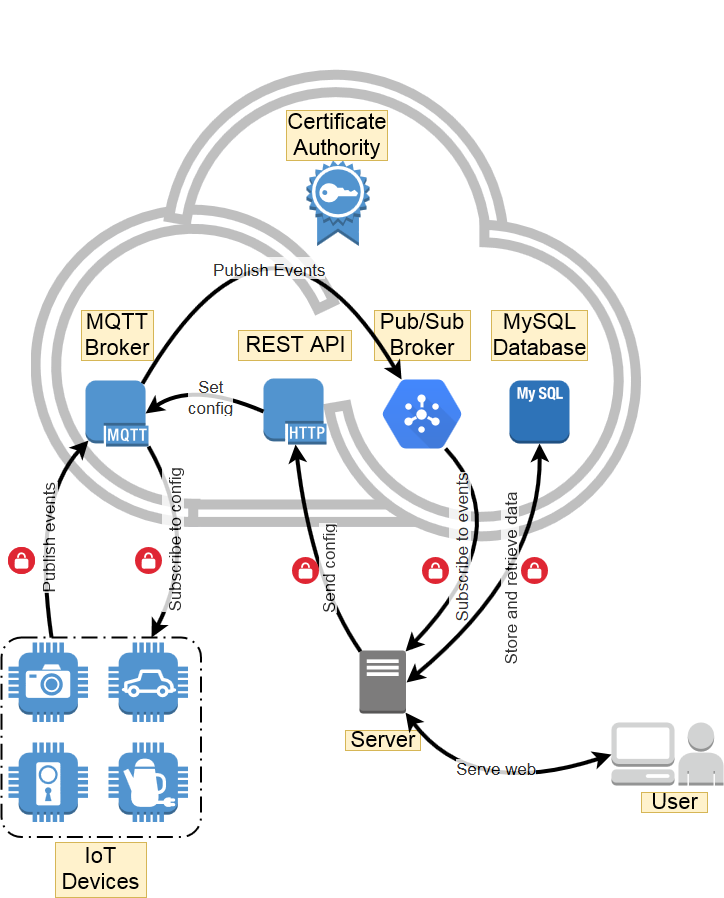
What is this tutorial for

This tutorial will help you get through the technical stages necessary to set up this infrastructure that was taught in class.



Upon completing this tutorial you should have a google account with all of the relevant entities properly setup and example code that performs all of the communications depicted by the arrows.

What do you need for this tutorial

Things you need to have

1. A device to become an IoT device, such as a Raspberry Pi.
2. A computer that will act as a server.
3. A Google account (not at post.bgu.ac.il).
4. A credit card - used only for verification and will not be charged unless you ask for a charge.

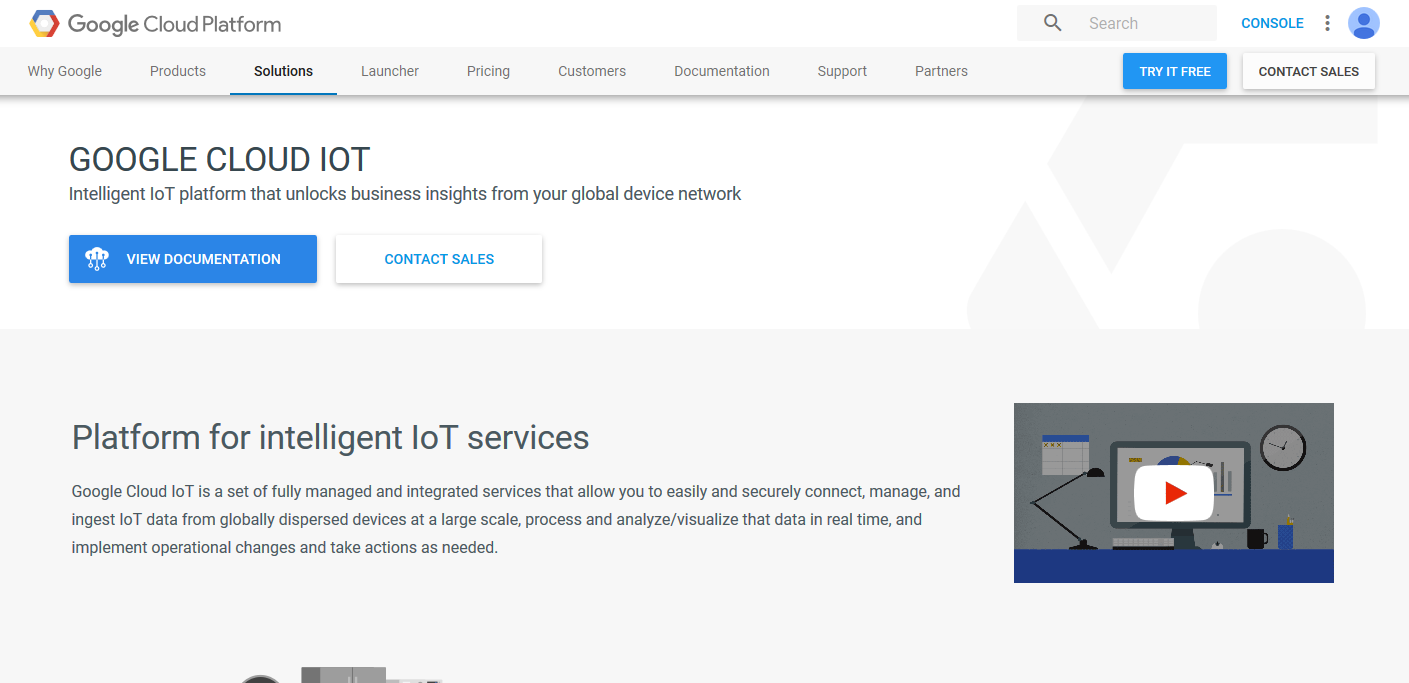
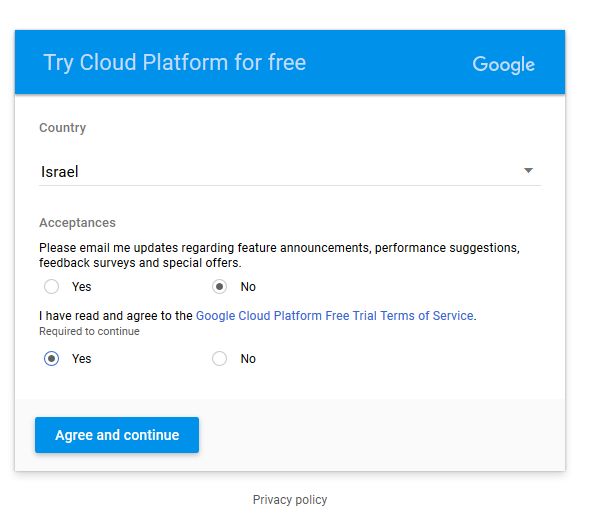
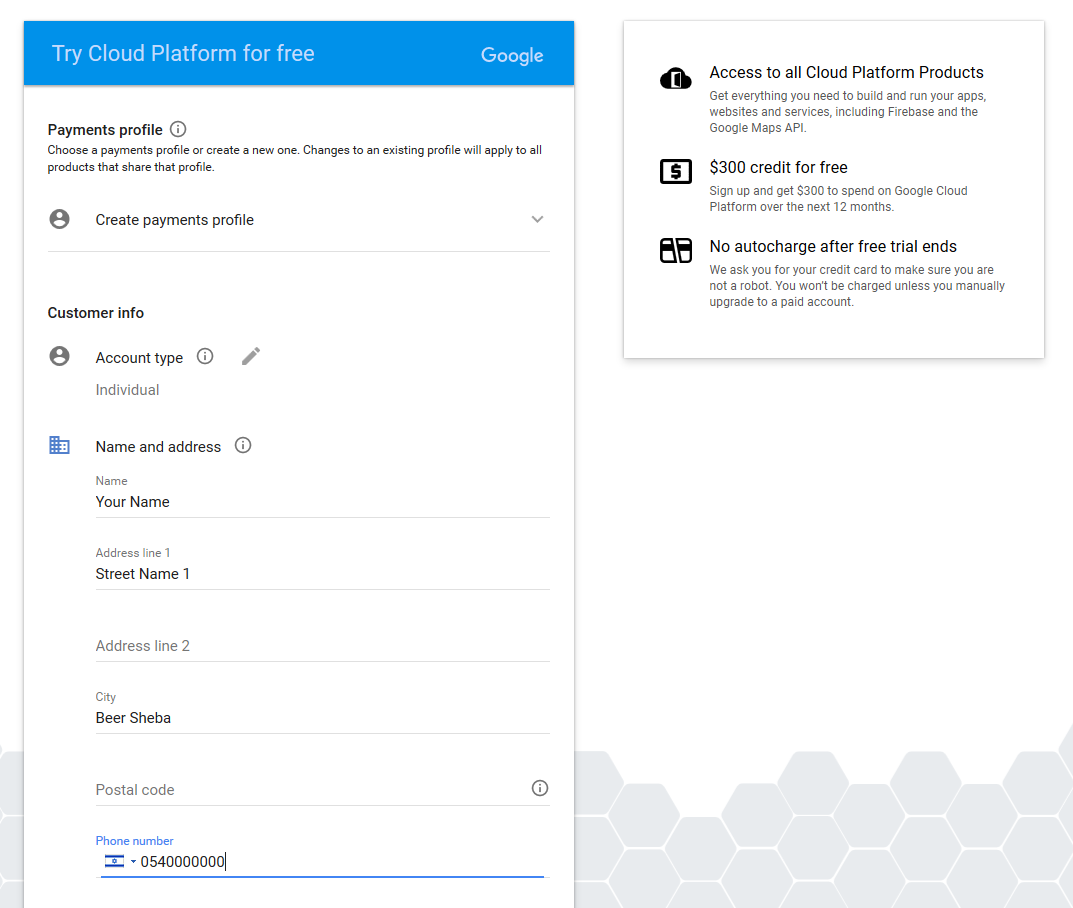
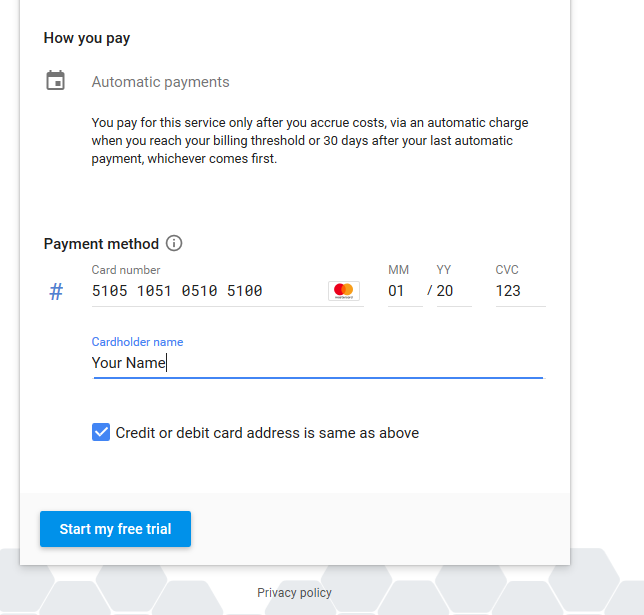
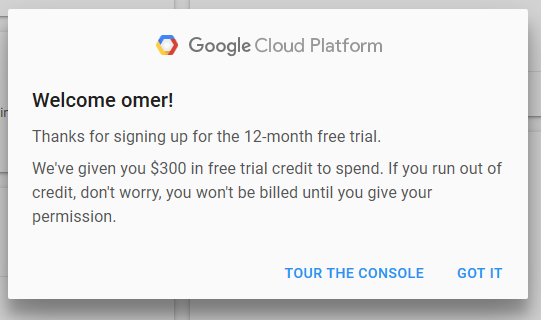
Things you need to know

1. SQL - We will use the MySQL server.
2. Networking concepts - DNS, TCP, HTTP/S.
3. Linux - basic shell commands, editing files, transferring files to/from a linux machine.
4. Python/Java - code examples are in python and using it is highly recommended.
5. Basic web technologies - PHP, HTML, CGI.
6. Public/Private key encryption concepts - including certificates.

**You don’t have to master all of the above, but when you encounter issues in these topics you are expected to try and solve them by yourself.**

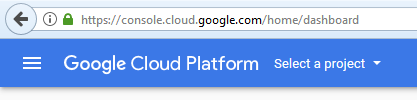
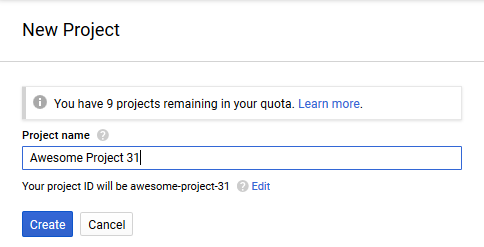
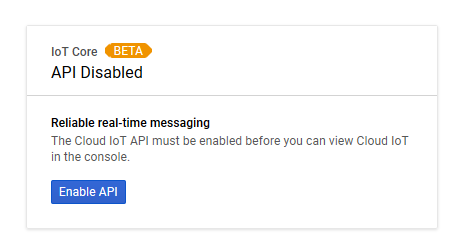
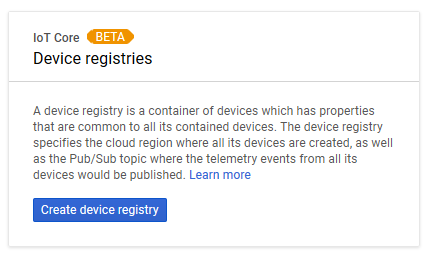
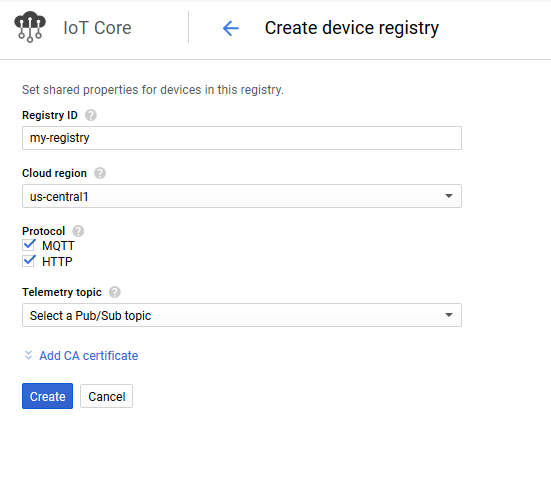
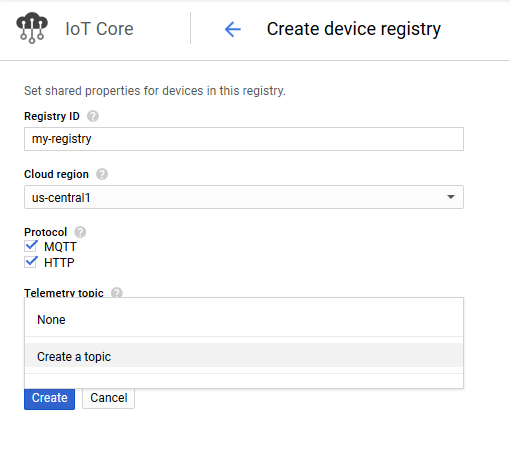
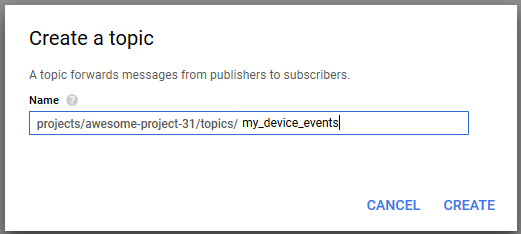
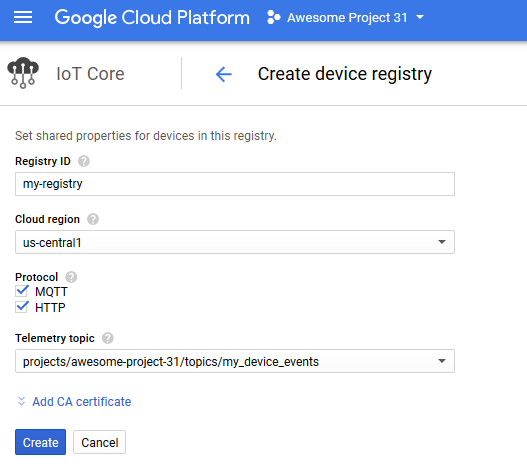
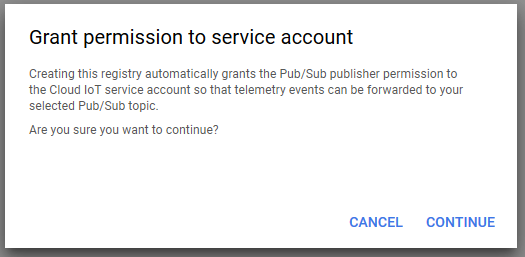
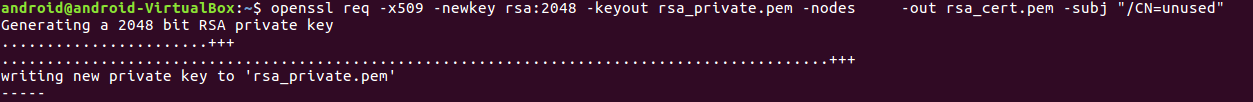
## 

Registering to a free trial of Google Cloud IOT

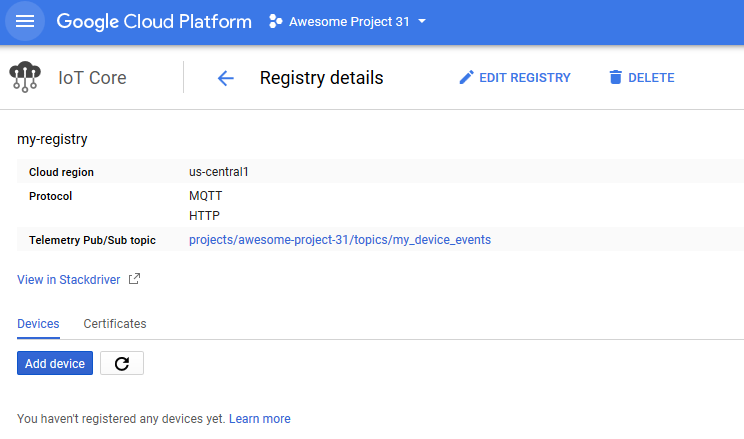
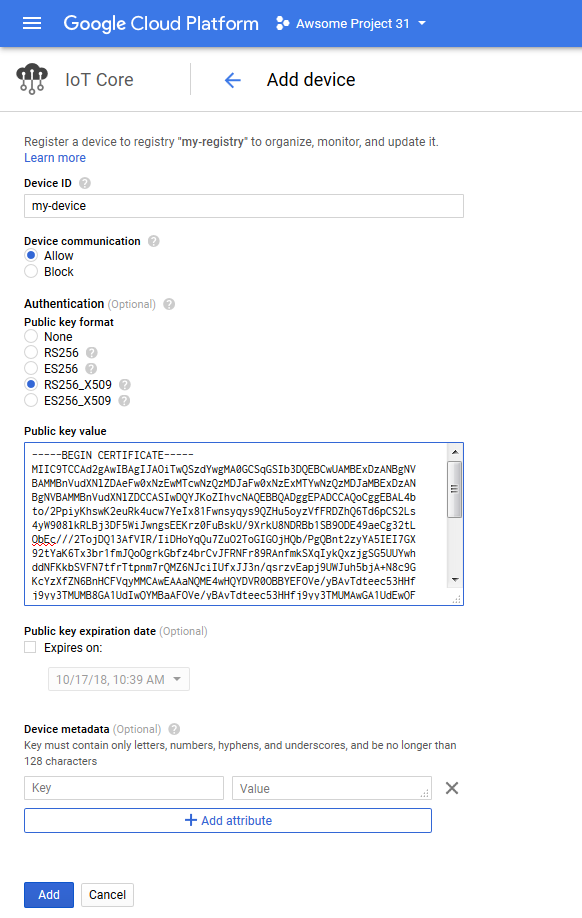
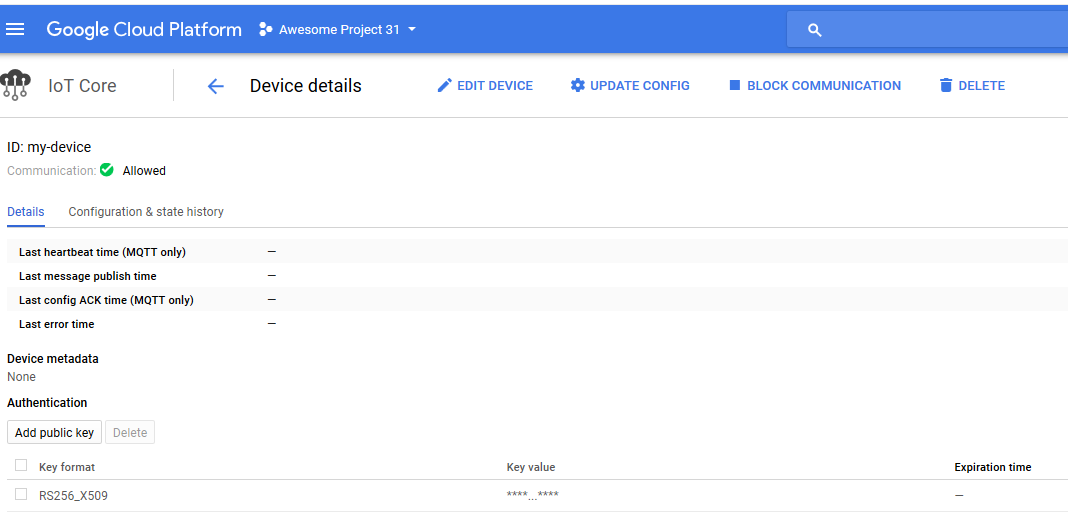
1. You will need a Gmail account, preferably not the post.bgu.ac.il account.  
   Log into your google account.
2. Go to <https://cloud.google.com/solutions/iot/> and click the “Try It Free” Button.  
   
3. Fill the form as depicted below and click “Agree and continue”.  
   
4. In order to activate your account, google requires a valid credit card number for verification reasons. **Google will not use or bill your credit card even when the trial ends. The trial is free.**  
   Select “Create payments profile” and fill in your details and credit card information.  
   You may also use an existing payment profile if you already have one.  
     
     
   
5. You should have a Google Cloud account now, with 300$ in credit for 1 year.  
   

## 

Creating a project and adding a device

1. Go to <https://console.cloud.google.com/home/dashboard> and click “Select a project”.  
   
2. Click the ‘+’ Button to add a project.  
   
3. Give your project a name, the name must be unique in all of Google Cloud. Note the project ID displayed below, this will be used greatly in the future. Creating might take a couple of minutes.  
   
4. Go to <https://console.cloud.google.com/iot/>  
   If you get an “API Disabled” message click “Enable API”. A bug might cause this message to appear twice, click it twice then.  
   
5. Click “Create device registry”, this will be the group of devices connected to your project.  
   
6. Choose a registry ID. You will use this ID in the future.  
   Select us-central1 as the Cloud region.  
   Enable both MQTT and HTTP protocols.  
   Click "Select a Pub/Sub topic" and select "Create a topic".  
     
   
7. Give your topic a name. You will use this name in the future.  
   
8. When everything looks like the form below, click "Create".  
   
9. If you encounter this message, click "Continue".  
   
10. Update the date on your raspberry pi like this:  
    sudo date -s "12 DEC 2017 14:20:00"
11. Now that you have a registry and a topic, you need to add your device. But first, you must create a key pair for your device to authenticate with.  
    In a linux machine (preferably on your Raspberry pi), type in the following command:  
      
    openssl req -x509 -newkey rsa:2048 -keyout rsa\_private.pem -nodes -out rsa\_cert.pem -subj "/CN=unused"  
      
    
12. Two files had been created:
    1. **rsa\_cert.pem** that contains the device’s public key.
    2. **rsa\_private.pem** that contains the device’s private key. Keep this file safe.

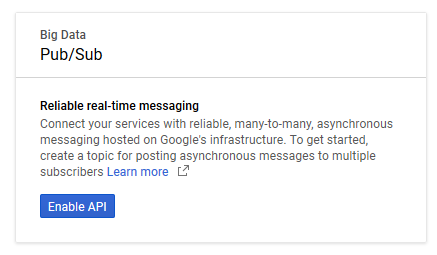
Copy the contents of rsa\_cert.pem.  


1. In your browser, Click “Add device”.  
   
2. Choose a device id. You will use this ID in the future.  
   Select RS256\_X509 authentication.  
   Fill in the public key you created and copied before.  
   Click “Add” to add your device.  
   
3. You should now have something like this:  
   

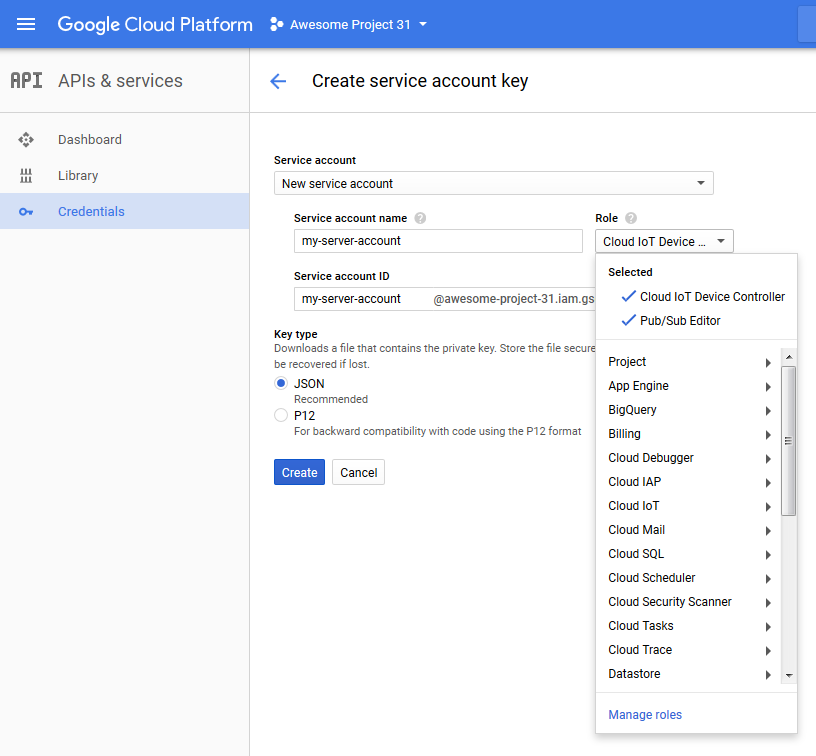
## 

## 

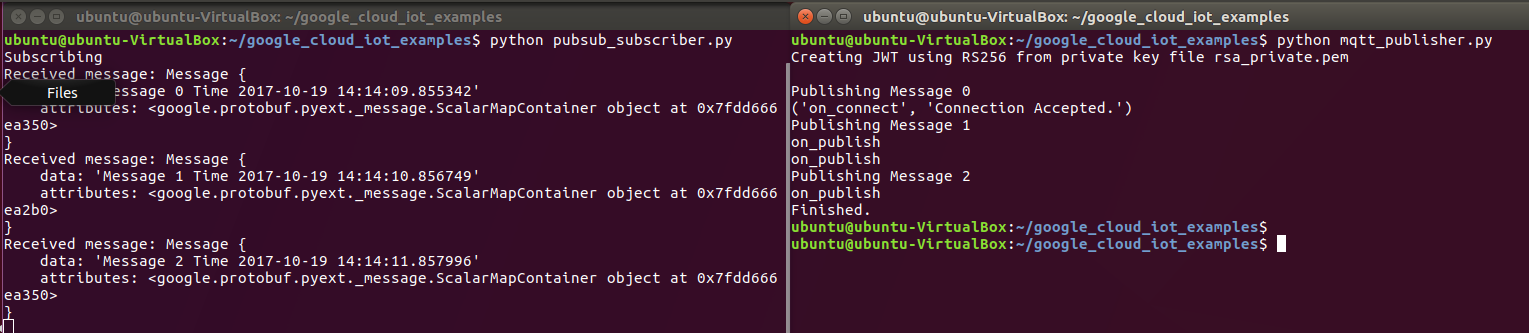
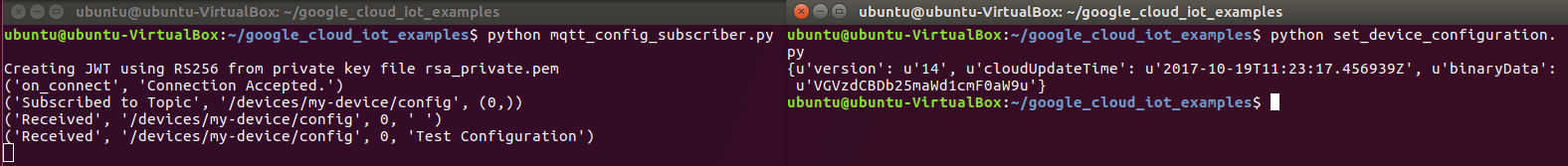
Creating server credentials for server-side services

1. Go to <https://console.cloud.google.com/cloudpubsub/topicList> and click “Enable API”.  
   
2. Go to <https://console.cloud.google.com/apis/credentials/serviceaccountkey>  
   Fill in an account name.  
   Under Role, add two roles:
   1. Cloud IoT Device Controller (under Cloud IoT) - this will allow the service to change device configuration.
   2. Pub/Sub Editor (under Pub/Sub) - this will allow the service to read, write and subscribe to Pub/Sub topics.

Click “Create”.

Save the file generated as “**service\_account.json**” on your computer.  


Running the sample code

1. On a linux machine type in the following commands  
   1. sudo apt update
   2. sudo apt upgrade
   3. sudo apt install git python-pip python-dev build-essential libffi-dev libssl-dev
   4. git clone https://github.com/omershv/google\_cloud\_iot\_examples.git
   5. cd google\_cloud\_iot\_examples
   6. sudo pip install -r requirements.txt
   7. sudo pip install --upgrade google-auth-oauthlib
   8. sudo date -s "12 DEC 2017 13:19:00"
   9. wget <https://pki.google.com/roots.pem>
2. If all commands were successful you should have now python with all of the required packages, the example code and Google’s certificates.
3. Copy **service\_account.json** and **rsa\_private.pem** from the previous sections to the *google\_cloud\_iot\_examples* directory
4. Notice that the credential files you just copied should never be in the same machine in production setting as **rsa\_private.pem** is the private key for the actual IoT device and **service\_account.json** contains the credentials of the server.
5. Edit the files *mqtt\_publisher.py*, *mqtt\_config\_subscriber.py*, *pubsub\_subscriber.py*, *set\_device\_configuration.py* and change the variables in their beginning to the ones that match your project (the lines you need to change have comments that start with “Enter”).
6. Open two terminal windows and run in them (in parallel):  
   1. python pubsub\_subscriber.py
   2. python mqtt\_publisher.py
7. You should see messages going from the MQTT publisher to the Pub/Sub subscriber.  
   
8. Open two terminal windows and run in them (in parallel):  
   1. python mqtt\_config\_subscriber.py
   2. python set\_device\_configuration.py
9. You should see the device configuration change.  
   
10. Congratulations! You now have all of the tools for secure IoT communication between a device and a server, you should now modify the example code to your needs and embed it in your project.

sudo pip install --upgrade google-auth-oauthlib