

UNIT 4: APPLICATIONS OF IOT AND CLOUD

OVERVIEW

Cloud computing and IoT are two different technologies which are often seen as complementary. The limitations of IoT devices having low processing, storage and connectivity can be addressed by the virtually unlimited computational power, storage capacity and extensibility of the cloud to yield systems with immense potential in diverse fields ranging from healthcare to space exploration [x1]. In this unit, we will look at the different approaches that can be used to process and store data (generated by IoT devices) on the cloud. You will also learn how to build and secure your own personal cloud service.

LEARNING OUTCOMES

By the end of this Unit, you should be able to do the following:

1. Analyse potential applications of IoT and the cloud.
2. Develop IoT solutions and implement cloud services through case studies.
3. Use visualisation techniques to show data generated from the IoT devices.
4. Implement and secure your own cloud service.

TOPIC 1: IoT AND CLOUD INTEGRATION

Case study: Getting Room Temperature and Sending the Data to the Cloud

Pre-requirements:

i. Hardware

- LM35 (Temperature sensor)
- USB cable,
- Jumper cables
- Breadboard.
- ESP5266 (NodeMCU)

- Blue and red LED
- 2 resistors 330 Ohm

ii. Software

- Arduino IDE
- ThingSpeak

Cloud storage

ThingSpeak is an IoT analytics platform which enables processing and visualisation of IoT data in the cloud [x9]. The ThingSpeak platform will be used in subsequent case studies to store and visualise the IoT data.

In this case study, we are going to use the hardware to get the room temperature at intervals of 10 minutes and send those values to the cloud platform, ThingSpeak.

Setting up the hardware (IoT testing without cloud)

You will need the Arduino Uno, the USB A to B cable to connect the Arduino to your PC/Laptop Temperature sensor LM35, Jumper cables and a breadboard.

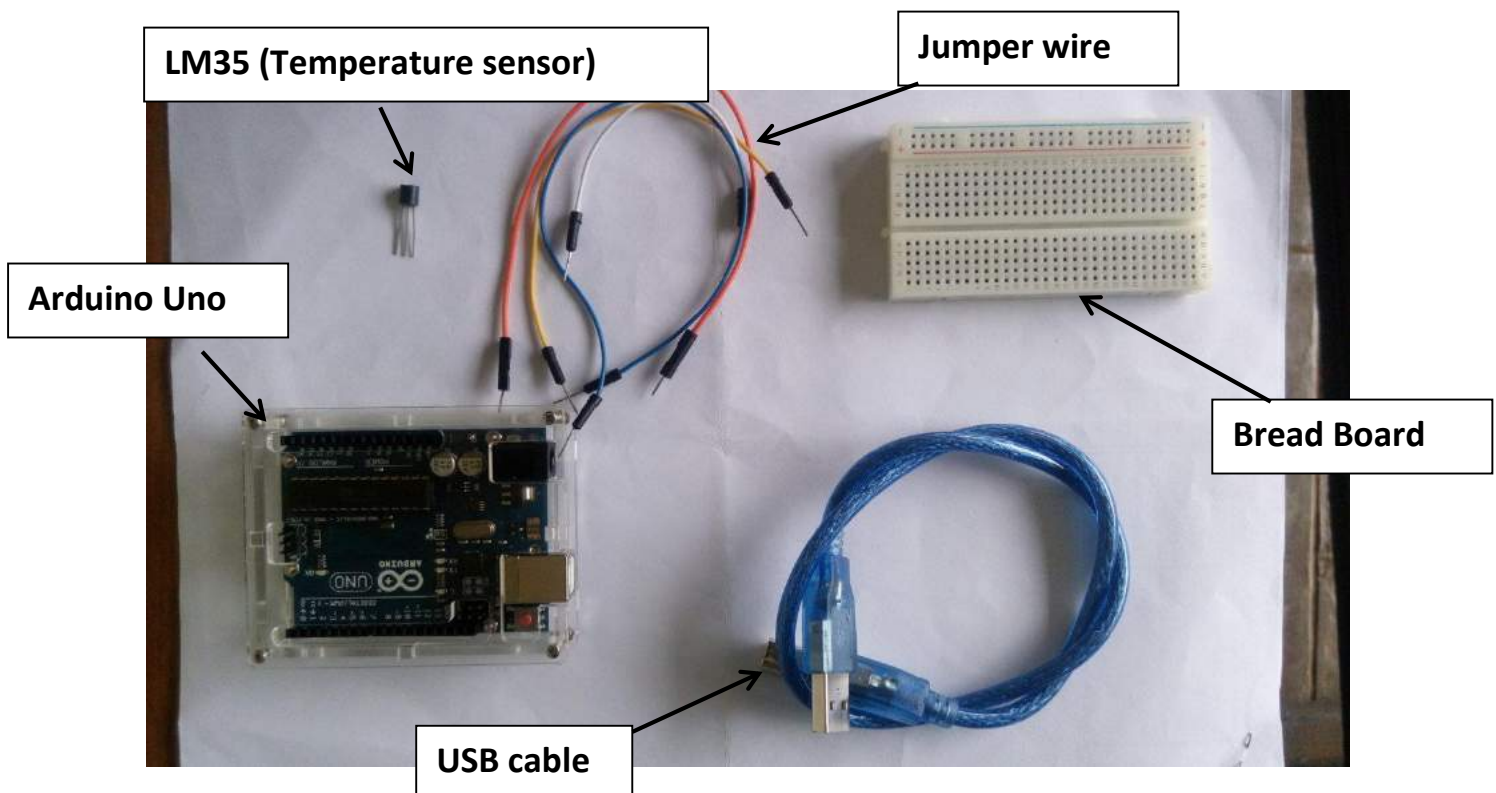
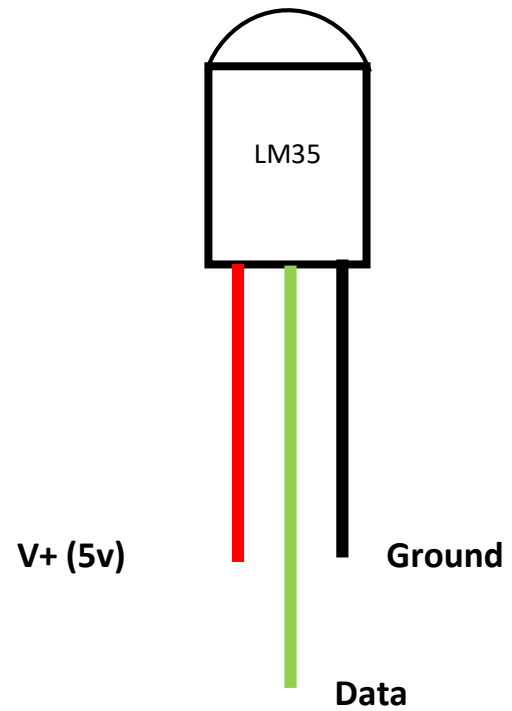


Figure 1



Facing the flat part of the sensor, the pins are labelled as shown in Figure 2 above. The circuit is connected as follows: the positive pin connects to the positive 5v while the last pin connects to the ground pin and the data pin connects to the analog pin A0 of the Arduino.

The sensor will be linked to the Arduino by jumper wires which are connected to the breadboard as shown below.

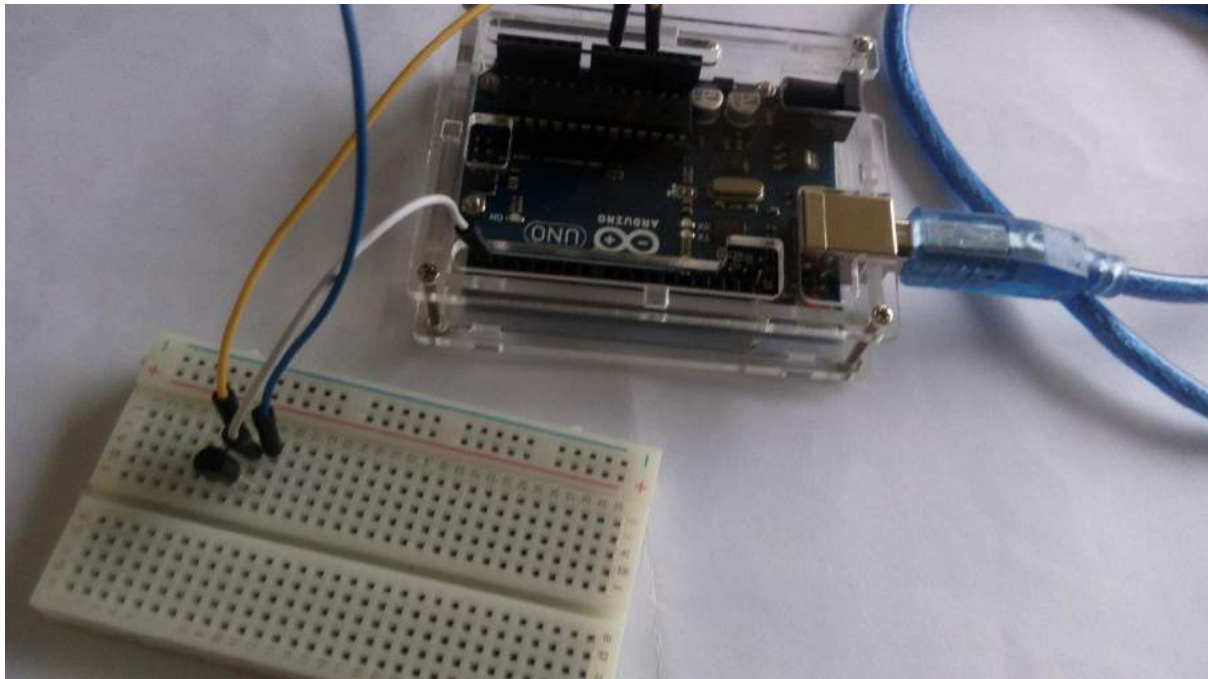


Figure 3

Next step is to write the code to get the temperature values from the sensor. Note that the sensor only sends electrical signal to the Arduino. The sketch will have to interpret these values and return the equivalent in degrees Celsius. For that purpose, we are going to use a formula to convert the electrical values to temperature values.

This can be summarised according to the pseudo code below

```
BEGIN
INITIALISE
REPEAT
    Read value from pin A0
    Apply formula to convert readings to degrees Celsius
    Send data to Serial port
    Wait 1 second to send next temperature value
END-REPEAT
END
```


The sketch resulting from the above pseudocode is shown below:

```
float resolution=3.3/1023;// 3.3 is the supply voltage & 1023 is max analog read value
void setup() {
  Serial.begin(9600);
}
void loop(){
  float temp = (analogRead(A0) * resolution) * 100; // converting the value obtained into temperature
  Serial.print("TEMPRATURE = ");
  Serial.print(temp);
  Serial.print("*C");
  Serial.println();
  delay(1000);
}
```

Code Listing 1

Code listing 2 adapted from:

<https://www.youtube.com/watch?v=BwDz6zHbGLI&feature=youtu.be>

Compile and upload the code to the Arduino and see the temperature values returned by clicking on the Serial monitor button . This will load the serial monitor and display the temperature values every second as shown in figure 4 below.

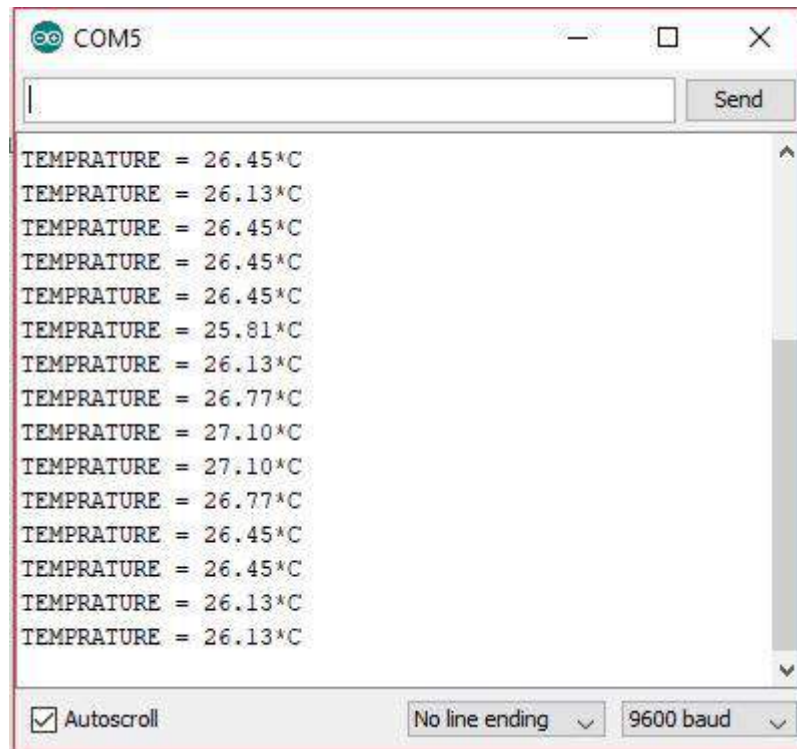


Figure 4

Sending the Data to the Cloud

Go to ThingSpeak (<https://thingspeak.com/>) website and Sign up for a new account.

Click on the 'Get Started for Free' button.

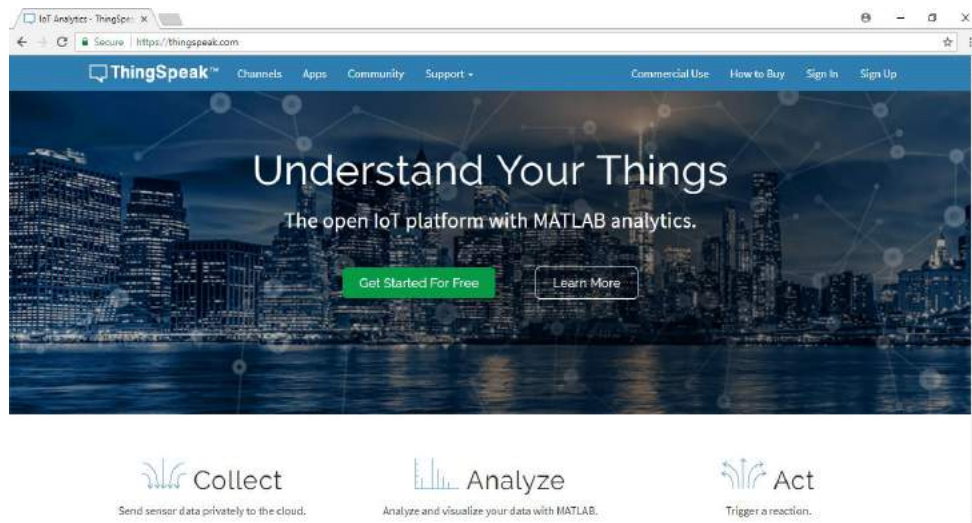


Figure 5

This will load the Sign up page. Fill in your details and click on continue. You will receive an activation email. Follow instructions to activate your account. When you are prompted to activate Matlab license, click on 'No Thanks' button.

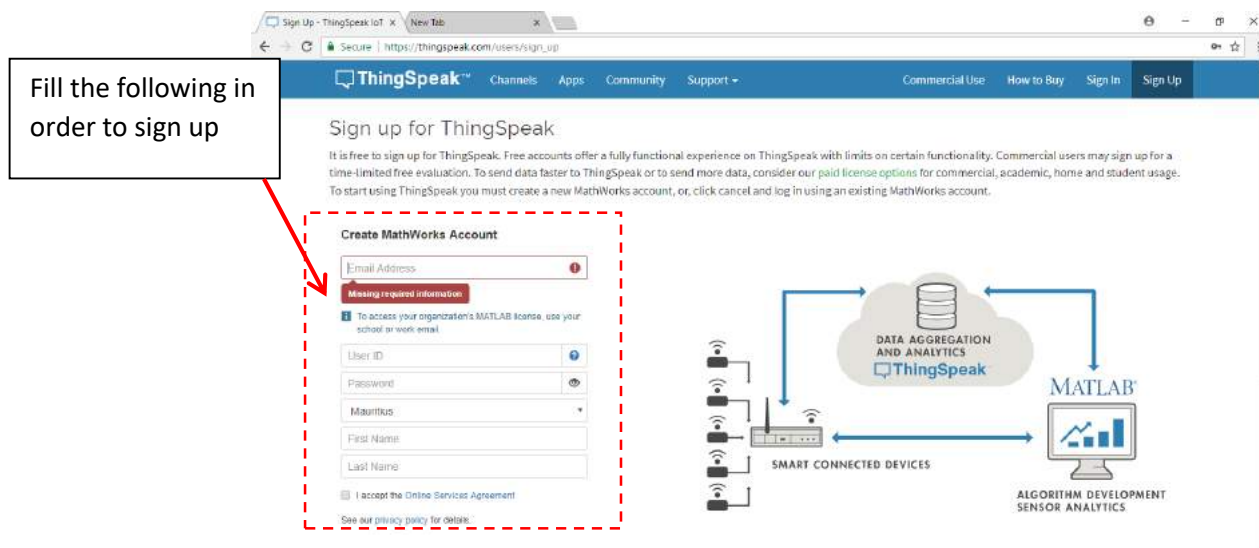


Figure 6

Go to the ThingSpeak site again and log in.

Next we need to add a new channel in order to retrieve the data from the NodeMCU and post it online.

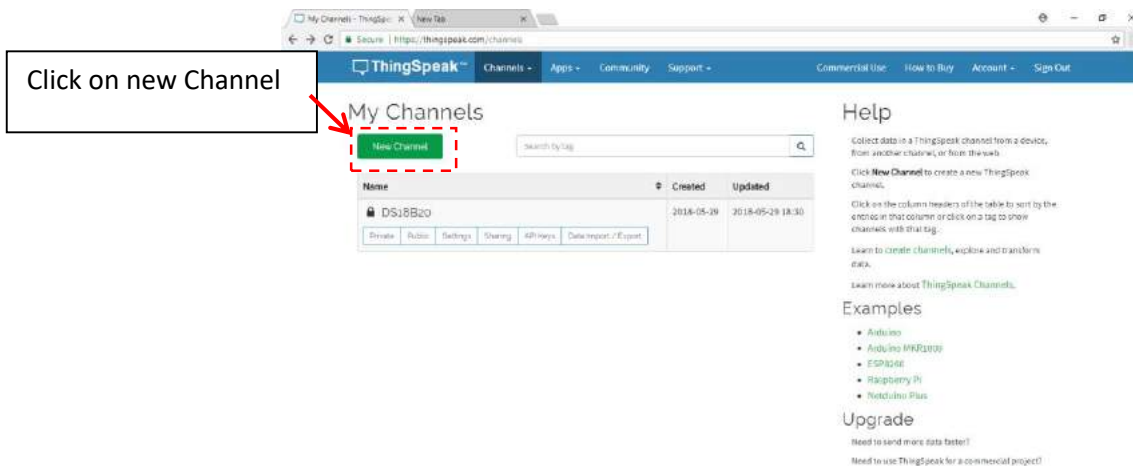


Figure 7

Fill Name as **LM35** and Field 1 as **Temperature** as shown below:

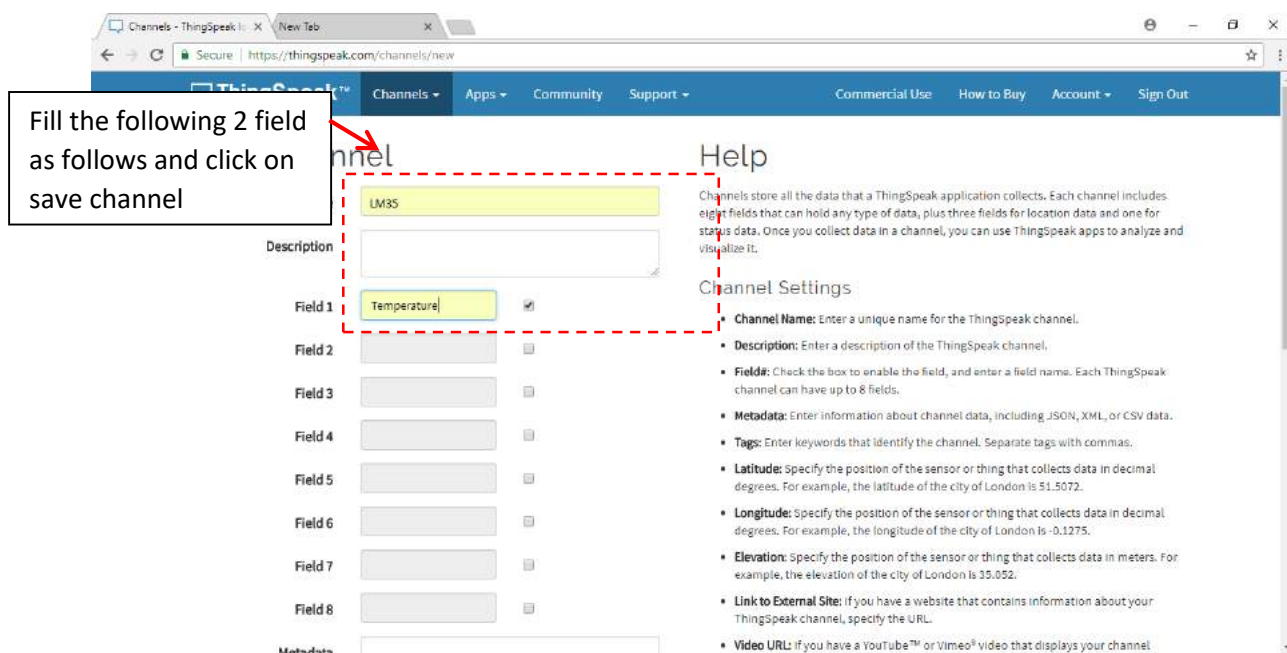


Figure 8

Once this is done, click on API key and copy and paste the Write and Read API key in Code Listing 2 below.

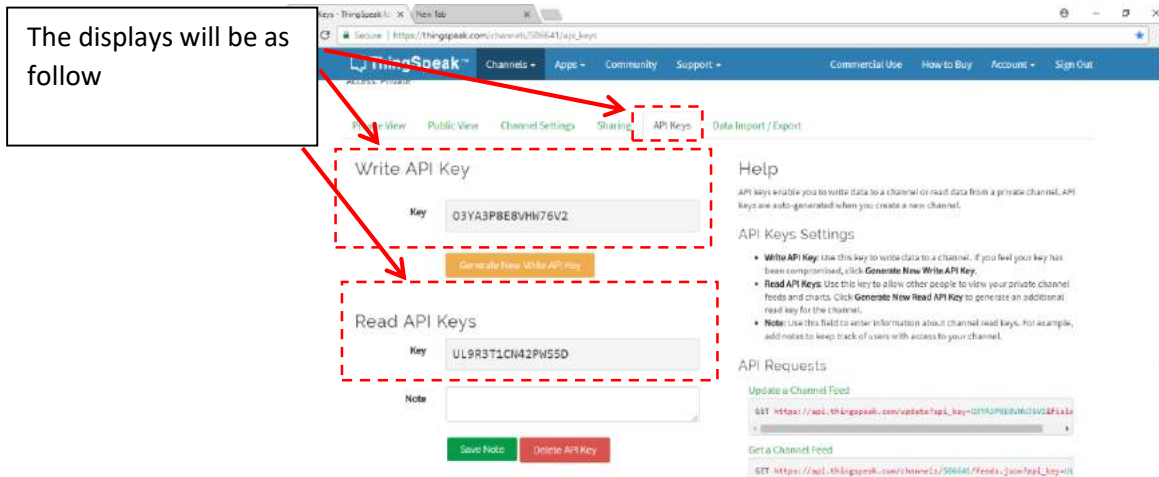


Figure 9

```
#include <ESP8266WiFi.h>

String apiKey = "your API key here"; // the API key as show above in the diagram
const char* MY_SSID = "your SSID here"; // your WIFI name
const char* MY_PWD = "your SSID password here"; // your WIFI password

const char* server = "api.thingspeak.com";

float resolution=3.3/1023; // 3.3 is the supply volt & 1023 is max analog read value

int RedLED = 2;           //output pin for LED
int BlueLED = 4;          //output pin for LED

WiFiClient client;

void setup() {
  pinMode(BlueLED,OUTPUT);
  pinMode(RedLED,OUTPUT);

  Serial.begin(115200);

  WiFi.disconnect();

  delay(10);

  WiFi.begin(MY_SSID, MY_PWD);
```

```
Serial.println();

Serial.println();

Serial.print("Connecting to ");

Serial.println(MY_SSID);


WiFi.begin(MY_SSID, MY_PWD);


while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}

Serial.println("");
Serial.print("NodeMcu connected to wifi...");
Serial.println(MY_SSID);
Serial.println();
}

void loop() {

    float temp = (analogRead(A0) * resolution) * 100; // converting the value obtained into temperature


    if (client.connect(server,80)) // NodeMCU use Port 80 to send data
    {
        String tsData = apiKey;

        tsData += "&field1=";

        tsData += String(temp);

        tsData += "\r\n\r\n";


        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");
        client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
    }
}
```

```

client.print("Content-Type: application/x-www-form-urlencoded\n");
client.print("Content-Length: ");
client.print(tsData.length());
client.print("\n\n");
client.print(tsData);

Serial.print("Temperature: ");
Serial.print(temp);
Serial.println("°C");
Serial.println("uploaded to Thingspeak server....");
if (temp < 30)
{
    digitalWrite(RedLED, LOW); // turn the LED on (HIGH is the voltage level)
    digitalWrite(BlueLED, HIGH); // turn the LED off by making the voltage LOW
}
else if (temp > 30)
{
    digitalWrite(RedLED, HIGH); // turn the LED on (HIGH is the voltage level)
    digitalWrite(BlueLED, LOW); // turn the LED off by making the voltage LOW
}
}
client.stop();

Serial.println("Waiting to upload next reading...");
Serial.println();
delay(15000); // thingspeak needs minimum 15 sec delay between updates
}

```

Code Listing 2

Three parameters need to be changed:

```
String apiKey ="your API key here"; // the API key as show above in the diagram
const char* MY_SSID = "your SSID here"; // your WIFI name
const char* MY_PWD = "your SSID password here";// your WIFI password
```

Crash Course on NodeMCU

In order to communicate with the cloud service **ThingSpeak**, we will have to learn how to use the ESP8266 board (NodeMcu). The NodeMcu is an open source hardware that allows us to send data over the internet via a WIFI chip. This hardware can be programmed using the Arduino IDE, but we will have to install some drivers first.

Note: Make sure the latest version of Arduino is installed on your machine.

Step 1: Connect NodeMCU to the Computer

Use the USB cable to connect the NodeMCU to the computer. The blue onboard LED will flicker once when powered up.

Step 2: Install the COM/Serial Port Driver

In order to upload code to the ESP8266 and use the serial console, connect a micro-usb data cable to ESP8266 IoT Board and your PC/Laptop. Download and install the relevant driver from the links provided below.

- The new version NodeMCUv1.0 comes with the CP2102 serial chip. Download and install the driver from: <https://www.silabs.com/products/development-tools/...>
- The NodeMCUv0.9 comes with the CH340 serial chip. Download and install the driver from: <https://github.com/nodemcu/nodemcu-devkit/tree/mas...>

Step 3: Set up NodeMCU in Arduino IDE (requires v1.6.4 or greater)

- Open the Arduino IDE
- Go to **files** and click on the **preference** in the Arduino IDE:
- Write the url **http://arduino.esp8266.com/stable/package_esp8266com_index.json** the in the Additional boards Manager URLs as shown below.

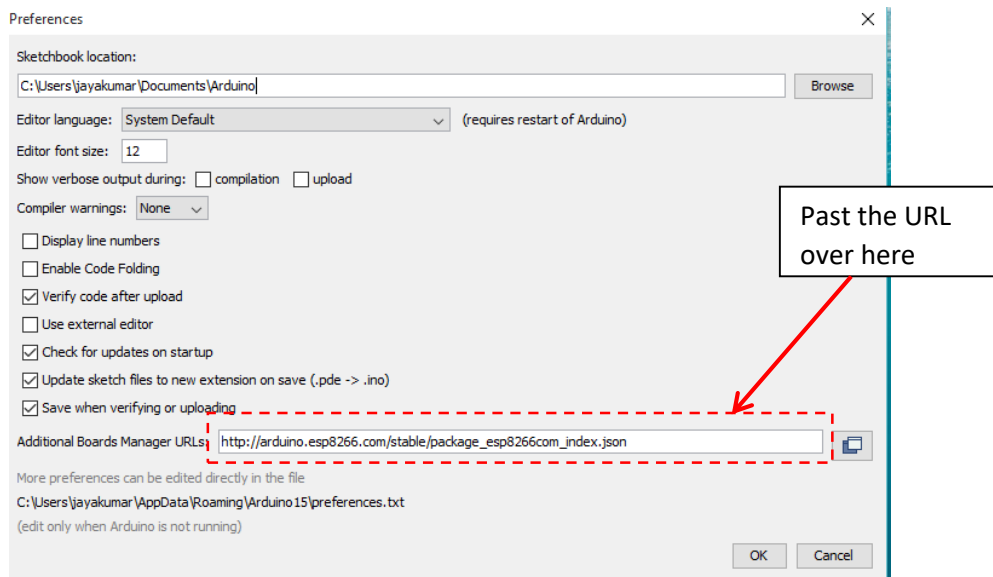


Figure 10

- click OK to close the preference Tab.
- After completing the above steps, go to **Tools** and **board**, and then select **board Manager**

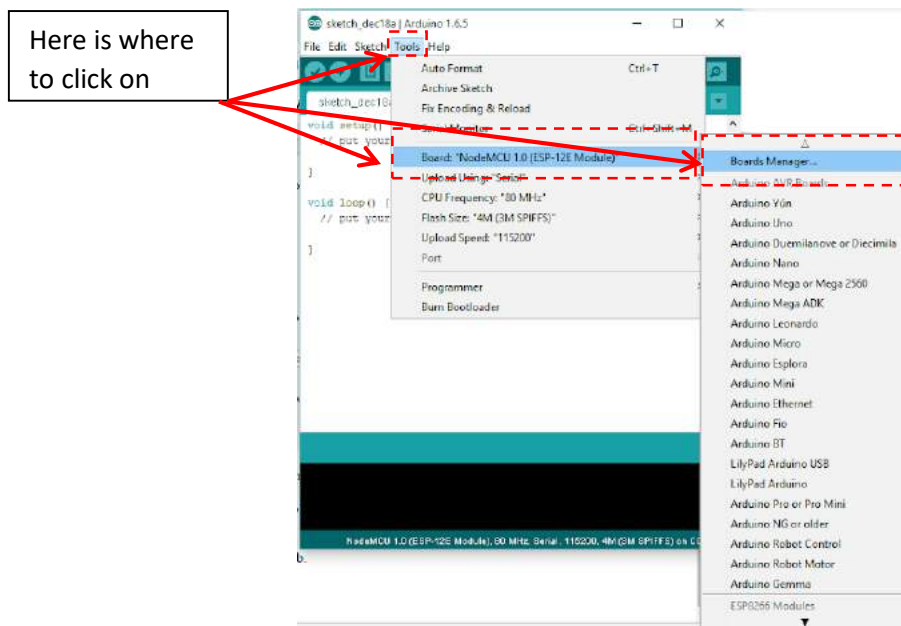


Figure 11

- Navigate to esp8266 by esp8266 community and install the software for Arduino.
- Once all the above process been completed everything is set to program our NodeMcu (esp8266) with Arduino IDE.



Figure 12

Sending Temperature to ThingSpeak

Construct the circuit as shown in **Figure 13**:

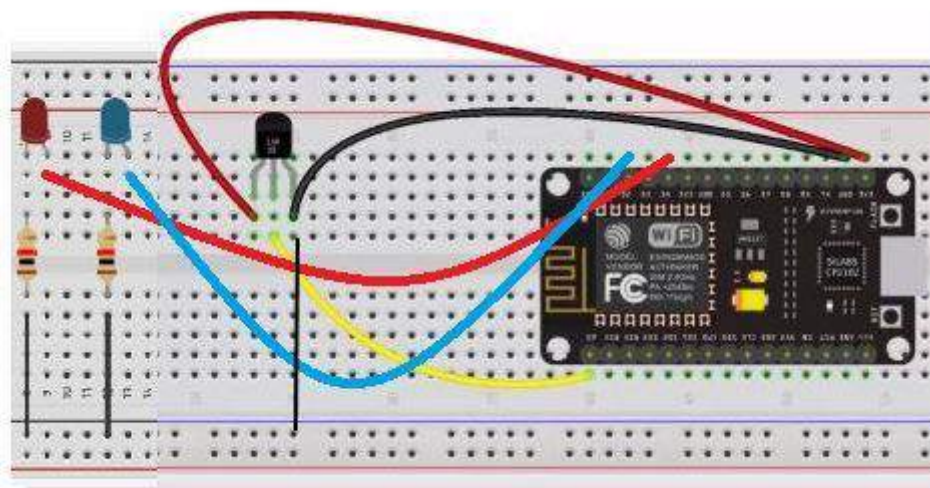


Figure 13

The different components needed:

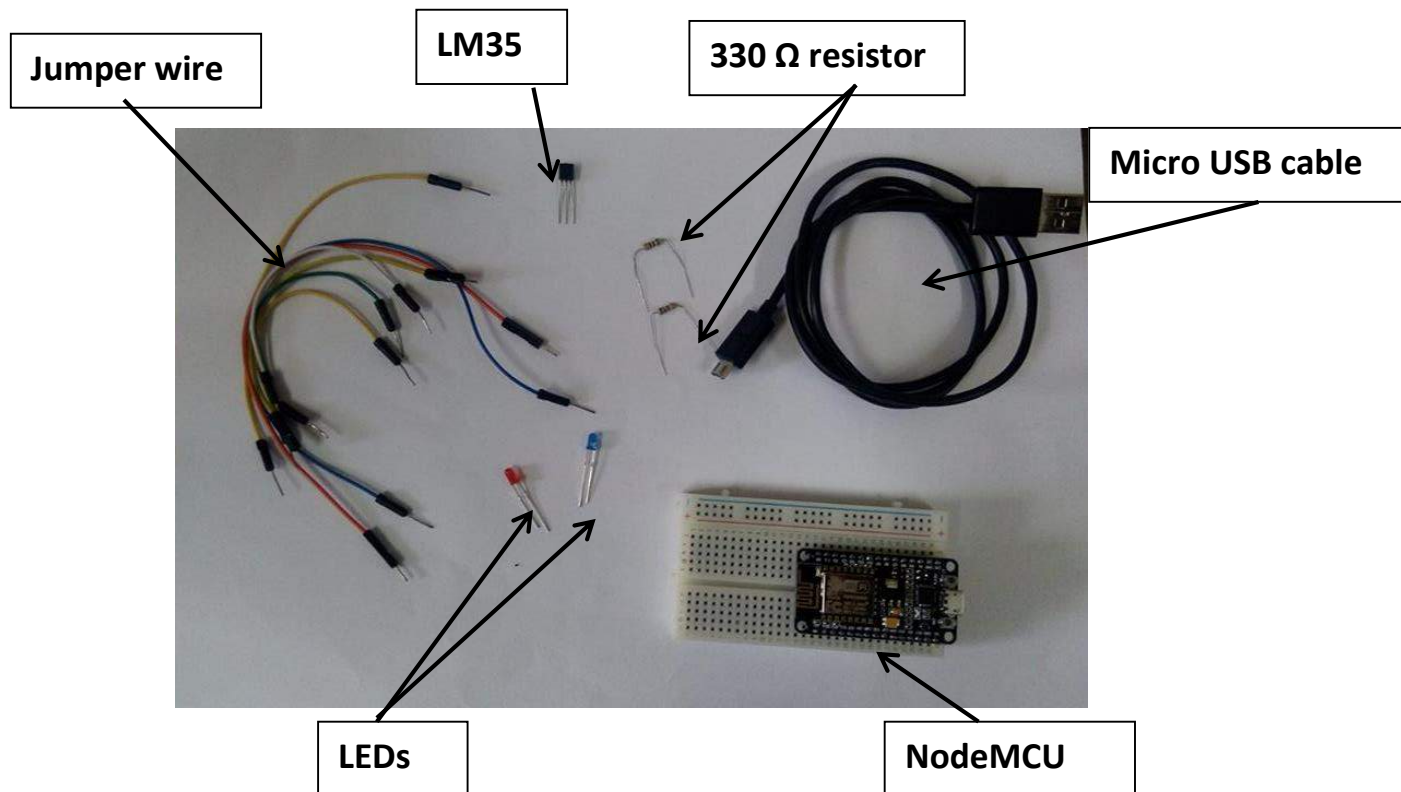


Figure 14

The circuit is connected as follow:

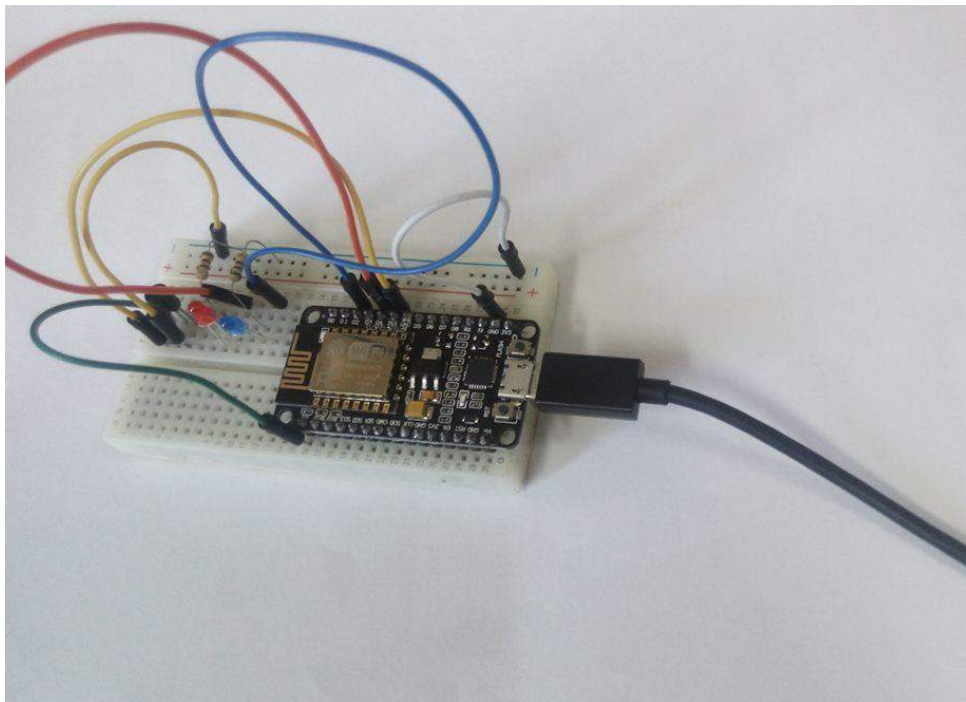


Figure 15

Connect the above circuit to your PC/Laptop and upload the code in code listing 2 via the Arduino IDE to the NodeMcu. Open ThingSpeak dashboard to see the following graph being generated.

Channel Stats

Created: about 2 hours ago
Updated: about 2 hours ago
Entries: 0

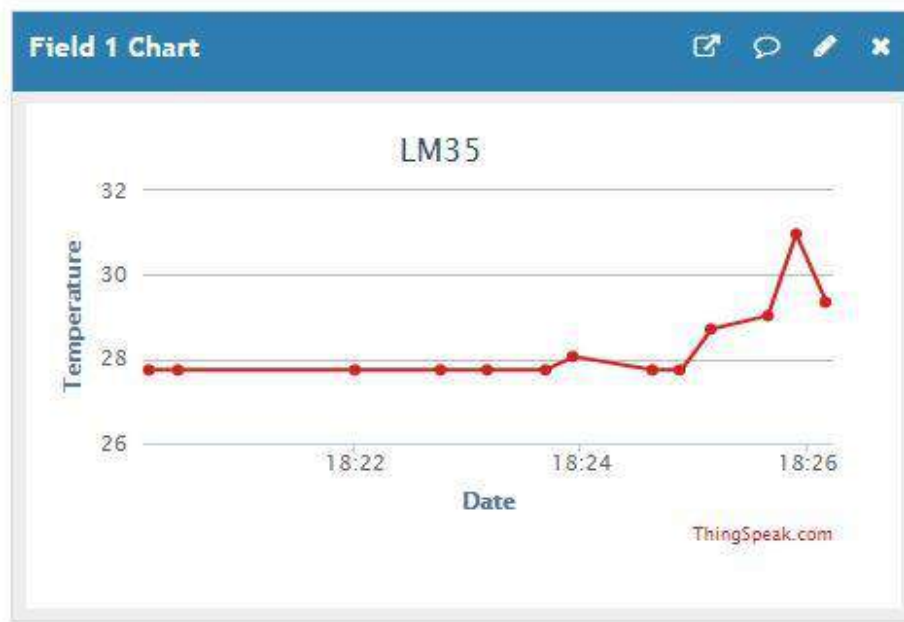


Figure 16

TOPIC 2: APPLICATION DEVELOPMENT AND CLOUD PROCESSING

Case study: Building Personal Cloud Storage Environment with Raspberry Pi and Nextcloud

Pre-requirement:

i Hardware

Raspberry PI 3 Model B, Casing for Raspberry Pi (optional), 16GB memory card with Raspbian preinstalled, power supply, USB mouse, USB keyboard, HDMI cable, LCD Screen

Availability for purchase at <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/> or other authorised distributors.

ii Software

Raspbian should be preloaded by default on the memory card which came with your Raspberry PI. Alternatively, Raspbian which can be downloaded at <https://www.raspberrypi.org/downloads/>

Nextcloud server can be downloaded at <https://nextcloud.com/install/> (Note that for this case study we will be using version 12.0.3 since there was some stability issues with version 13 at the time of writing).

Mount raspberry Pi in the PI case



Figure 17

I. Initial Setup

1. Connect the Raspberry Pi (with the memory card installed) to the TV and power on.
2. On the Initial (Noobs) setup screen, select Raspbian as the OS.
3. Select the keyboard type attached and the preferred language that the Pi will use, etc.
4. Click on Install and let setup complete and reboot.

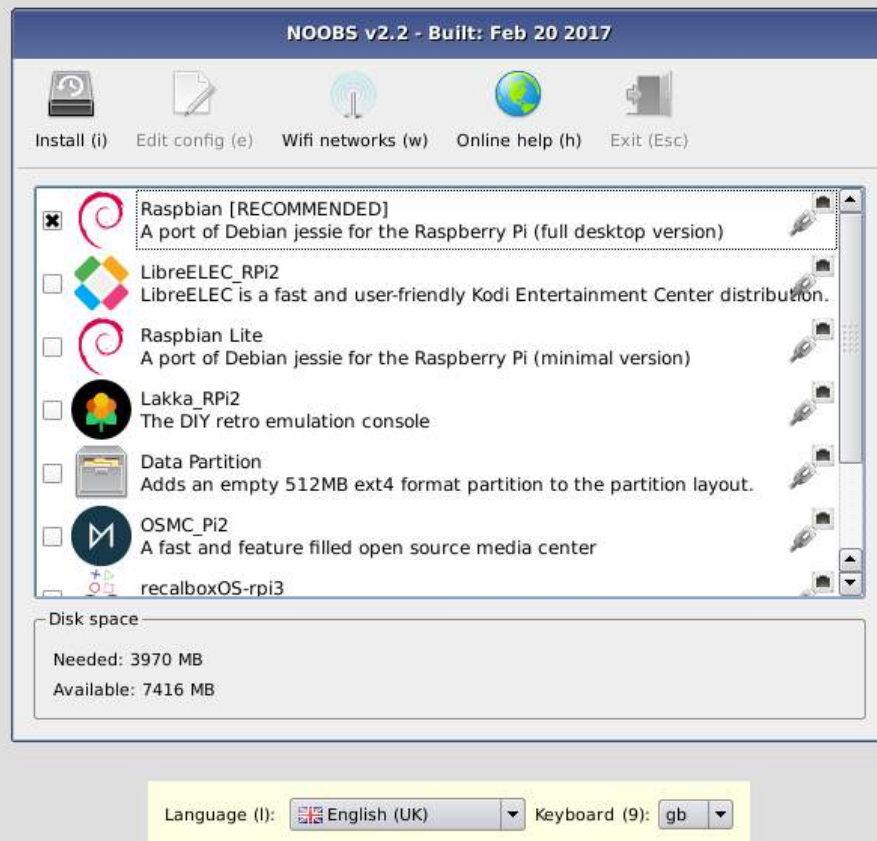


Figure 18

Image downloaded from: <https://www.raspberrypi.org/documentation/installation/noobs.md>

You will be presented with the Raspbian desktop as shown in Figure 19.

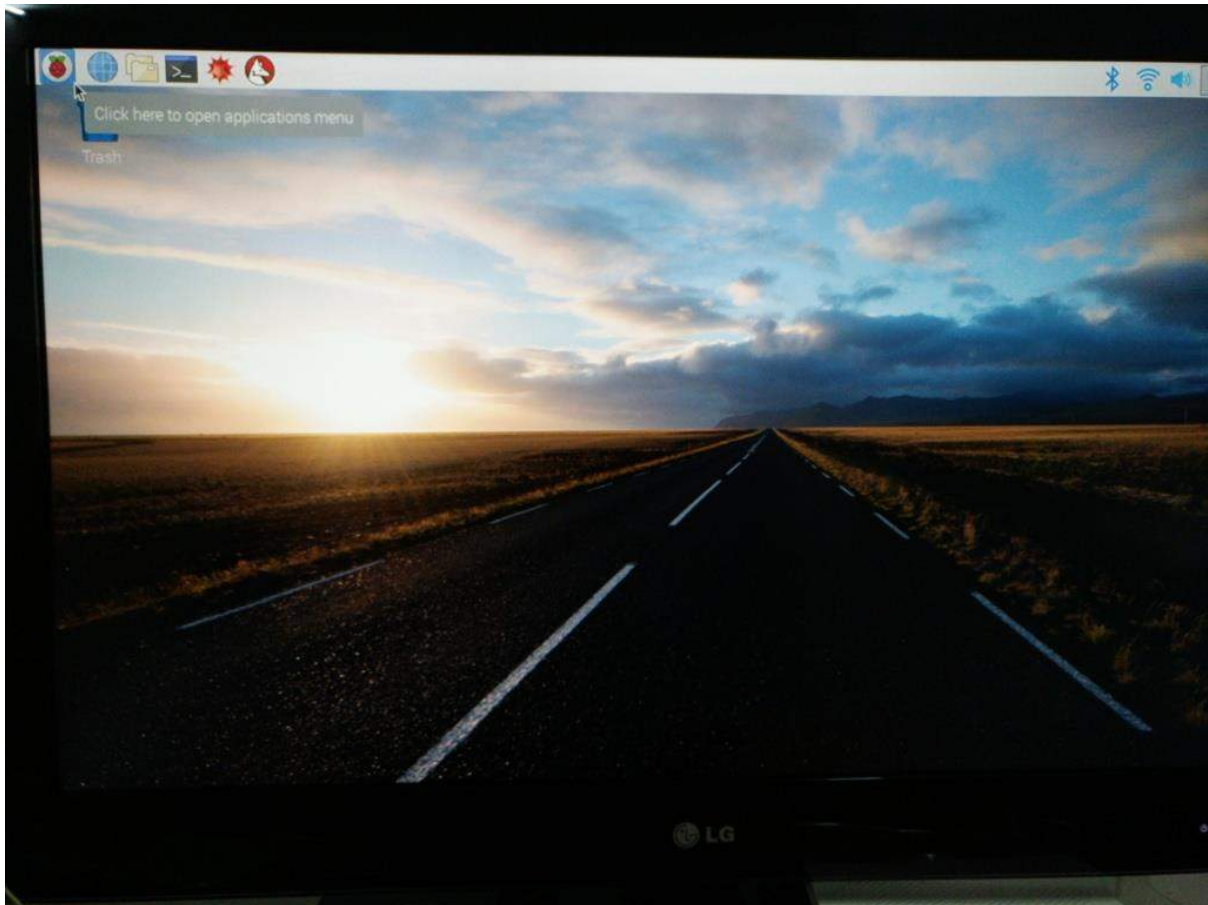


Figure 19

In case you are presented with the terminal and prompted for username and password, enter **pi** as username and **raspberry** as password. Then type **startlxde** to load the Pi desktop.

From there you will have options to navigate to different applications such as the chromium web browser, the terminal and other preinstalled applications.

As with many Linux distributions (Raspbian is based on Debian), most of the work is done through the terminal.

Open the terminal by clicking on the terminal icon on the taskbar.

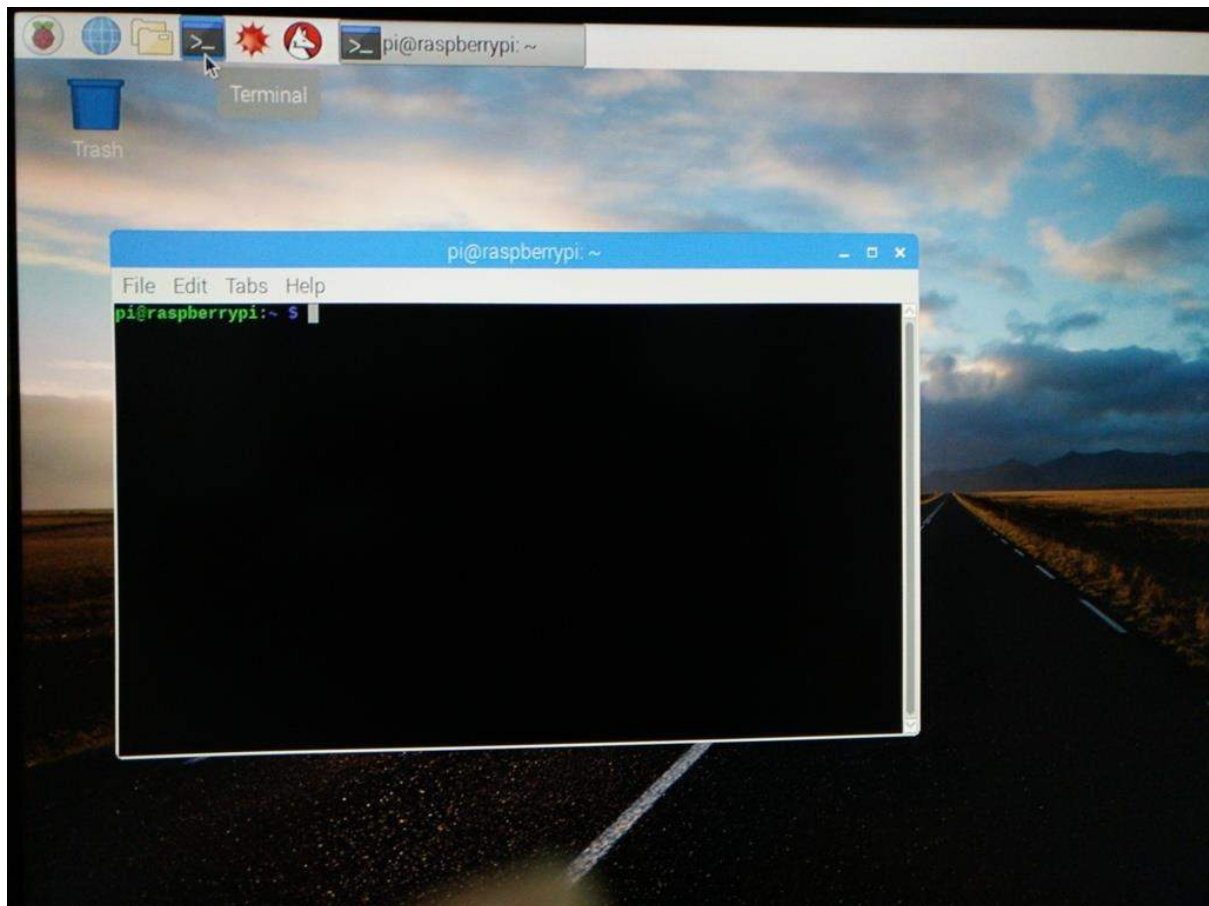


Figure 20

II. Update and Upgrade

Once the Raspberry Pi has been configured, it is important to update the OS files and libraries.

Open the terminal and type the following commands:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

to complete the updates and upgrades.

III. Installing a Web Server

Without a web server, Nextcloud cannot run. We will install the Apache web server and the required server-side scripting engine and a DBMS (optional) for the web server to function correctly.

Run the following commands to:

a) Install Apache

```
sudo apt-get install apache2
```

b) Install PHP 7 (server-side scripting engine)

```
sudo apt-get install php7.0 php7.0-gd sqlite php7.0-sqlite php7.0-curl
```

c) Install the DBMS MySQL (optional)

```
sudo apt-get install mysql-server python-mysqldb php-mysql
```

d) Creating new user to access MySQL (recommended if MySQL installed)

```
sudo mysql --user=root mysql
```

Then run the following command at the prompt for creation of an admin user

```
CREATE USER '<AdminUser>'@'localhost' IDENTIFIED BY '<AdminUser's Password>';  
GRANT ALL PRIVILEGES ON *.* TO '<AdminUser>'@'localhost' WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```

Note: that <AdminUser> and <AdminUser's Password> are parameters you need to specify.

e) Restart Apache (for changes to take effect)

```
sudo service apache2 restart
```

f) Check web server

Open the Chromium web browser and type the following in the address bar

```
http://localhost
```

You should be presented with the following HTML

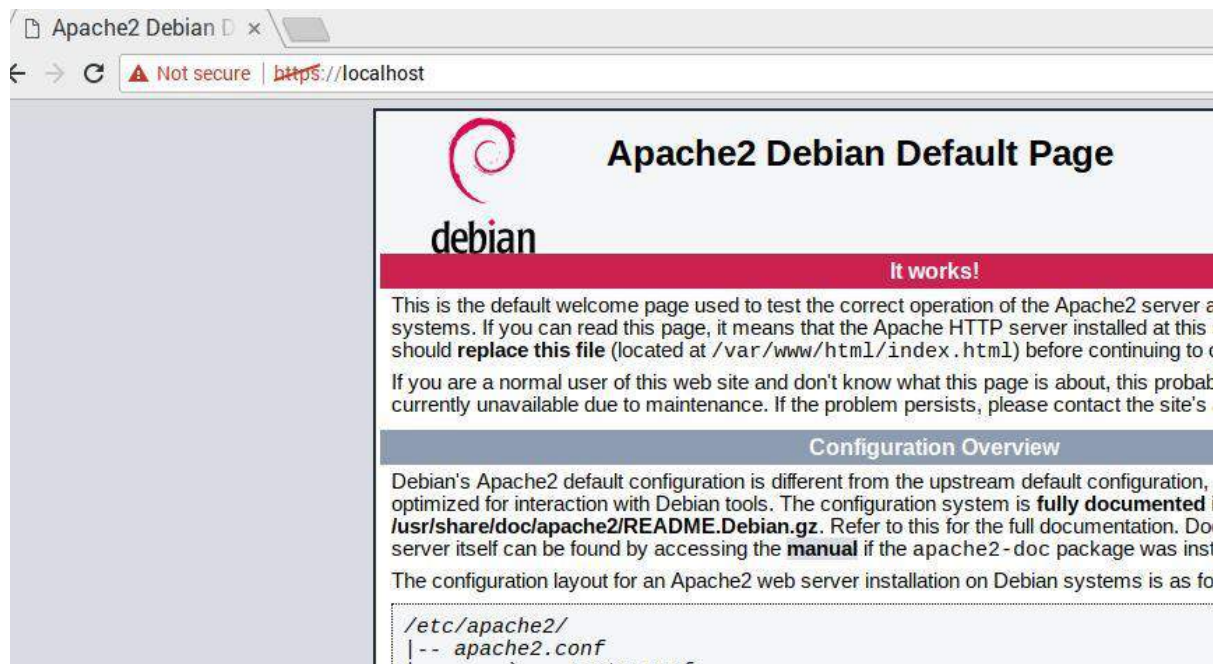


Figure 21

IV. Installing Nextcloud

Setting up Nextcloud requires that your web server is up and running. It involves downloading of the Nextcloud archive, extracting the contents in the web server's document root directory and creating the data structure for Nextcloud to operate correctly.

Run the following commands to:

a) Download and extract Nextcloud in the web server's document root folder (DRF)

```
cd /var/www/html (this will point to the DRF)  
sudo wget https://download.nextcloud.com/server/releases/nextcloud-12.0.3.zip  
sudo unzip nextcloud-12.0.3.zip (this will extract nextcloud in the DRF)
```

b) Create data structure and give users the proper permissions

```
cd /var/www/html/nextcloud  
sudo mkdir -p /var/www/html/nextcloud/data (create operational directory for nextcloud)  
sudo chown www-data:www-data /var/www/html/nextcloud/data (give users permission to use data)  
sudo chmod 750 /var/www/html/nextcloud/data  
sudo chown www-data:www-data config apps (propagate permissions to config and apps directory)
```

c) Check if Nextcloud has been installed properly

Type <http://localhost/nextcloud> in the Chromium web browser.

The initial Nextcloud setup page should be displayed as shown below:

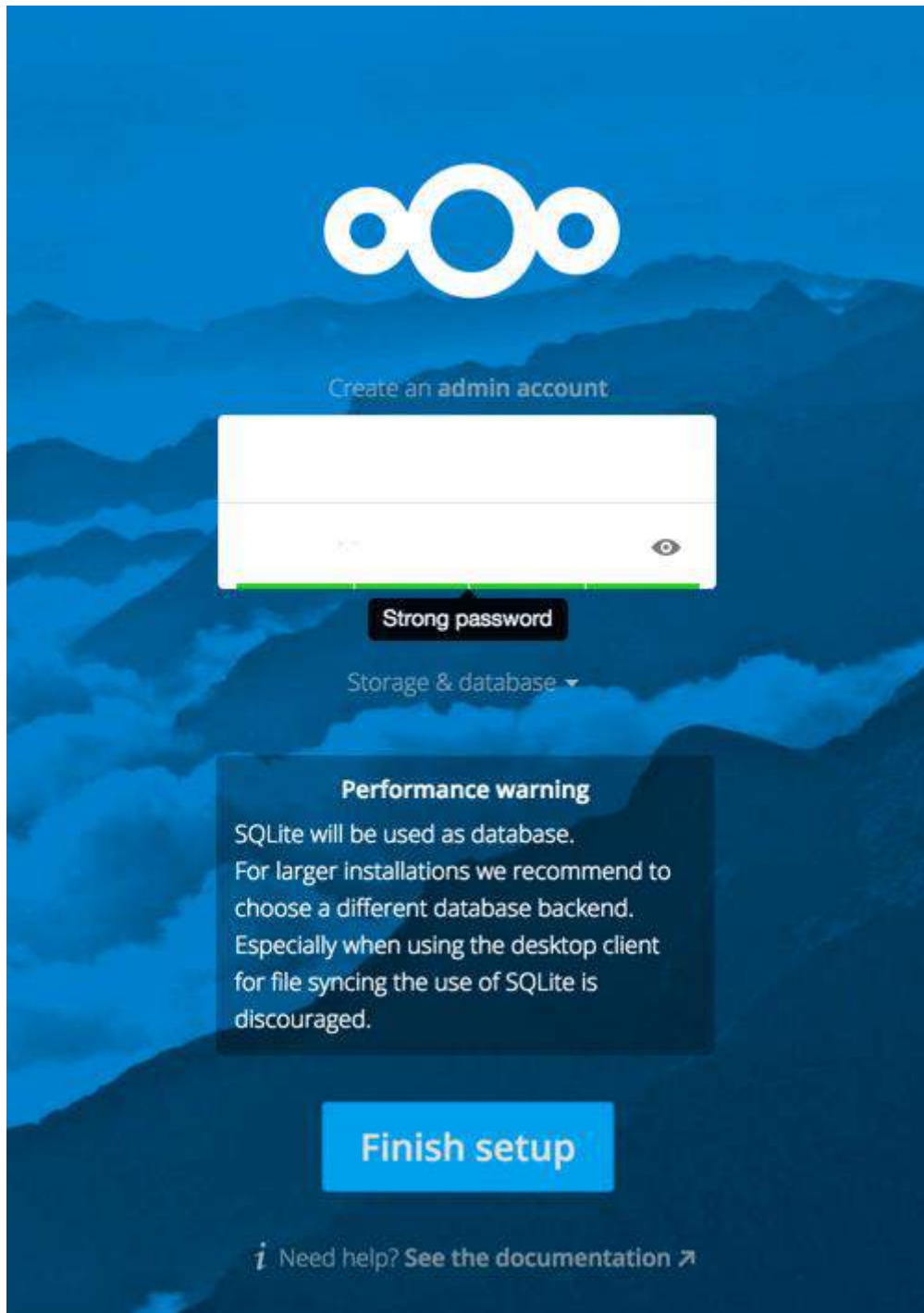


Figure 22

Picture taken and modified from

https://docs.nextcloud.com/server/12/admin_manual/installation/installation_wizard.html

Enter the admin username and password you intend to use and conclude setup by clicking on the finish setup button.

Once this is done, you will be presented with the following screen. Click on the close button to go on your Nextcloud home.



Figure 23

Nextcloud home shown below:

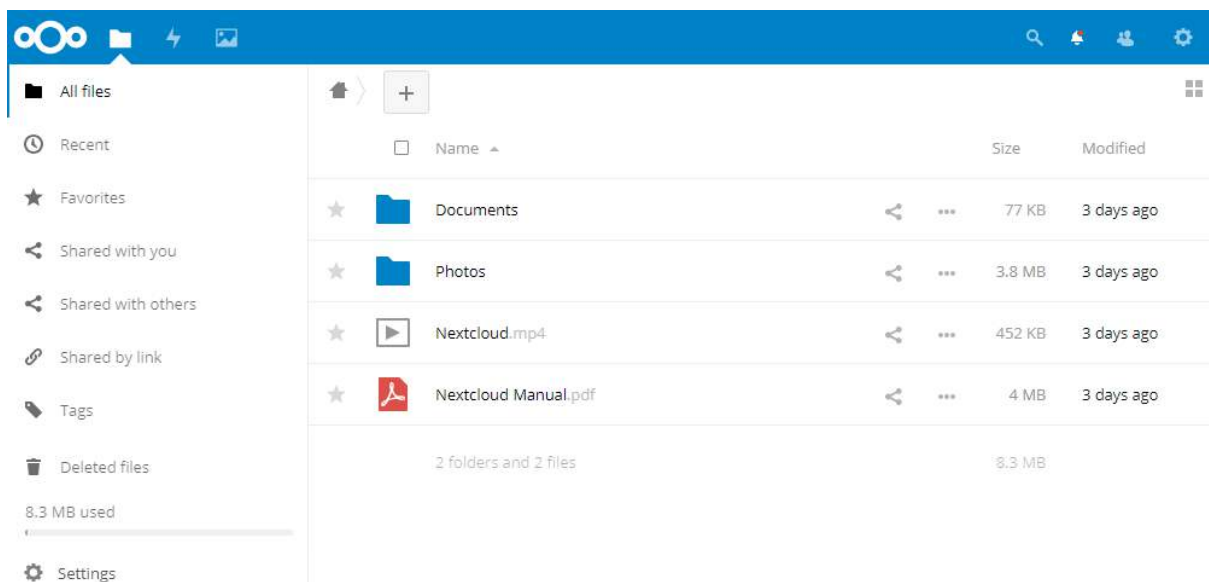


Figure 24

You can upload pictures or documents or any files you wish. There are other apps which you can add to Nextcloud, but covering the functionalities of Nextcloud is beyond the scope of this module.

V. Increasing the Upload/Download Size

With the default setup, you will not be able to upload file greater than 2 MB which is extremely inconvenient.

To increase this limit and accommodate larger files, type the following code to change the relevant PHP settings

```
sudo nano /etc/php/7.0/apache2/php.ini
```

Search for the following lines in the php.ini

```
post_max_size = 8M (change to 50M)  
upload_max_filesize = 2M (change to 50M)
```

8M represents 8 megabytes. Change both values to 50M or more. Save the file and restart apache for the changes to take effect by running the code below

```
sudo service apache2 restart
```

VI. Allow Apache Configuration Override

Edit the apache configuration

```
sudo nano /etc/apache2/apache2.conf
```

Change the following:

```
<Directory /var/www/>  
    Options Indexes FollowSymLinks  
    AllowOverride None  
    Require all granted  
</Directory>
```

To:

```
<Directory /var/www/>  
    Options Indexes FollowSymLinks  
    AllowOverride ALL
```

```
Require all granted
</Directory>
```

Save file and exit

Restart Apache for changes to take effect

```
sudo service apache2 restart
```

Now you can take a picture and upload in the photos folder in Nextcloud. The picture will be kept in your own personal cloud.

At this point, Nextcloud is only accessible on your LAN. The fundamental principle of any cloud system is to be accessible anywhere, anytime if a suitable internet connection is available. In the next step, we will make the Raspberry Pi accessible outside your home LAN and visible to the world.

CLOUD SERVICE VISIBLE TO THE WORLD

This section assumes that you have a connection with dynamic IP allocation such as an internet connection provided by your ISP through a home router.

This involves:

- i. Giving the Raspberry Pi a static address so that port forwarding can be configured.
- ii. Implement port forwarding so that the home router can forward http/https requests (using default ports 80/443) to the web server running on your Raspberry Pi on your local LAN.
- iii. Setting up Dynamic DNS service so that the Raspberry Pi can still receive requests if the router's IP is updated by the ISP.

Step 1: Setting up Static IP

Configuring a static IP requires the IP addresses of the gateway and the DNS (usually DNS address is same as the address of the gateway).

Type the following commands:

```
ip route | grep default | awk '{print $3}' this will display the IP address of the router -- (1)  
cat /etc/resolv.conf this will list the IP address of the nameserver (DNS) -- (2)
```

Copy both of these addresses. We are going to use them in the dhcpd.conf file to set the static IP.

Modify the dhcpd.conf file to add the static address.

```
sudo nano /etc/dhcpd.conf
```

Add the following lines:

```
interface wlan0  
  
static ip_address= [static IP]/24 (the first 3 numbers of this IP taken from (1) and the last number between 100 and 200 e.g., 192.168.1.127 provided your gateway is 192.168.1.1)  
  
static routers= ip address found in (1)  
  
static domain_name_servers= ip address found in (2)
```

Save and reboot the Pi.

Now the Pi should be automatically assigned the static IP address you have specified.

You can check if the assigned IP address is correct by running the following:

```
ip -4 addr show | grep global
```

[x7]

Step 2: Implementing Port Forwarding

This step is important so that your router makes the Pi accessible to the internet.

Type the IP address of your gateway (*found in (1)*) in your browser on your laptop connected to your LAN.

This should bring up the login page for your router.

Enter the username and password provided by your ISP to get access to the router.

Navigate through the router's configuration till you find port forwarding section.

Here, you need to add **two** rules for forwarding/port mapping http (port 80) data and https (port 443) to your Pi as shown below.

	WAN Name	Mapping Name	Protocol	External Port	Internal Port	Internal Host	Enable
<input type="checkbox"/>	1_INTERNET_██████████	RaspberryPi	TCP	443-443	443-443	192.168.██████████	Enable
<input type="checkbox"/>	1_INTERNET_██████████	RaspberryPiUpdat.....	TCP	80-80	80-80	192.168.██████████	Enable

Enable Port Mapping:	<input checked="" type="checkbox"/>		
WAN Name:	1_INTERNET_R_VII.▼	Protocol:	TCP ▼
Start External Port:	443 *	End External Port:	443 *
Start Internal Port:	443 *	End Internal Port:	443
Start External Source Port:	0	End External Source Port:	0
Mapping Name:	RaspberryPi	External Source IP Address:	
Internal Host:	192.168.██████████ *	raspberrypi ▼	

Figure 25

The WAN Name is from your router configuration which has been set by your ISP.

First create a rule for port 80, set both the external and internal ports to **80**, the protocol to **TCP** and the internal host to the **static IP** you have given to your Pi.

Create a second rule and follow the same steps except specify the port **443** instead of 80.

Apply changes and reboot both your router and your Pi.

Step 3: Setting up dynamic DNS

It is highly unlikely that ISPs will give a static IP for your router. Consequently, you have to set up a dynamic DNS so that every time your IP is refreshed by your ISP, you can still connect to your Pi outside of your LAN.

This involves setting up:

1. A DDclient which will update your external IP on a Dynamic DNS Network Services provider such as No-IP (www.noip.com).
2. An account with Dynamic DNS Network Services provider. We will use No-IP in our case.
3. The DDclient to use the No-IP hostname.
4. The DDclient to run as a daemon so that it periodically checks if the external IP has been refreshed and notify the Dynamic DNS Network Services provider accordingly.

Step 3.1: Setting up the DDclient

Run the following commands:

```
sudo apt-get update  
sudo apt-get install ddclient libjson-any-perl
```

Get updated DDclient and extract the compressed files

```
wget http://downloads.sourceforge.net/project/ddclient/ddclient/ddclient-3.8.3.tar.bz2  
tar -jxvf ddclient-3.8.3.tar.bz2
```

Overwrite the current installed version with the updated version

```
sudo cp -f ddclient-3.8.3/ddclient /usr/sbin/ddclient  
sudo mkdir /etc/ddclient  
sudo mv /etc/ddclient.conf /etc/ddclient
```

Open the DDclient configuration

```
sudo nano /etc/ddclient/ddclient.conf
```

Delete everything and paste the following code to allow the DDclient to use of SSL.

```
use=web, web=checkip.dyndns.com/, web-skip='IP Address'  
ssl=yes
```

Step 3.2: Create an account with Dynamic DNS Services provider, No-IP

Go to www.noip.com to create a free account.

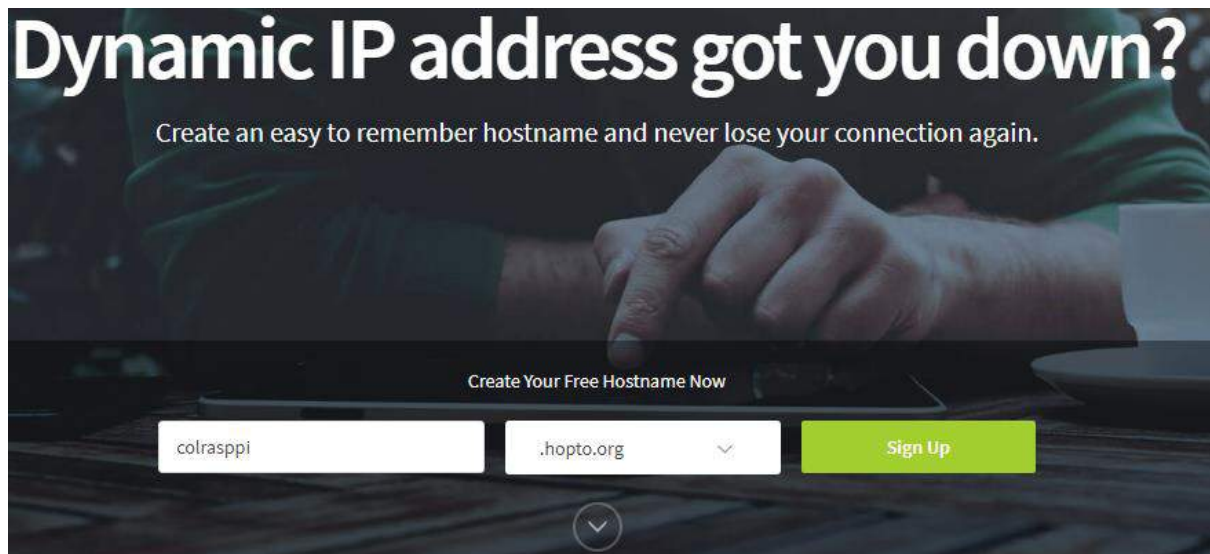


Figure 26

Write the hostname to represent your Pi (use a unique name as far as possible). In the above example I have used **colrasppi** as hostname. Click on the Sign Up button to proceed to the creation of the No-IP account as shown below:

The image shows the "Create Your No-IP Account" form. At the top left is the title "Create Your No-IP Account" and at the top right is a user icon with a plus sign. Below the title, on the right, is a red asterisk icon followed by the text "Indicates required fields". The form consists of three main input fields: "Email", "Password", and a combined field for "hostname" and "domain". The "Email" and "Password" fields each have a red asterisk icon to their right. The "hostname" field contains "colrasppi" and the "domain" dropdown menu is set to ".hopto.org". At the bottom left of the form is a checkbox labeled "Create my hostname later".

Figure 27

Add a valid email address and a password to login into your No-IP account and click on Free Signup.

You will receive an email to activate your newly created account. Once activated, you can log in to your No-IP account and you will find the new hostname you can use to get access to your Pi from anywhere on the internet.

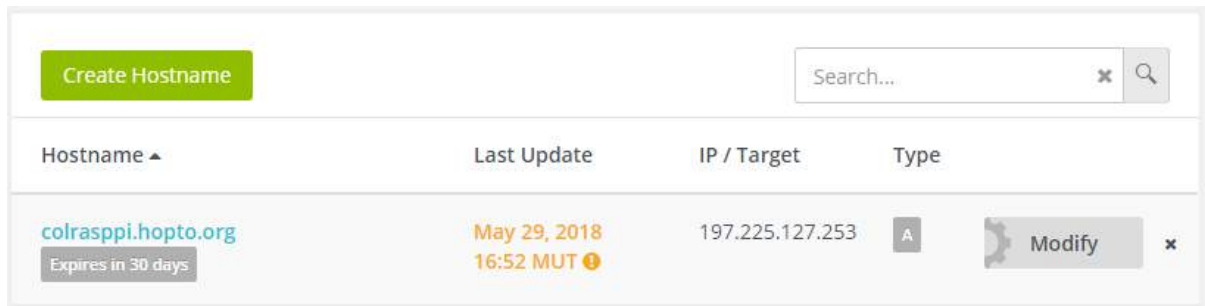


Figure 28

Note that the free service expires after 30 days. You can either upgrade or create the hostname again.

Step 3.3: Update DDclient Configuration to Use No-IP Hostname

Edit the DDclient configuration file:

```
sudo nano /etc/ddclient/ddclient.conf
```

Enter the following details at the end of the file

```
protocol=dyndns2
server=dynupdate.no-ip.com
login=your_username (used to login to No-ip.com)
password=your_password (used to login to No-ip.com)
colrasppi.hopto.org
```

Restart the DDclient so the changes take effect

```
sudo /etc/init.d/ddclient restart
```

Step 3.4: Run DDclient as a daemon

Execute the command.

```
sudo nano /etc/default/ddclient
```

Check that the following directives are active (not commented) and values are correct. If these are not found, add them and save.

```
run_daemon="true"
run_dhclient="false"
run_ipup="false"
```

Start the ddclient service and check if it is running correctly.

```
sudo service ddclient start
sudo service ddclient status
```

Force ddclient to update IP at least once a week (some DNS service require that the IP be updated frequently, otherwise they might stop providing the DNS service to the particular host).

Execute the following:

```
sudo nano /etc/cron.weekly/ddclient
```

Add the following and save

```
#!/bin/sh
/usr/sbin/ddclient -force
```

Give script permission to execute

```
sudo chmod +x /etc/cron.weekly/ddclient
```

Check if service is running correctly

```
sudo service ddclient status
```

Now you should be able to access your Nextcloud instance by writing the following in your web browser on any machine outside your LAN.

`http:// colrasppi.hopto.org/nextcloud/`

VII. Accessing your Nextcloud instance via Mobile Phone

The Nextcloud client is available for both iOS and Android platforms. Go to Apple/Google store on your smartphone and Install Nextcloud. Once installed, provide the following link when prompted

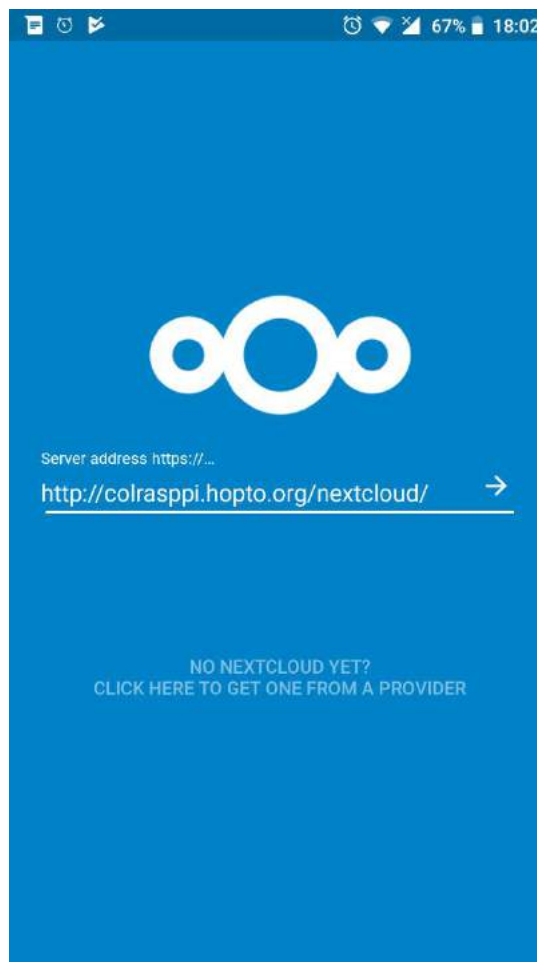


Figure 29

Nextcloud will ask you to grant access to your mobile. Click on grant access and you will be presented with the login screen.

Insert your Nextcloud username and password and you will be redirected to the Nexcloud home as shown below:

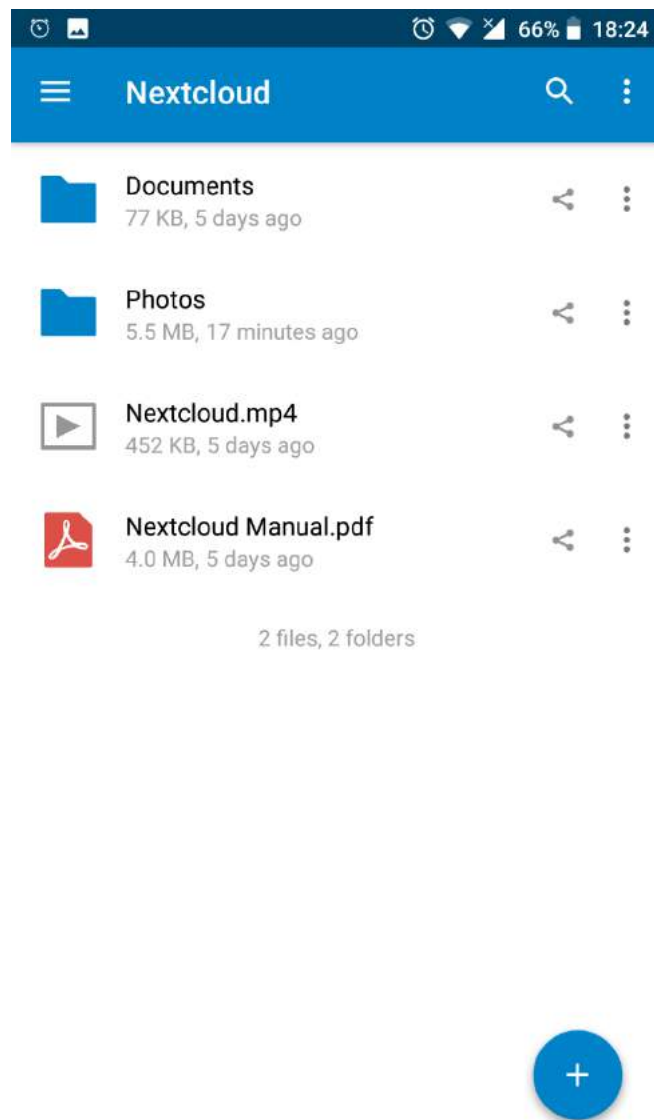


Figure 30

You can now use your Nextcloud instance from your mobile and upload documents, pictures, etc.

TOPIC 3: SECURITY AND PRIVACY FOR IOT/CLOUD COMPUTING

Case study: Securing the Raspberry Pi which holds our Nextcloud instance.

The Pi system is now visible to the whole internet and with that comes all the privacy and security issues which is not present in your local LAN.

To enhance security and privacy, several measures could be implemented such as:

- i. Moving the Nextcloud data from the web server's document folder to a more secure location
- ii. Make use of https which encrypts data to a certain level, thus making it harder for hackers to intercept and decipher the data.
- iii. Make use of software firewalls to control access to the Pi.

I. Moving the Nextcloud data from the web server's document folder to a more secure location

Create a folder nextcloud in the var directory

```
sudo mkdir -p /var/nextcloud
```

Move Nextcloud data to the newly created folder by executing the following:

```
sudo mv -v /var/www/html/nextcloud/data /var/nextcloud/data
```

Change Nextcloud configuration to reflect to new location

Open the Nextcloud configuration file:

```
sudo nano /var/www/html/nextcloud/config/config.php
```

Change the value of datadirectory to /var/nextcloud/data as shown below:

```
'datadirectory' => '/var/nextcloud/data'
```

Save and exit.

II. Setting up SSL (using a self-signed certificate) to Communicate in https instead

Create folder to store the new SSL certificate

```
sudo mkdir -p /etc/apache2/ssl
```

Generate certificate using openssl

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -  
out /etc/apache2/ssl/apache.crt
```

You will be prompted for information to be included in the certificate such as Organization Name, Email address, etc. Fill-in all the details and the certificate will be generated.

Enable SSL module for Apache

```
sudo a2enmod ssl
```

Modify Apache SSL configuration to use the certificates generated in the SSL folder.

```
sudo nano /etc/apache2/sites-available/default-ssl.conf
```

Change the value of *SSLCertificateFile* and *SSLCertificateKeyFile* as shown below:

```
SSLCertificateFile /etc/apache2/ssl/apache.crt  
SSLCertificateKeyFile /etc/apache2/ssl/apache.key
```

Save and Exit.

Enable SSL configuration and restart Apache

```
sudo a2ensite default-ssl.conf  
sudo service apache2 restart
```

Test https by typing the following:

```
https://<static ip of raspberry Pi>/nextcloud/
```

You will get a warning about invalid certificate (since it is self-signed). Add to exception and proceed.

To enhance security, enforce SSL by redirecting http connections to https

Open Apache configuration

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Replace everything with the following rules:

```
<VirtualHost *:80>

    ServerAdmin col@mysite.com

    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^(.*)$ https://%{HTTP_HOST}$1 [R=301,L]

</VirtualHost>
```

Save and exit.

Enable redirect and restart Apache

```
sudo a2enmod rewrite

sudo service apache2 restart
```

Test https redirection with the following:

```
http:// colrasppi.hopto.org/nextcloud/
```

It should change the http automatically to https.

III. Monitoring User Login Using Fail2Ban

Fail2Ban is an intrusion prevention software which scans log files (e.g. /var/log/apache/error_log) and bans IPs which show malicious signs such too many password failures, etc. *[x2]*

Install Fail2Ban

```
sudo apt-get install fail2ban
```

✓ **Change Banning Rules**

Run the following commands to copy default rules to a copy that we can edit with our own rules as shown below:

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

Edit the new *jail.local* to change rules.

```
sudo nano /etc/fail2ban/jail.local
```

You can change the default bantime to more than 600 (e.g., bantime = 900 will result in a bantime of 15 minutes instead of the default 10 minutes).

You can check how to configure Fail2Ban here:

<https://www.digitalocean.com/community/tutorials/?q=fail2ban>

Once you have made changes, restart the service by executing the following:

```
sudo service fail2ban stop  
sudo service fail2ban start
```

IV. Restrict Connection to Pi using UFW Firewall

Install UFW (Uncomplicated FireWall)

```
sudo apt-get install ufw
```

Set firewall rules to allow SSH from all devices on the LAN and any http/https connection to the Pi from any IP.

```
sudo ufw allow from <your-router-ip>/24 to any port 22  
sudo ufw allow http  
sudo ufw allow https
```

Enable firewall

```
sudo ufw enable
```

The following message is displayed: Command may disrupt existing ssh connections. Proceed with operation (y|n)?

Press Y and continue.

The firewall is now active.

SUMMARY

In this unit, you have learned how to use IoT devices to generate and send data to the cloud for processing, thereby visualising the trends in the data. You have also acquired the necessary knowledge to mount and secure your own cloud service, which is accessible on the internet.

REFERENCES

- [x1] Botta A, De Donato W, Persico V, Pescapé A, “Integration of Cloud computing and Internet of Things: A survey”, 2015.

- [x2] Fail2Ban Main Page, January 2008; Updated May 2016
Available at: https://www.fail2ban.org/wiki/index.php/Main_Page
Accessed on 30 May 2018

- [x3] How To Setup a Firewall with UFW on an Ubuntu and Debian Cloud Server, June 2013
Available at: <https://www.digitalocean.com/community/tutorials/how-to-setup-a-firewall-with-ufw-on-an-ubuntu-and-debian-cloud-server>
Accessed 03 June 2018

- [x4] Building a Raspberry Pi based Personal Cloud Server, November 2017; Updated December 2017
Available at: <http://techmantium.com/raspberry-pi-raspbian-cloud-server/>
Accessed on 20 June 2018

- [x5] How to Setup a Raspberry Pi Nextcloud Server, June 2017; Updated November 2017
Available at: <https://pimylifeup.com/raspberry-pi-nextcloud-server/>
Accessed on 25 May 2018

- [x6] Raspberry Pi Port Forwarding & Dynamic DNS, April 2015; Updated May 2018
Available at: <https://pimylifeup.com/raspberry-pi-port-forwarding/>
Accessed on 28 May 2018

- [x7] How do I set up networking/WiFi/static IP address?, November 2015
Available at: <https://raspberrypi.stackexchange.com/questions/37920/how-do-i-set-up-networking-wifi-static-ip-address/37921#37921>
Accessed on 28 May 2018

[x8] Securing your Raspberry Pi, [No creation date]

Available at: <https://www.raspberrypi.org/documentation/configuration/security.md>

Accessed on 29 May 2018

[x9] Learn More About ThingSpeak.

Available at https://thingspeak.com/pages/learn_more

Accessed 03 June 2018

[x10] Code listing 1 & 2 adapted from: LM35 with Esp8266 Thingspeak, December 2015

Available at: <https://www.youtube.com/watch?v=BwDz6zHbGLI&feature=youtu.be>

Accessed on 03 June 2018

[x4-x6] Covers the whole of Units 2 & 3 in the document