  autoconf automake autopoint autotools-dev debhelper dh-autoreconf dh-php
  dh-strip-nondeterminism gettext intltool-debian libarchive-zip-perl
  libexporter-tiny-perl libfile-stripopndeterminism-perl libjemalloc1
  liblist-moreutil-perl libltdl-dev libmail-sendmail-perl
  libnginx-mod-http-auth-pam libnginx-mod-http-dav-ext libnginx-mod-http-echo
  libnginx-mod-http-geoip libnginx-mod-http-image-filter
  libnginx-mod-http-memc-filter libnginx-mod-http-upstream-fair
  libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libpcre3-dev libpcre2-3 libpcrecpp0v5 libsigsegv2 libssl-dev libssl-doc
  libsys-hostname-long-perl libtime-date-perl libvarnishapi libzip4 md
  nginx-common nginx-full php-common php7.0-json php7.0-opcache
  php7.0-readline pkg-php-tools po-debconf shtool xml2
Önerilen paketler:
  autoconf-archive gnu-stardards autoconf-doc dh-make gettext-doc
  libaspintf-dev libgettextpo-dev libcurl4-doc libcurl3-dbg libidn11-dev
  liblua5-dev liblua2-dev librtmp-dev libssl2.1-dev libssl1.0-dev
  libssl-dev libtool-doc gfortran | fortran95-compiler gcj-jdk m4-doc
  libcache-memcached-perl libmemcached libanyevent-perl libyaml-perl
  libterm-readkey-perl fcgiwrap nginx-doc libmail-box-perl sqlite3-doc
openssl-1.1.0-1-openssl-doc
Aşağıdaki VNI paketler kurulacak:
  autoconf automake autopoint autotools-dev debhelper dh-autoreconf dh-php
  dh-strip-nondeterminism gettext intltool-debian libapr1 libarchive-zip-perl
  libcurl4-openssl-dev libexporter-tiny-perl libfile-stripopndeterminism-perl
```

Adım5.PNG

5. SSL sertifikası oluşturulması, bu işlem sırasında sorular ile karşılaşacağız buraya görsel destek bırakıyorum. (Sorulara gerekli cevapların yazılıp enter'a basılması gerekmektedir.)

```
sudo openssl req $@ -new -x509 -days 730 -nodes -out
/etc/nginx/cert.pem -keyout /etc/nginx/cert.key
```

```
pi@raspberrypi:~$ sudo openssl req $@ -new -x509 -days 730 -nodes -out /etc/nginx/cert.pem -keyout
/etc/nginx/cert.key
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/nginx/cert.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TR
State or Province Name (full name) [Some-State]:istanbul
Locality Name (eg, city) []:istanbul
Organization Name (eg, company) [Internet Widgits Pty Ltd]:daldalcloud
Organizational Unit Name (eg, section) []:boss
Common Name (e.g. server FQDN or YOUR name) []:sercan daldal
Email Address []:sercandaldal@gmail.com
```

Adım6.PNG

6. SSL sertifikamıza ek olarak özel dhparam dosyası oluşturuyoruz. Bu dosya ile SSL bağlantımızın güvenliği üst düzeye çekilecektir. Bu adım beklenilenden uzun sürecektir. (~4 saat)

```
sudo openssl dhparam -out /etc/nginx/dh4096.pem 4096
```

7. Oluşturduğumuz üç adet cert dosyasını chmod'a çekeceğiz.

```
sudo chmod 600 /etc/nginx/cert.pem
```

```
sudo chmod 600 /etc/nginx/cert.key
```

```
sudo chmod 600 /etc/nginx/dh4096.pem
```

- 8.** Server config dosyamızın içerisini temizliyoruz. Server kurulumumuzu yapabilmek için.

```
sudo sh -c "echo ' ' > /etc/nginx/sites-available/default"
```

- 9.** Web sunucumuzun yapılandırılması için kullanılacak Owncloud için server configi ayarlayalım.

```
sudo nano /etc/nginx/sites-available/default
```

- 10.** Owncloud kurulumu ve gerekli kök izin ayarları için aşağıdaki hazır kodu kullanacağız. Kod içerisinde server\_name parametresiyle IP adresimi 192.168.0.20 olarak belirttim. Kullanıcı kullandığı cihazın lokal IP'si ile değiştirmelidir.

```
upstream php-handler {
    server 127.0.0.1:9000;
    server unix:/var/run/php/php7.0-fpm.sock;
}

server {
    listen 80;
    server_name 192.168.0.20;

    #Allow letsencrypt through
    location /.well-known/acme-challenge/ {
        root /var/www/owncloud;
    }

    # enforce https
    location / {
        return 301 https://$server_name$request_uri;
    }
}

server {
    listen 443 ssl http2;
```

```

server_name 192.168.0.20;

ssl_certificate /etc/nginx/cert.pem;
ssl_certificate_key /etc/nginx/cert.key;

ssl_session_timeout 5m;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:AES256+EECDH:AES256+EDH';
ssl_dhparam /etc/nginx/dh4096.pem;
ssl_prefer_server_ciphers on;
keepalive_timeout 70;
ssl_stapling on;
ssl_stapling_verify on;

add_header X-Content-Type-Options nosniff;
add_header X-Frame-Options "SAMEORIGIN";
add_header X-XSS-Protection "1; mode=block";
add_header X-Robots-Tag none;
add_header X-Download-Options noopen;
add_header X-Permitted-Cross-Domain-Policies none;

root /var/www/owncloud/;

location = /robots.txt {
    allow all;
    log_not_found off;
    access_log off;
}

# The following 2 rules are only needed for the
user_webfinger app.

# Uncomment it if you're planning to use this app.

```

```

#rewrite                                ^/.well-known/host-meta
/public.php?service=host-meta last;

#rewrite                                ^/.well-known/host-meta.json
/public.php?service=host-meta-json last;


location = /.well-known/carddav {
    return 301 $scheme://$host/remote.php/dav;
}
location = /.well-known/caldav {
    return 301 $scheme://$host/remote.php/dav;
}


# set max upload size
client_max_body_size 512M;
fastcgi_buffers 8 4K;
fastcgi_ignore_headers X-Accel-Buffering;


gzip off;


error_page 403 /core/templates/403.php;
error_page 404 /core/templates/404.php;


location / {
    rewrite ^ /index.php$uri;
}


location                                ~
^/(?:build|tests|config|lib|3rdparty|templates|data)/ {
    return 404;
}

location                                ~
^/(?:\.|autotest|occ|issue|indie|db_|console) {
    return 404;
}

```

```

        location ~
^/(?:index|remote|public|cron|core/ajax/update|status|ocs
/v[12]|updater/.+|ocs-
provider/.+|core/templates/40[34])\.php(?:$|/) {

    fastcgi_split_path_info ^(.+\.(php))(/.*)$;
    include fastcgi_params;

    fastcgi_param          SCRIPT_FILENAME
$document_root$fastcgi_script_name;
    fastcgi_param SCRIPT_NAME $fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_path_info;
    fastcgi_param HTTPS on;
    fastcgi_param modHeadersAvailable true;
    fastcgi_param front_controller_active true;
    fastcgi_read_timeout 180;
    fastcgi_pass php-handler;
    fastcgi_intercept_errors on;
    fastcgi_request_buffering off; #Available since
NGINX 1.7.11
    }

    location ~ ^/(?:updater|ocs-provider)(?:$|/) {
        try_files $uri $uri/ =404;
        index index.php;
    }

    location ~ \.(?:css|js)$ {
        try_files $uri /index.php$uri$is_args$args;
        add_header Cache-Control "max-age=15778463";

        # Before enabling Strict-Transport-Security
headers please read into this topic first.

        #add_header Strict-Transport-Security "max-
age=15552000; includeSubDomains";
        add_header X-Content-Type-Options nosniff;
        add_header X-Frame-Options "SAMEORIGIN";
        add_header X-XSS-Protection "1; mode=block";

```

```

        add_header X-Robots-Tag none;
        add_header X-Download-Options noopen;
        add_header      X-Permitted-Cross-Domain-Policies
none;

        access_log off;
    }

    location ~
\. (?:svg|gif|png|html|ttf|woff|ico|jpg|jpeg|map)$ {
        add_header Cache-Control "public, max-age=7200";
        try_files $uri /index.php$uri$is_args$args;
        access_log off;
    }
}

```

**11.** Şimdi kök dosyayı ctrl+X kontrolü ile çıkış diyeceğiz. Bu sıra kaydetme işlemini soracak, onaylayacağız ve çıkacağız..

**12.** PHP config dosyamızın içerisine giriyoruz. Bazı kısımları güncelleyeceğiz.

```
sudo nano /etc/php/7.0/fpm/php.ini
```

Değiştireceğimiz kısımlar:

```
upload_max_filesize = 2000M
post_max_size = 2000M
```

Bu kısımda belirtilen **2000M** ifadeleri belirtilen parametrelerin sınırını belirler, değiştirebiliriz. Kayıt edip çıkış yapıyoruz.

**13.** conf dosyamızın içerisine giriyoruz. Bazı kısımları güncelleyeceğiz.

```
sudo nano /etc/php/7.0/fpm/pool.d/www.conf
```

Değiştireceğimiz kısımlar:

```
listen = 127.0.0.1:9000
```

Bu kısımda alıcı (dinleyici) hattımızı güncelledik. Kayıt edip çıkış yapıyoruz.

**14.** dphys-swap dosyamızın içerisine giriyoruz. Bazı kısımları güncelleyeceğiz.

```
sudo nano /etc/dphys-swapfile
```

Değiştireceğimiz kısımlar:

```
CONF_SWAPSIZE = 512
```



Gerekli değişikliği yapıp, kaydederek çıkış yapacağız.

**15.** Pi'ımızı yeniden başlatarak bilgilerin güncellenmesini yapalım.

```
sudo reboot
```

**16.** Pi'ımızı yeniden başlattıktan sonra açılınca Owncloud'u Pi'ımıza kuralım.

```
sudo mkdir -p /var/www/owncloud
sudo
https://download.owncloud.org/community/owncloud-10.0.9.tar.bz2
sudo tar xvf owncloud-10.0.9.tar.bz2
sudo mv owncloud/ /var/www/
sudo chown -R www-data:www-data /var/www
rm -rf owncloud owncloud-10.0.9.tar.bz2
```

wget

**17.** .user.ini dosyamızın içerisine giriyoruz. Bazı kısımları güncelleyeceğiz.

```
sudo nano /var/www/owncloud/.user.ini
```

Değiştireceğimiz kısımlar:

```
upload_max_filesize=2000M
post_max_size=2000M
memory_limit=2000M
```

Gerekli değişikliği yapıp, kaydederek çıkış yapacağız.

Bu işlemler sonucu Pi'ımızın lokal IP adresi üzerinden ağ bağlantısı kurarak Owncloud'a bağlanabiliriz.

## **B. Depolama Alanlarına Erişim**

Bu talimatlar öncesinde Pi'ımızın USB girişlerine elimizdeki harici diskleri ve USB'leri takalım.

**1.** Öncelikle, NTFS sürücü paketi kurulumunu gerçekleştireceğiz.

```
sudo apt-get install ntfs-3g
```

**2.** Artık portlarda bulunan harici diskler ile irtibata geçebiliriz. Ortak bir dizin oluşturalım.

```
sudo mkdir /media/ownclouddrive
```

**3.** İleride ayarlamalarını yapabilmemiz için GID, UID ve UUID'leri açacağız.

**3.1.** GID için:

```
id -g www-data
```

### 3.2. UID için:

```
id -u www-data
```

### 3.3. UUID için: (Bu işlem ile harici disk farklı bir USB porta taksanız bile anımsayacaktır.)

```
ls -l /dev/disk/by-uuid
```

Son satırda (../sda1 uzantılı ) bulunan mavi renkteki harfleri ve numaraları kopyalayınız.)

```
pi@raspberrypi:~ $ ls -l /dev/disk/by-uuid
toplam 0
lrwxrwxrwx 1 root root 15 Nis 16 17:38 5fa1ec37-3719-4b25-be14-1f7d29135a13 -> .
../mmcblk0p2
lrwxrwxrwx 1 root root 15 Nis 16 17:38 DDAB-3A15 -> ../mmcblk0p1
lrwxrwxrwx 1 root root 10 Nis 16 17:38 EA7C-204D -> ../sda1
```

### depolamaAdım3\_3.PNG

4. Portumuza bağlı diskimizi fstab dosyasına ekleyelim. Bu sayede doğru izinleri alabilecektir.

```
sudo nano /etc/fstab
```

5. fstab dosyamızın en alt satırına aşağıda bulunan kodu ekleyeceğiz. Kodda bulunan UUID değerini adım 3.3'de kopyaladığımız kısım ile değiştiriniz.

```
UUID=EA7C-204D /media/ownclouddrive auto
nofail,uid=33,gid=33,umask=0027,dmask=0027,noatime 0 0
```

Pi'imizi yeniden başlatalım (reboot). Pi'miz yeniden başlatılınca portlarda bulunan diskler hazır olacaktır.

## C. Port Yönlendirme ve Harici Erişim

Bulut sunucumuza yerel ağımız dışında bağlanabilmemiz için bağlantı noktamızı yönlendirmemiz (port yönlendirme) gerekmektedir. Ayrıca .config dosyamızda birkaç değişiklik yapılacaktır.

Statik IP'imizi internet dağıtımımız aracılığıyla almış bulunmaktayız. Modem arayüzünü açarak orada port yönlendirme işlemini yapmalıyız. İçeri de Pi'nin lokal IP'si olacak, dışarıda ise Statik IP'miz olacaktır. Port ayarımız ise 443. Port yani HTTPS olmalıdır.

1. Varsayılan sunucu dosyamızı oluştururuz.

```
sudo nano /etc/nginx/sites-available/default
```

2. Owncloud .config dosyamızı açıyoruz ve gerekli güncellemeleri yapıyoruz.

```
sudo nano /var/www/owncloud/config/config.php
```

```
'trusted_domains' => kısmında aşağıda bulunan görsel olduğu gibi
1 => 'xxx.xxx.xxx.xxx' kısmını ekliyoruz. Burada belirtilen x'lere Statik
IP'miz konumlandırılacak. Aynı şekilde;
'overwrite.cli.url' => 'https://xxx.xxx.xxx.xxx',
Kısımına da uyguluyoruz.
```

```
GNU nano 2.7.4 File: /var/www/owncloud/config/config.php
<?php
$CONFIG = array (
  'instanceid' => 'ocu56rsud0oz',
  'passwordsalt' => '2xb6AQd3yLh5GB5ce41+m+Ab5TZrgD',
  'secret' => 'ne4Ac6HZc3BV1jT5FymWZkP09q73np4CFmGGAsnYILR0/VhS',
  'trusted_domains' =>
  array (
    0 => '192.168.0.20',
    1 => '46.196.105.190',
  ),
  'datadirectory' => '/media/ownclouddrive',
  'overwrite.cli.url' => 'https://46.196.105.190',
  'dbtype' => 'sqlite3',
  'version' => '10.0.9.5',
  'logtimezone' => 'UTC',
  'installed' => true,
);

[ Read 17 lines ]
^G Yardım Al ^O Write Out ^W Ara ^K Metni Kes ^J Yasla ^C İmleç Pozisy
^X Çık ^R Dosya Oku ^\ Değiştir ^U Uncut Text ^T Denetime ^_ Satıra Git
```

portAdım2.PNG

3. Nginx servisi yeniden başlatarak güncellenmeli.

```
sudo service nginx restart
```

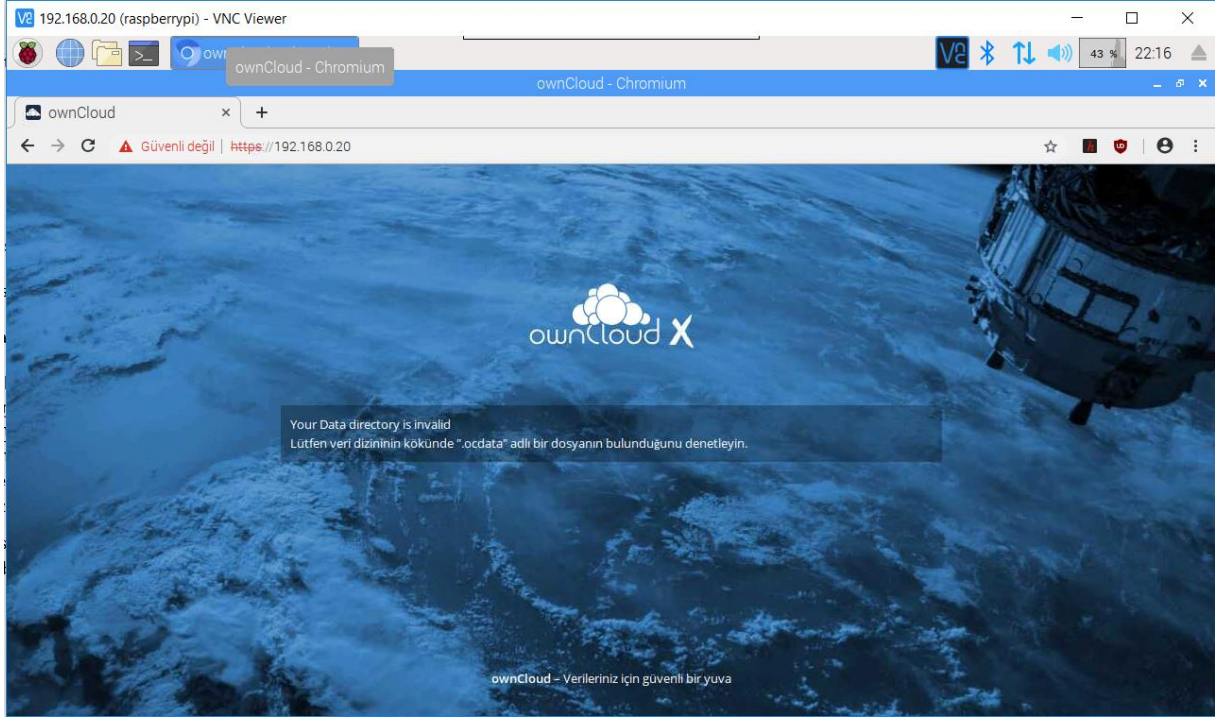
Pi yeniden başlatılabilir.

#### D. Owncloud'u İlk Açma

1. Tarayıcıma Pi'nin lokal IP'sini giriniz veyahut Starik IP kurulumu yaptıysanız onuda kullanabilirsiniz.
  2. Bir güvenlik hatası alacaksınız. Bunu görmezden gelebilirsiniz, sertifika izinlerinden olmaktadır.
  3. Owncloud ilk açılımda kullanıcı adı ve şifre ister orada biz belirleriz. İstedığınız kullanıcı adı ve şifreyi belirleyiniz.
  4. Sonrasında erişim izni olan kök dosyanın konumunu ister buna da bizim giriş yapmamız gerekir. Biz kodlarımız da /media/owncloud konumunu belirledik.
  5. En son olarak kurulumu tamamlama butonuna basıp bitiririz.
-

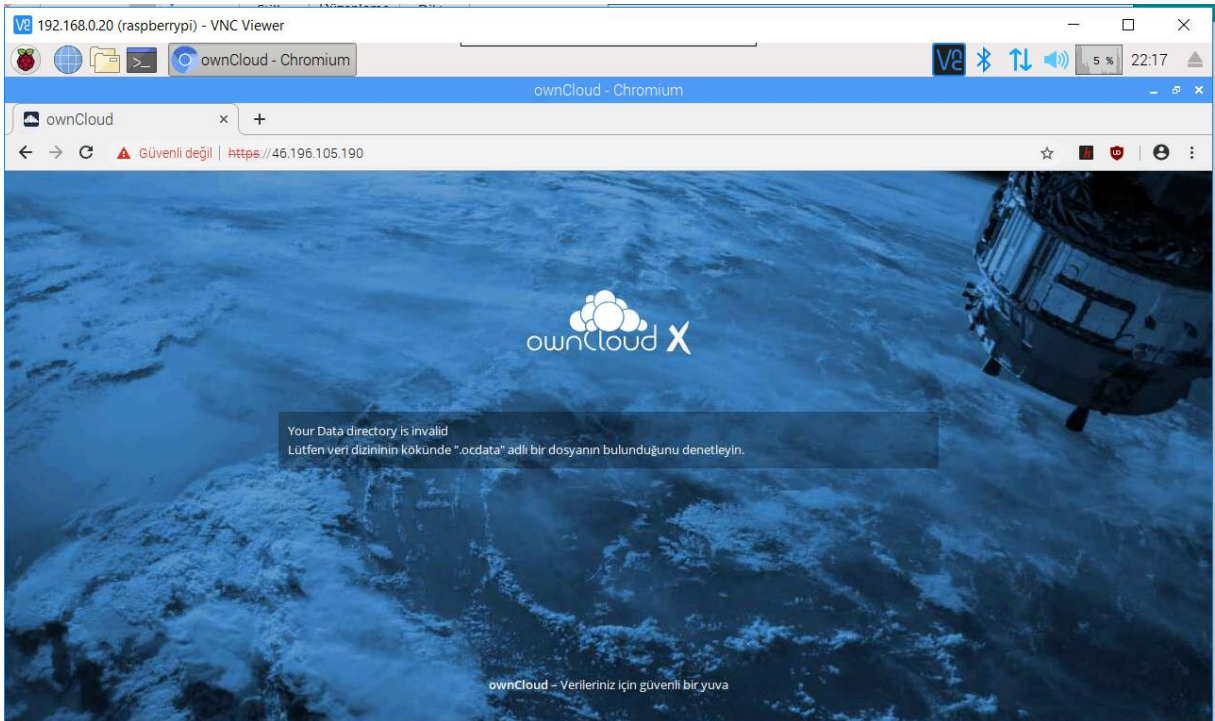
## Projenin Çalışması ile Alakalı Gerekli Ekran Görüntüleri:

Pi'nin lokal IP'si olan 192.168.0.20 üzerinden bağlanması:



0.20.PNG

Pi'nin Statik IP'si olan 46.196.105.190 üzerinden bağlanması:



105.190.PNG

Bağlantıların fotoğrafı:



20190416\_222836.JPG

---

#### 16.04.2019 NOTLAR:

Merhaba hocam, son safha olan Owncloud kullanıcı adı ve şifre belirlenimi ve kök dosya belirlenimi kısmındayım. Henüz bitirememiş olmamın sebebi karşılaştığım hatalar ve bunları çözmeye odaklanmamdır.

En son olarak .ocdata kök dosyasına erişilememesi söz konusudur. Rapor teslimime yetiştiremem için üzgün olmamla beraber gerekli hata ayıklamaları sonrası githubda bulunan kodlarımı güncelleyeceğim.

Youtube'a video atmış bulunmaktayım. Tamamen giriş videosu oldu. Projem çalışınca daha detaylısını çekeceğim. Bende istiyorum ki internet üzerinde Türkçe kaynakların sayısı artsın.

<https://youtu.be/iXCd3EYrlag>