Google Cloud Hosting

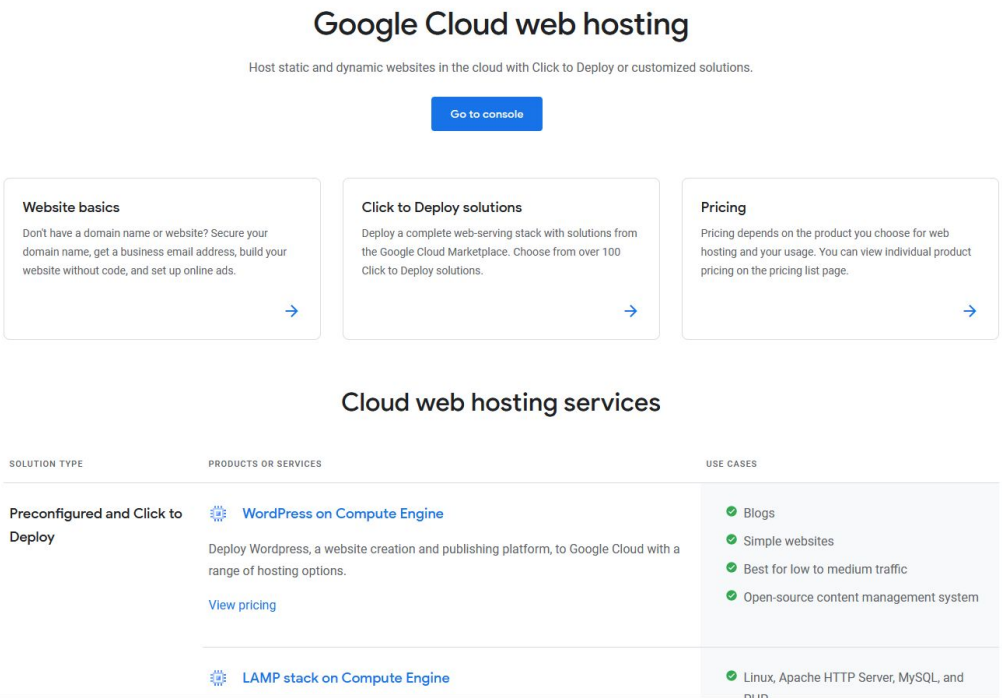
Web Development
Foundations

Setting up Google Cloud Hosting

Google Cloud Hosting

Google offers various website and web application hosting solutions via their Google Cloud web hosting platform.

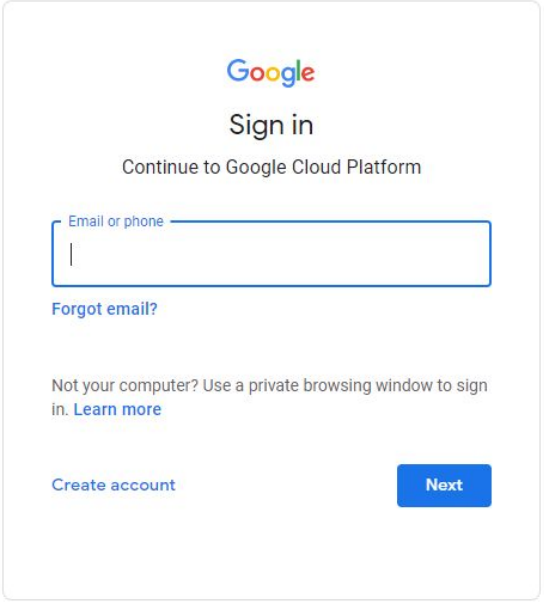
They have an extensive selection of options that are well documented both through Google's internal guides and many users' guides across the web.



Getting Started

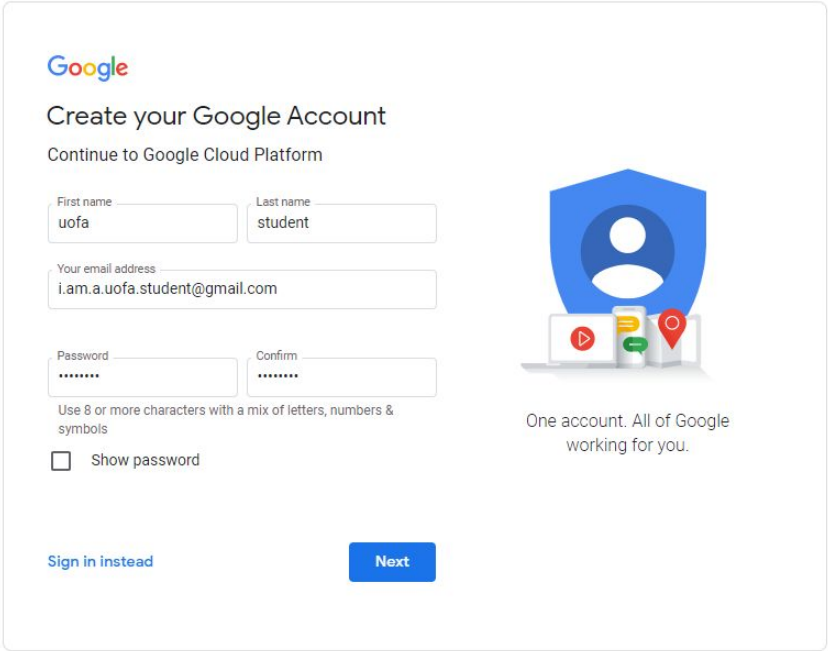
Firstly, you'll need a Google account. Start a Google Cloud web hosting trial by clicking here (the "Get started for free" button on the last slide.) It will ask you to sign in—or up—with a Google account if you aren't already signed in.

If you have your CCID and an associated Google account via the University of Alberta, you should have no problem using that account while following along in this exercise.



Create your Google Account

For personal use, or if you do not yet have access to your Google Workspace features via the University of Alberta, you may make a new Google Account as per normal.



Try Google Cloud Platform for free

Step 1 of 2

Uofa Student
i.am.a.university.of.alberta.student@gmail.com

SWITCH ACCOUNT

Country

Canada

Terms of Service

I agree to the Google Cloud Platform Terms of Service, and the terms of service of any applicable services and APIs. I have also read and agree to the Google Cloud Platform Free Trial Terms of Service.

Required to continue

Email updates

I would like to receive periodic emails on news, product updates and special offers from Google Cloud and Google Cloud Partners.

CONTINUE

Access to all Cloud Platform Products

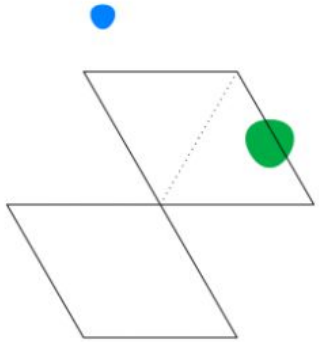
Get everything you need to build and run your apps, websites and services, including Firebase and the Google Maps API.

$300 credit for free

Put Google Cloud to work with $300 in credit to spend over the next 90 days.

No autocharge after free trial ends

We ask you for your credit card to make sure you are not a robot. You won't be charged unless you manually upgrade to a paid account.



Try Google Cloud Platform for free

Step 2 of 2

Your payment information helps us reduce fraud and abuse. **You won't be charged unless you turn on automatic billing.**

Customer info

Account type

Business

Name and address

Business name

Name

Uofa Student

Address line 1

Address line 2

City

Province

Postal code

Primary contact

Uofa Student
+1 7806165254
i.am.a.university.of.alberta.student@gmail.com

How you pay

Monthly automatic payments

You pay for this service on a regular monthly basis, via an automatic charge when your payment is due.

Payment method

Add credit or debit card

Card details

Credit or debit card address is same as above

START MY FREE TRIAL

Privacy policy | FAQs

Access to all Cloud Platform Products

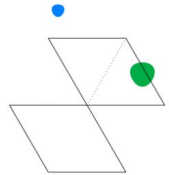
Get everything you need to build and run your apps, websites and services, including Firebase and the Google Maps API.

$300 credit for free

Put Google Cloud to work with $300 in credit to spend over the next 90 days.

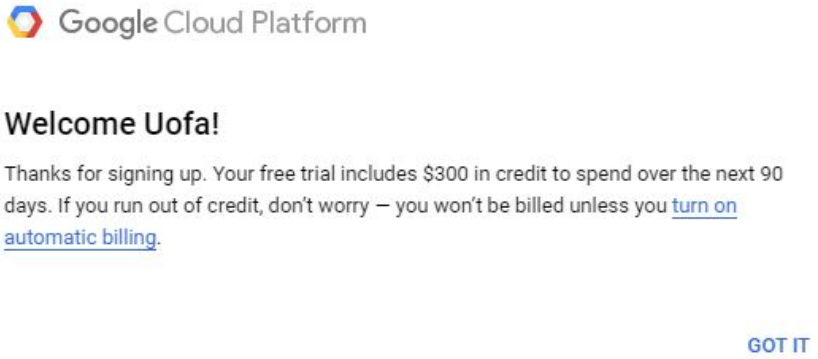
No autocharge after free trial ends

We ask you for your credit card to make sure you are not a robot. You won't be charged unless you manually upgrade to a paid account.



Pay Attention to Billing and Policies

Pay attention to their billing policies and settings, terms of service, privacy policy, and any other stipulations/notices, as **we are not responsible if you set up a payment plan, choose non-free options, or for any data sharing or consequences that arise from your Google account or the Google Cloud platform**.

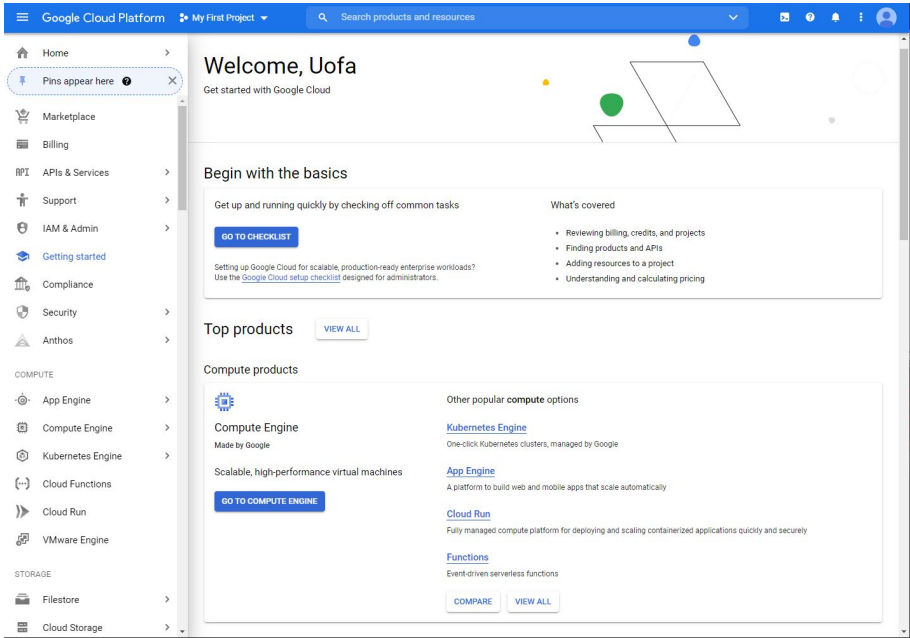


The Dashboard

You'll be greeted with the Google Cloud Platform dashboard!

From here, you have a wide array of options to choose from… we'll be focusing on a few that will help get you started with a basic web server.

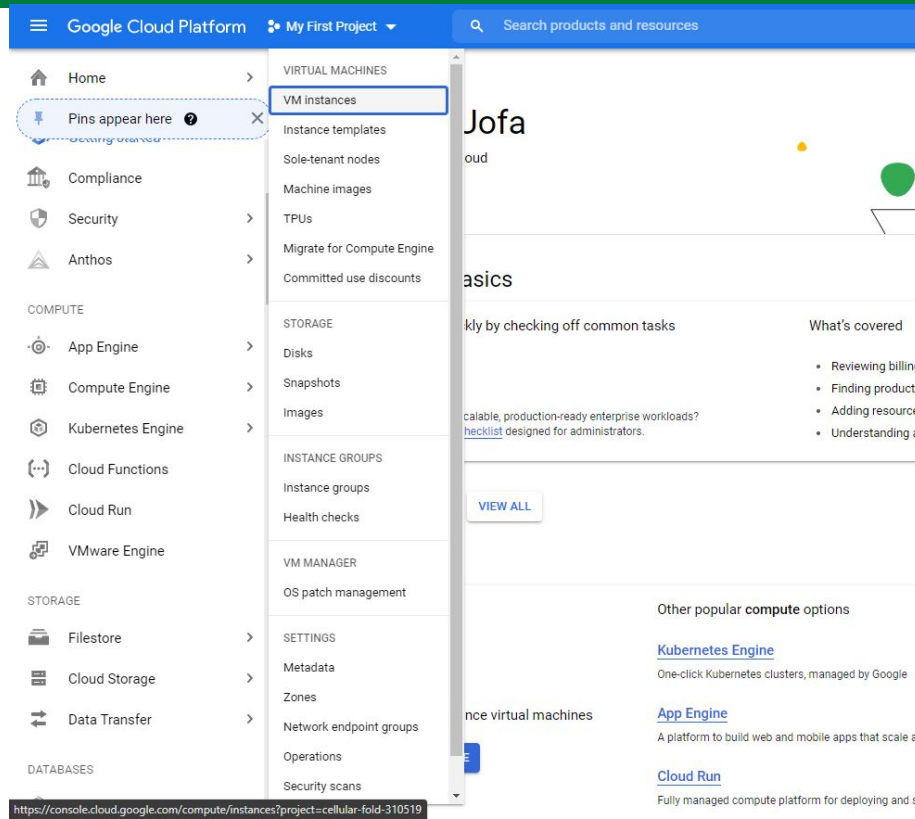
At the time of this writing, it is free to carry forth with these steps.



Compute Engine

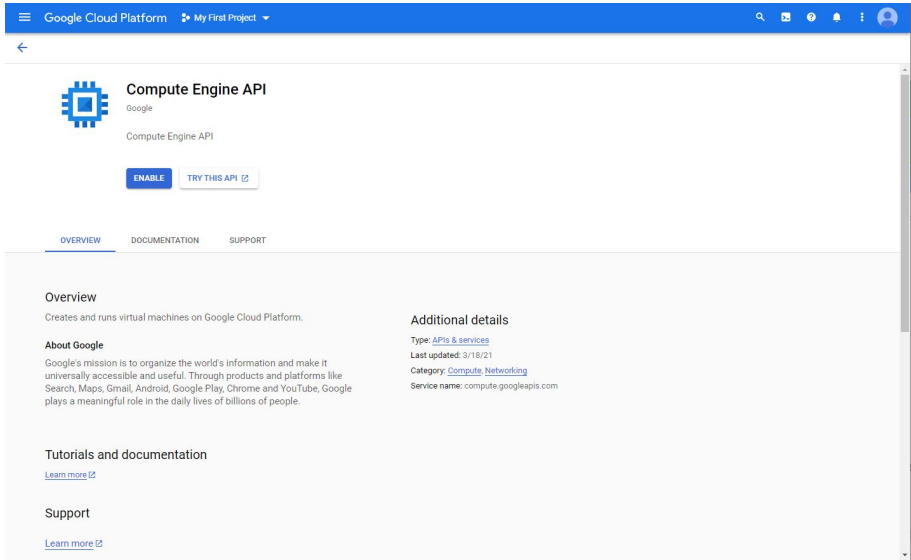
Firstly, we will need a virtual machine space for our server programs and processes to run in.

Navigate to “Compute Engine” → “VM Instances.”



Enable Compute Engine

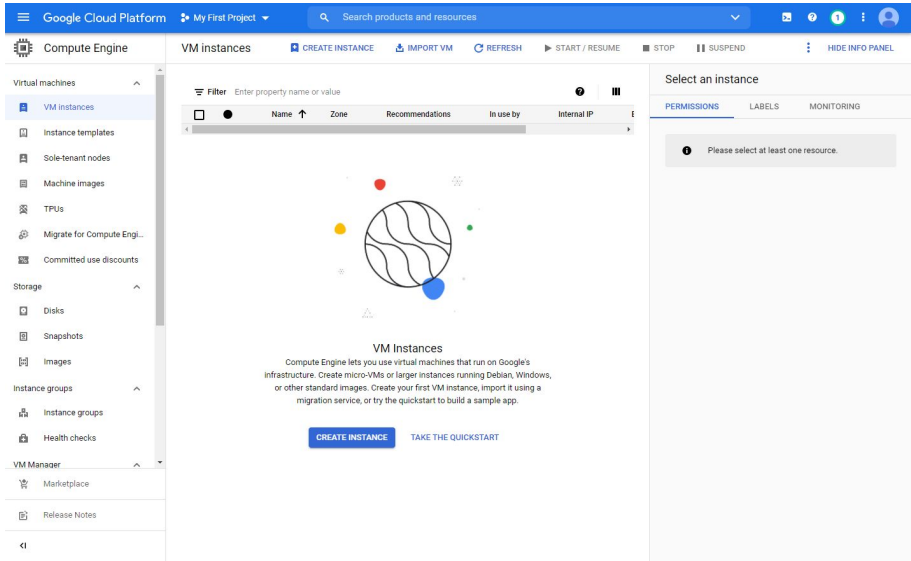
Enable the “Compute Engine API” feature.



Create Instance

Click the “Create Instance” button. This will open a page and dialogue regarding a virtual machine configuration.

There are many options available to you, but again we will be focused on a light and free hosting option today.



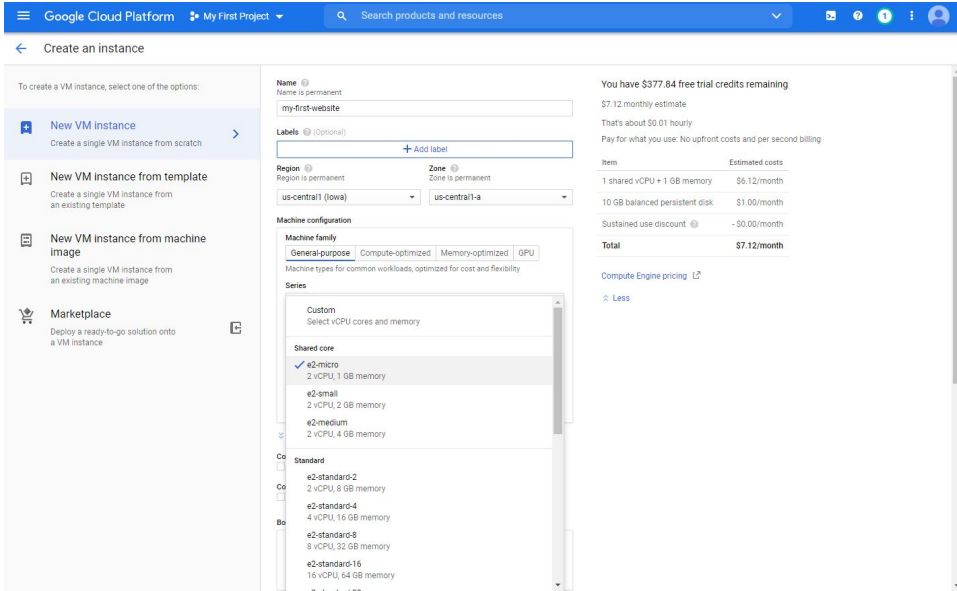
Beginning to Configure your Instance

Fill in a name for your virtual machine instance.

Select the E2 series.

For Machine type, select the micro option under “Shared core.”

These options should be within the free tier.

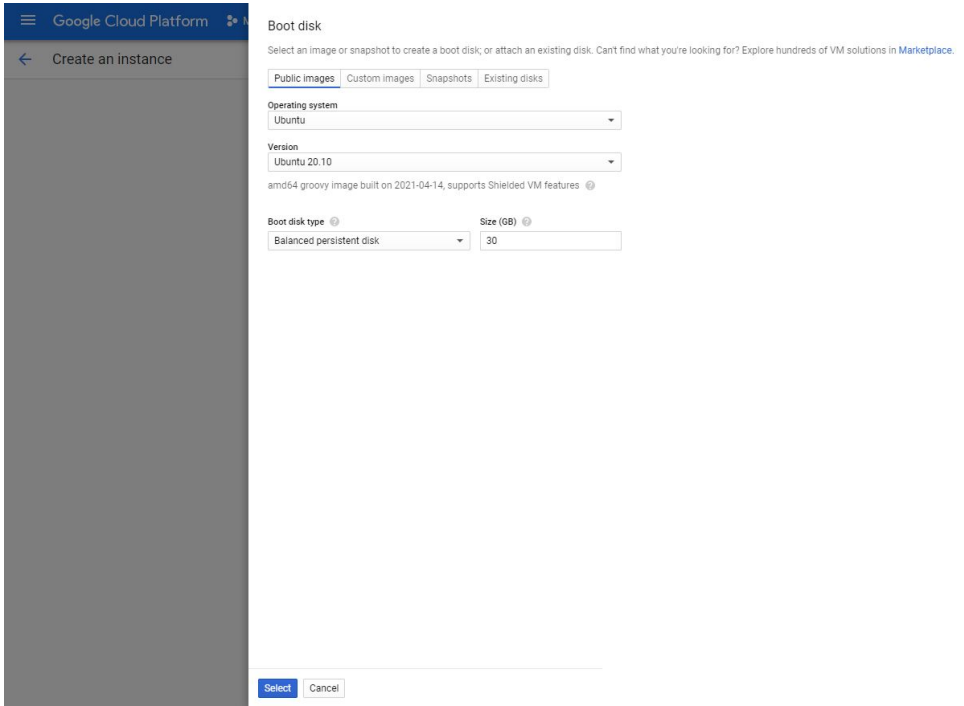


Select a Boot Disk

Click the “Change” button under the “Boot Disk” heading.

For our example, we’ll be using Ubuntu 20.10 as our operating system. Keep in mind there are other popular options for web hosting like CentOS that may be good to look into.

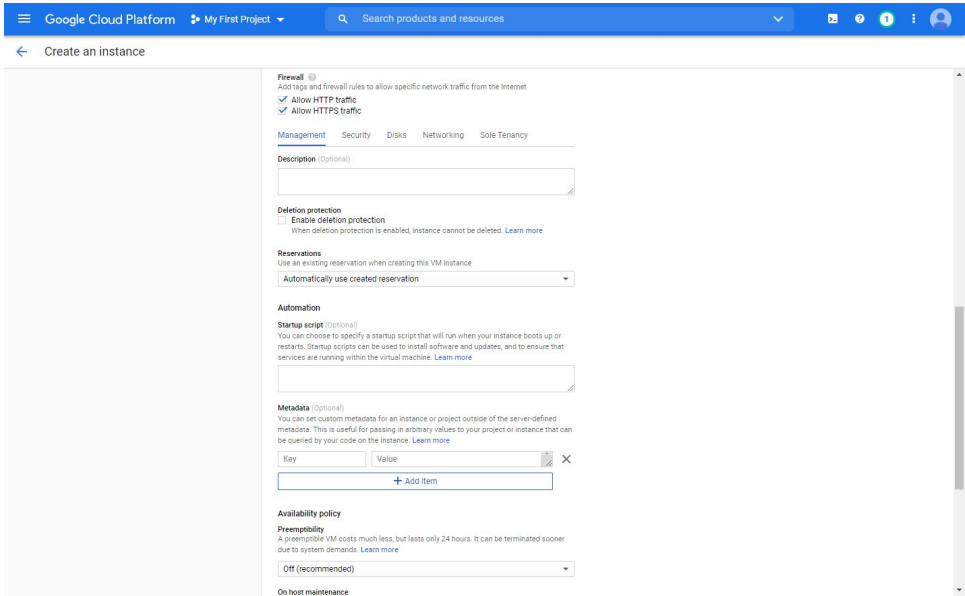
“Boot disk type” can be set up to 30GB, and remain free (the default may be ten, so it is recommended you increase it if this will be the only instance on your account.



Other Settings

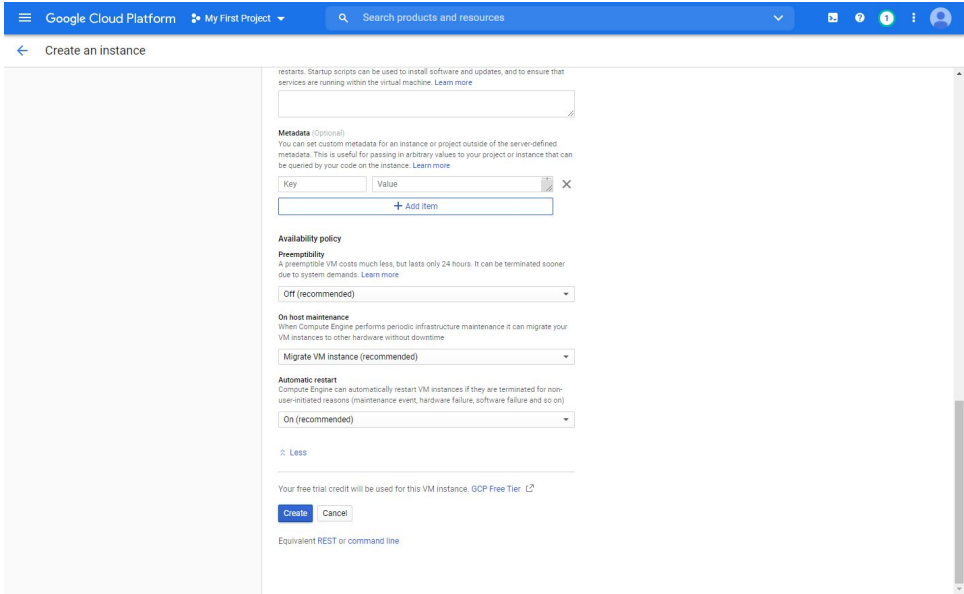
If you'd like the public to be able to view files from the server, and have easier access to it outside of your Google control panels, check the HTTP and HTTPS traffic boxes.

Make note of the Startup script box under the "Automation" heading. As you become a more advanced user these can be used to immediately run certain commands when your virtual machine boots.



Creating the Instance

Before clicking “Create,” read the GCP Free Tier guidelines to ensure you’re following current policy and restrictions.

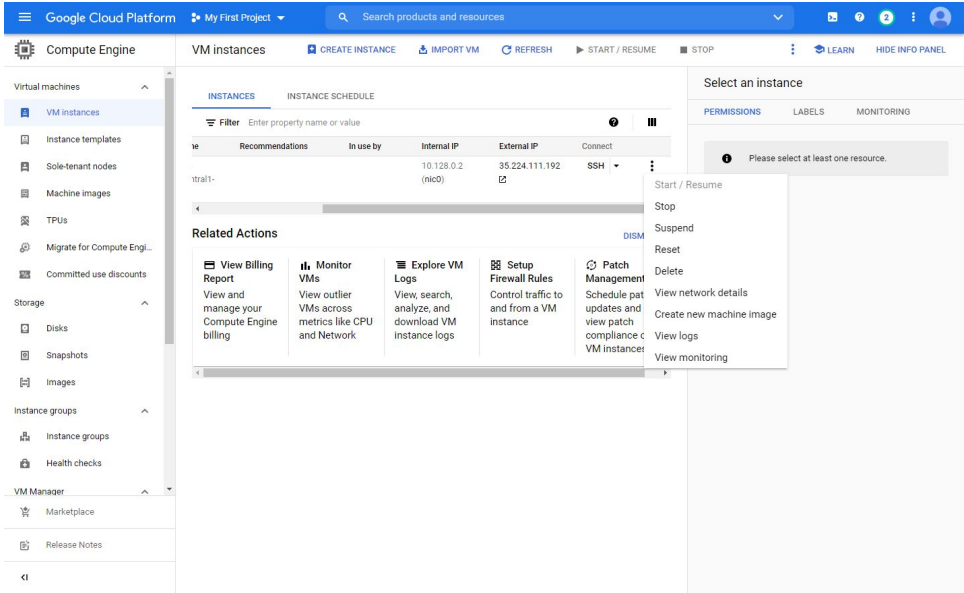


Getting Familiar with the Dashboard

Once the instance is created, you'll be directed to the "VM instances" page. Here you'll see a list of the instances you have under your account. If you've been following along, the one we just made should appear there.

You can scroll from left to right to see the entirety of the table—note the options available on the right-end of the table.

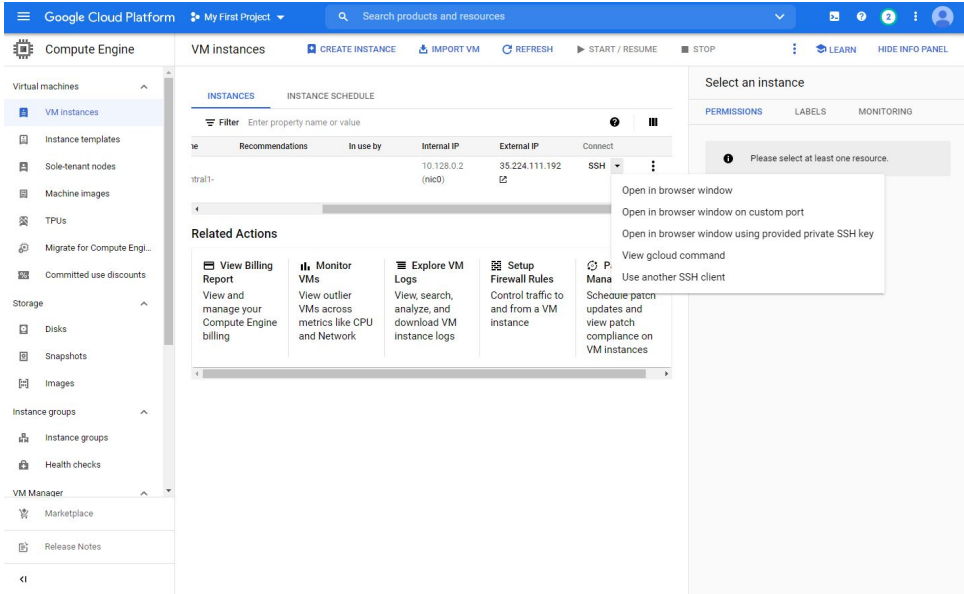
Note the "External IP" address, as this is what you can use in your web browser address bar to access the website once we set it up.



Accessing the Google Cloud Console

To access the console in your server to run commands in it, you can use the SSH button and/or dropdown in the instance table.

If you click “Open in browser window” you’ll be greeted with the following.



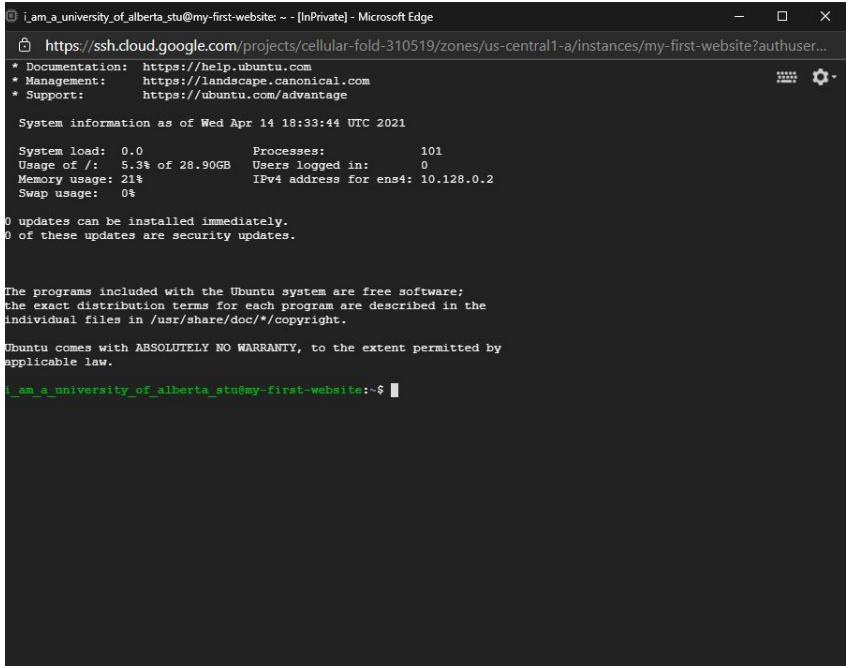
Google Cloud Console

Here in the Google Cloud Console, you have access to your server's Bash command-line interface.

You can run commands like you would in your own Bash terminal, but they will run on the server you are connected to.

Give it a try with a familiar command like "ls /" to see what's in the root directory!

```
i_am_a_university_of_alberta_stu@my-first-website:~$ ls /
bin   dev  home  lib32  libx32      media  opt   root  sbin  srv  tmp  var
boot  etc  lib   lib64  lost+found  mnt    proc  run   snap  sys  usr
```



Server-Side FTP Setup

FTP on Your Server

You can install an FTP (File Transfer Protocol) server software on your server to make it easier to transfer files to and from the server from your home or office computer.

To see if you have this installed on our setup, run:

```
sudo apt list --installed
```

See if “vsftpd” appears in the list.

If not, you can proceed with the installation:

```
sudo apt install vsftpd
```

Once installed, copy the default configuration file so that we can begin configuring as we need to.

```
sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default
```

This leaves the original configuration in case we ever need to refer to it or copy it again.

To run the service, run the following:

```
sudo systemctl start vsftpd
```

```
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo apt install vsftpd
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libnuma1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ssl-cert
Suggested packages:
  openssl-blacklist
The following NEW packages will be installed:
  ssl-cert vsftpd
0 upgraded, 2 newly installed, 0 to remove and 3 not upgraded.
Need to get 132 kB of archives.
After this operation, 402 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-central1.gce.archive.ubuntu.com/ubuntu groovy/main amd64 ssl-cert all 1.0.39 [17.0 kB]
Get:2 http://us-central1.gce.archive.ubuntu.com/ubuntu groovy/main amd64 vsftpd amd64 3.0.3-12 [115 kB]
Fetched 132 kB in 0s (875 kB/s)
Preconfiguring packages ...
Selecting previously unselected package ssl-cert.
(Reading database ... 95881 files and directories currently installed.)
Preparing to unpack .../ssl-cert_1.0.39_all.deb ...
Unpacking ssl-cert (1.0.39) ...
Selecting previously unselected package vsftpd.
Preparing to unpack .../vsftpd_3.0.3-12_amd64.deb ...
Unpacking vsftpd (3.0.3-12) ...
Setting up ssl-cert (1.0.39) ...
Setting up vsftpd (3.0.3-12) ...
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /lib/systemd/system/vsftpd.service.
Processing triggers for man-db (2.9.3-2) ...
Processing triggers for systemd (246.6-1ubuntu1.3) ...
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo systemctl start vsftpd
```

Enable FTP Writing

One easy way to access and edit the file is using “nano” which is pre-installed on many operating systems:

```
sudo nano /etc/vsftpd.conf
```

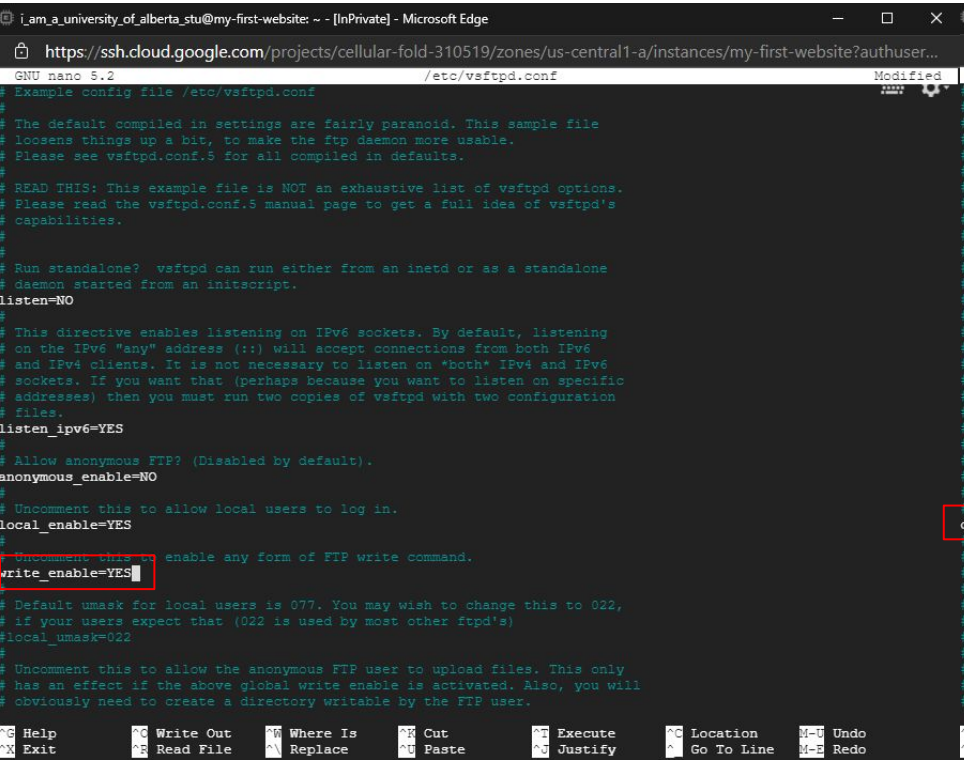
Open the configuration file, and uncomment the “write_enable” line so that it looks like the following:

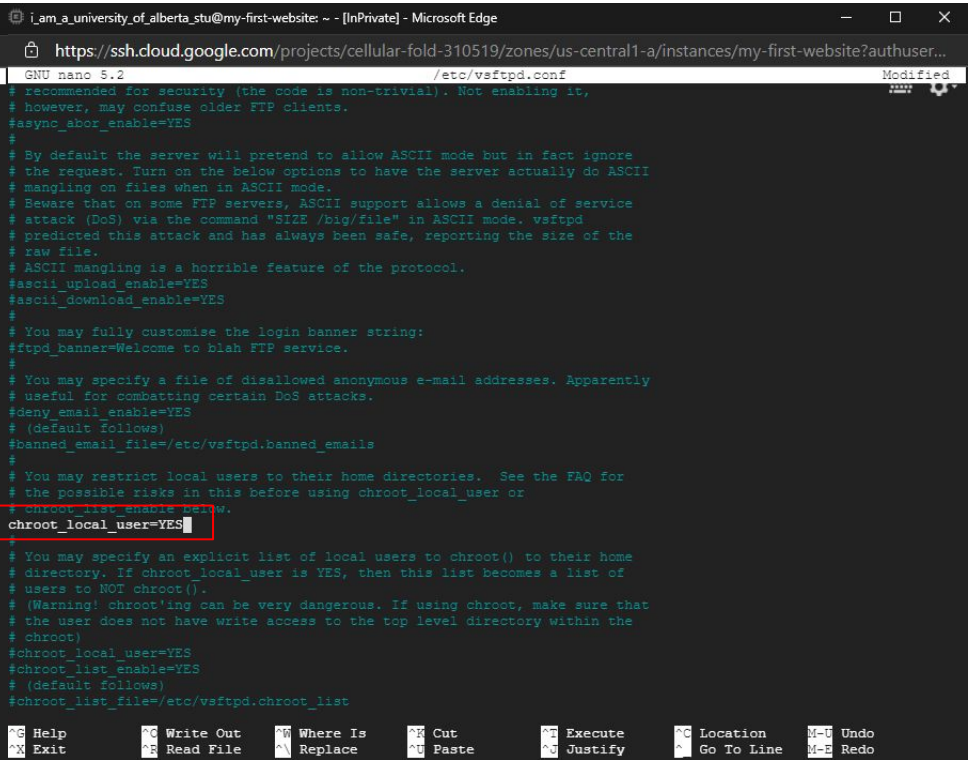
```
write_enable=YES
```

To restrict access to local users, you can also adjust the “chroot_local_user” line like so:

```
chroot_local_user=YES
```

```
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo nano /etc/vsftpd.conf
```





Enable a User Whitelist

We'll need to update one more area. See the following updates to "chroot_list_enable", "chroot_list_file", and "local_root":

chroot_list_enable=YES

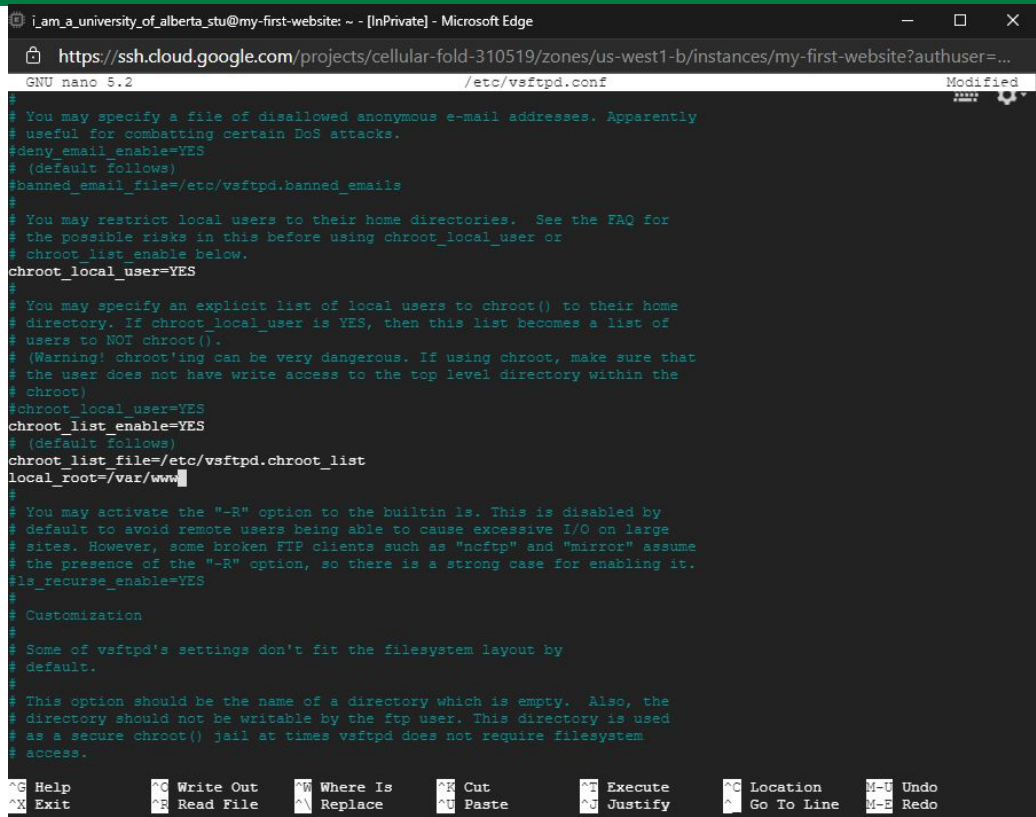
chroot_list_file=/etc/vsftpd.chroot_list

local_root=/var/www

Create and open a new file for the list of users allowed to :

sudo touch /etc/vsftpd.chroot_list

sudo nano /etc/vsftpd.chroot_list

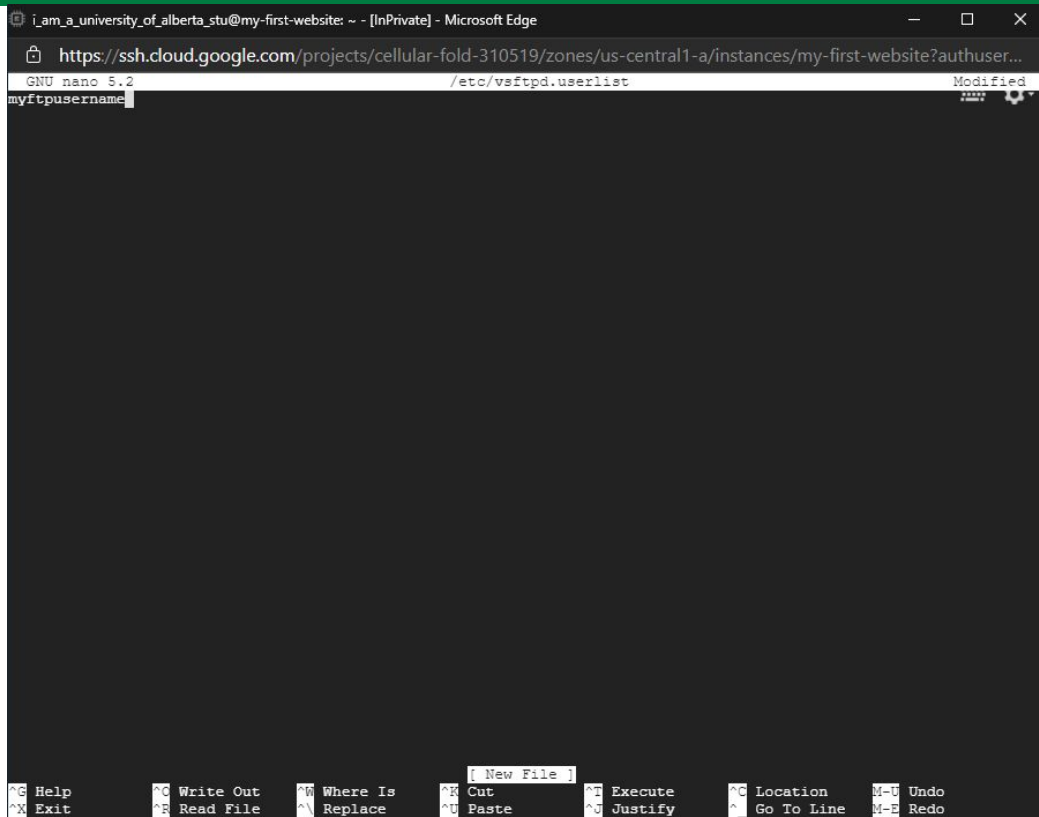


Add names to the List

In the file:

/etc/vsftpd.userlist

You can add a list of any users that should be granted access to your FTP-enabled folders and files.



Add an FTP User to your Server

To add a user by name, replace “myftpusername” with a name of your choice in the following command:

sudo useradd -m myftpusername

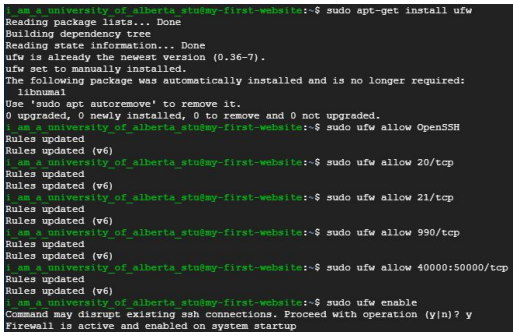
You can add a password for use of this account via the following, ensure you respond to the prompt with a **secure** password of your choice:

sudo passwd myftpusername

```
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo useradd -m myftpusername
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo passwd myftpusername
New password:
Retype new password:
passwd: password updated successfully
```

Your FTP setup may not be accessible if the Ubuntu firewall is blocking related FTP ports. To allow access via one of these ports, try using ufw (Uncomplicated Firewall):

sudo ufw allow OpenSSH
sudo ufw allow 20/tcp
sudo ufw allow 21/tcp
sudo ufw allow 990/tcp
sudo ufw allow 40000:50000/tcp



Add Permissions to the User

Now that you have a user, we'll want to give it permissions to use and modify files and folders in our website:

sudo chown myftpusername /var/www/html

```
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo chown myftpusername /var/www/html
```

Restarting the Service

Once you have configured the FTP service and added a user, it is a good idea to restart the service. In doing this you can be sure that any changes will be noticed by the system and/or program.

Use the following command:

sudo systemctl restart vsftpd.service

```
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo systemctl restart vsftpd.service
```

Restarting the Server

To ensure all settings for everything take effect, it can be useful to restart the server operating system in its entirety. Try the following...

```
sudo reboot
```

After this command, you may have to wait a moment and reconnect to the VM when it has started again. Once restarted, the firewall we set up will be activated. Then try:

```
sudo systemctl start vsftpd
```

This will ensure once more that the FTP software is initialized and accepting connections.

Making FTP Secure

Create a Certificate

To encrypt traffic—that is to say, make it difficult or near-impossible to read your requests and web traffic when engaging with your server—we must have a certificate and enable SSL/TLS features on our server and in our FTP software.

So, first thing's first:

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
```

Here we are generating a key and file, using OpenSSL, that will be used for encryption. Note that the number of days can be altered to your liking or requirements. 365 would be a 1 year period before expiration. For more details on the options available to you, visit the official documentation.

This command will ask you for some information that will be attached to the key(s) in question, fill it in as you see fit.

```
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyo
ut /etc/ssl/private/vsftpd.pem -out /etc/ssl/private/vsftpd.pem
Generating a RSA private key
.....................................................................................................
.....................................+++++
.....................................................................................................
.............................................+++++
writing new private key to '/etc/ssl/private/vsftpd.pem'
-----
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]:CA
State or Province Name (full name) [Some-State]:Alberta
Locality Name (eg, city) []:Edmonton
Organization Name (eg, company) [Internet Widgits Pty Ltd]:UofA Student
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:MFW
Email Address []:i.am.a.university.of.alberta.student@gmail.com
```

Update your VSFTPD Configuration

We will need to let our FTP server software know where the new certificate and key is! Open up the configuration file using Nano once more:

```
sudo nano /etc/vsftpd.conf
```

Update the “rsa_cert_file” and “rsa_private_key_file” values with the locations of the respective files:

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem
rsa_private_key_file=/etc/ssl/private/vsftpd.pem
```

Enable the SSL featureset in the software by uncommenting the “ssl_enable” line:

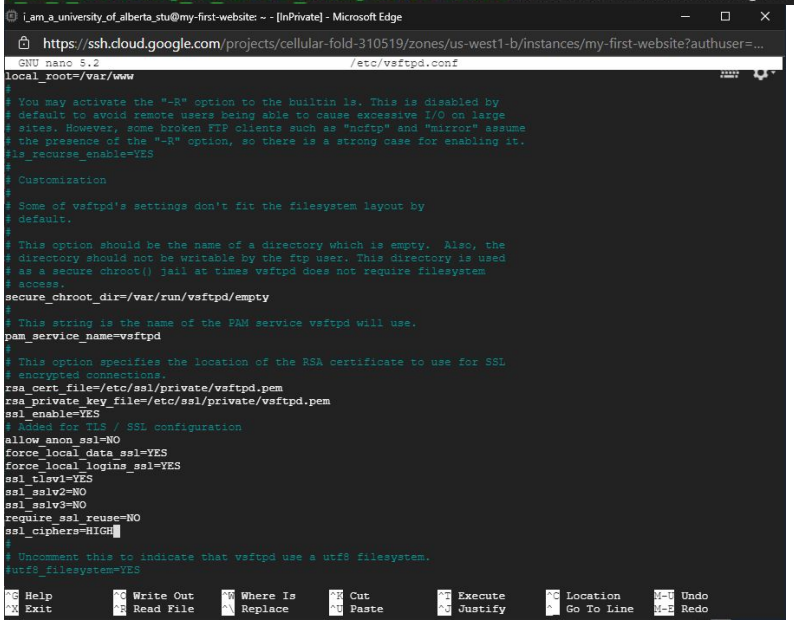
```
ssl_enable=YES
```

Add the following as well, to ensure SSL is enforced. This prevents anonymous users and will force you to connect securely—this way you know future server interactions will be difficult or near-impossible to interpret if your requests are intercepted.

```
allow_anon_ssl=NO
force_local_data_ssl=YES
force_local_logins_ssl=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
require_ssl_reuse=NO
ssl_ciphers=HIGH
```

Restart the FTP Service

```
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo nano /etc/vsftpd.conf
```



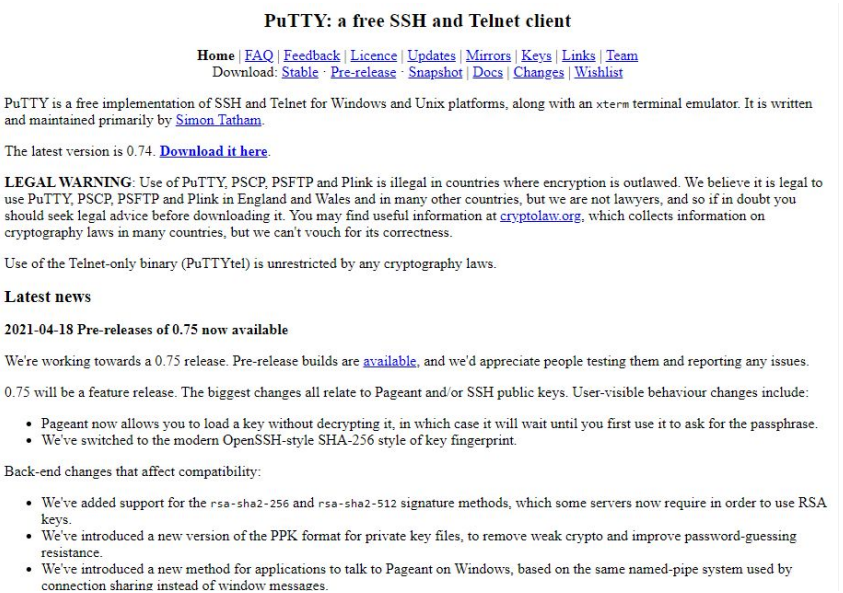
Once you've made the necessary changes and additions, restart the server FTP service:

sudo systemctl restart vsftpd.service

```
i_am_a_university_of_alberta_stu@my-first-website:~$ sudo systemctl restart vsftpd.service
```

Install PuTTY

Navigate to the PuTTY website and install the PuTTY program. This will come with tools for generating keys for SSH access to your server.



PuTTY: a free SSH and Telnet client

Home | FAQ | Feedback | Licence | Updates | Mirrors | Keys | Links | Team
Download: Stable · Pre-release · Snapshot | Docs | Changes | Wishlist

PuTTY is a free implementation of SSH and Telnet for Windows and Unix platforms, along with an `xterm` terminal emulator. It is written and maintained primarily by Simon Tatham.

The latest version is 0.74. **Download it here**.

LEGAL WARNING: Use of PuTTY, PSCP, PSFTP and Plink is illegal in countries where encryption is outlawed. We believe it is legal to use PuTTY, PSCP, PSFTP and Plink in England and Wales and in many other countries, but we are not lawyers, and so if in doubt you should seek legal advice before downloading it. You may find useful information at cryptolaw.org, which collects information on cryptography laws in many countries, but we can't vouch for its correctness.

Use of the Telnet-only binary (PuTTYtel) is unrestricted by any cryptography laws.

Latest news

2021-04-18 Pre-releases of 0.75 now available

We're working towards a 0.75 release. Pre-release builds are available, and we'd appreciate people testing them and reporting any issues.

0.75 will be a feature release. The biggest changes all relate to Pageant and/or SSH public keys. User-visible behaviour changes include:

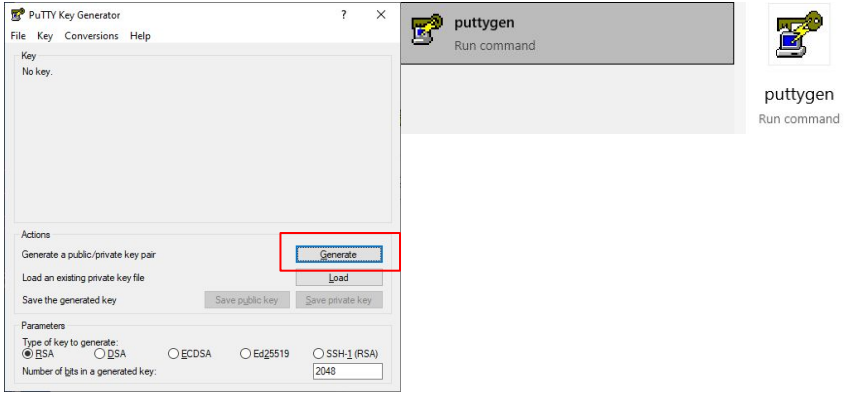
- Pageant now allows you to load a key without decrypting it, in which case it will wait until you first use it to ask for the passphrase.
- We've switched to the modern OpenSSH-style SHA-256 style of key fingerprint.

Back-end changes that affect compatibility:

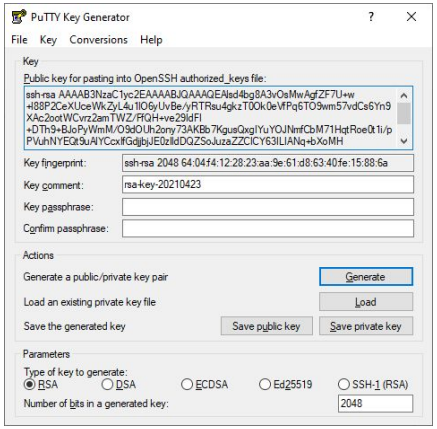
- We've added support for the `rsa-sha2-256` and `rsa-sha2-512` signature methods, which some servers now require in order to use RSA keys.
- We've introduced a new version of the PPK format for private key files, to remove weak crypto and improve password-guessing resistance.
- We've introduced a new method for applications to talk to Pageant on Windows, based on the same named-pipe system used by connection sharing instead of window messages.

Generate Keys

When you install PuTTY, it comes with a key generator. Open the “puttygen” program (it should show up in a Windows program search in your Start menu now.)



Click the “Generate” button. It may ask you to move your mouse cursor in the “Key” fieldset square to assist in randomizing the key data.



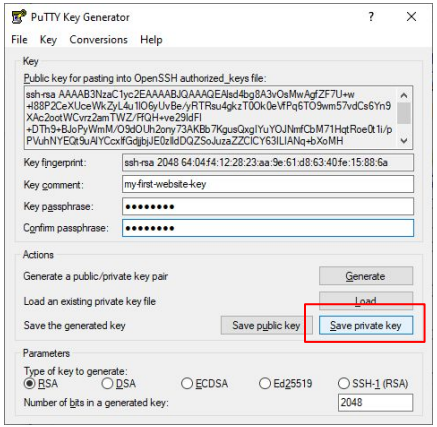
Save the Private Key

Feel free to modify the Key comment to help describe what the key is or what it is for.

Add a password to improve the security in using this key pair.

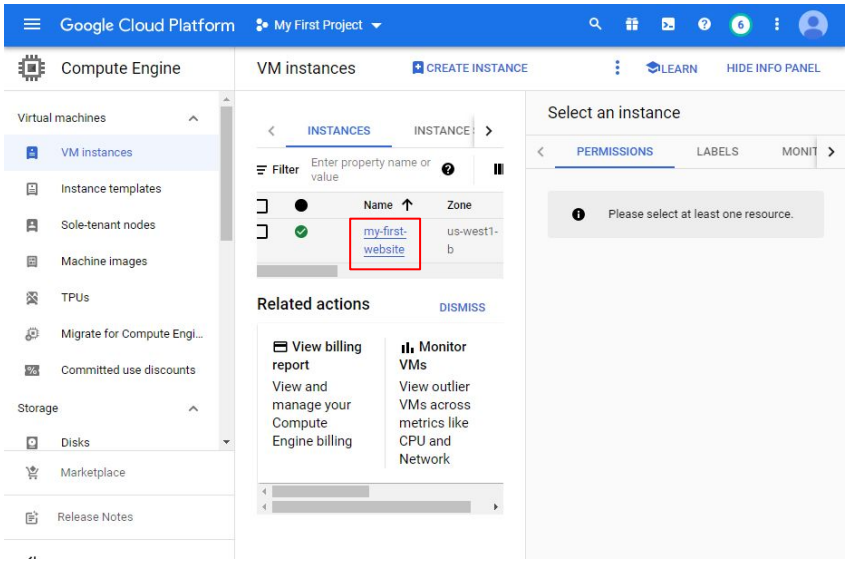
Save the private key to a folder on your computer. Make note of where you save this, as we will need it to sign in to your server later.

Leave the PuTTY Key Generation application open for the next steps.

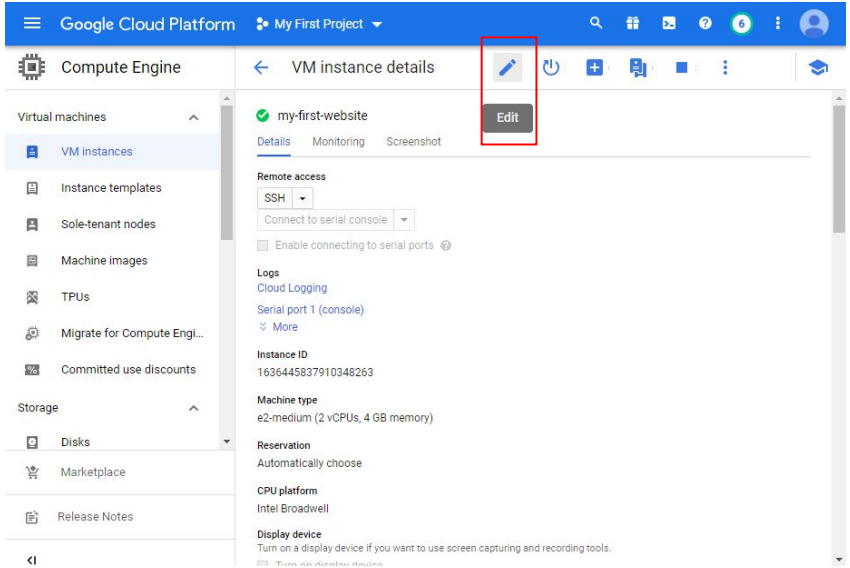


Navigate to your VM Instance Edit Screen

Navigate to your VM instance in your Google Cloud dashboard.

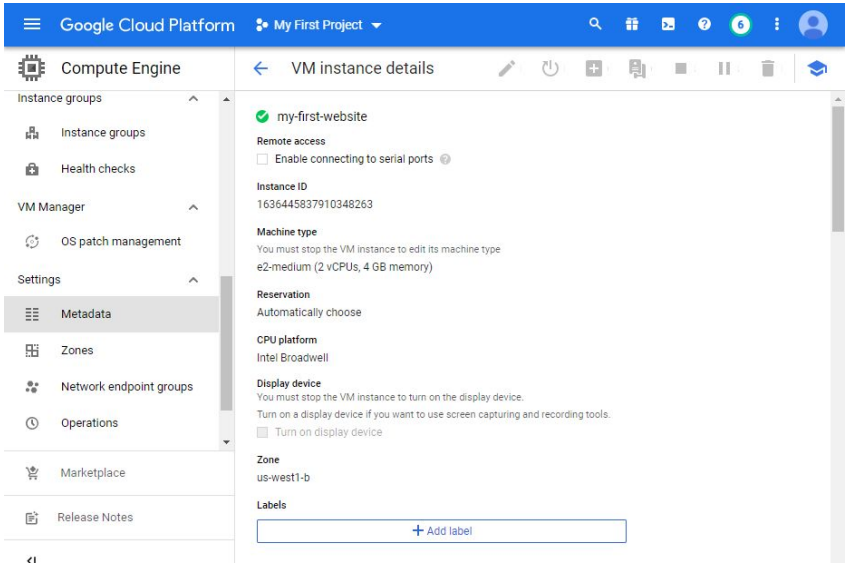


Click the “Edit” button, so that we may update our instance configuration.

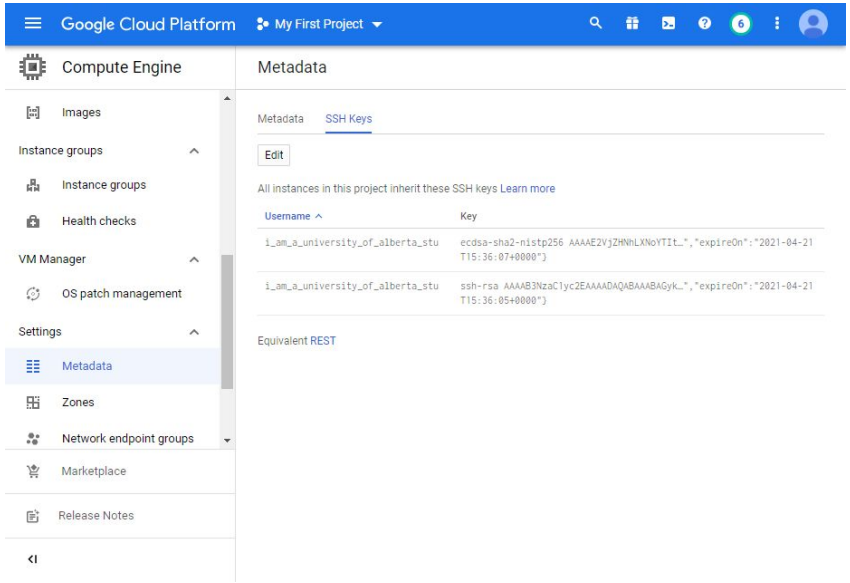


Configure Google Cloud with an SSH Key

Navigate to the “Metadata” settings.



Click the “SSH Keys” tab there.

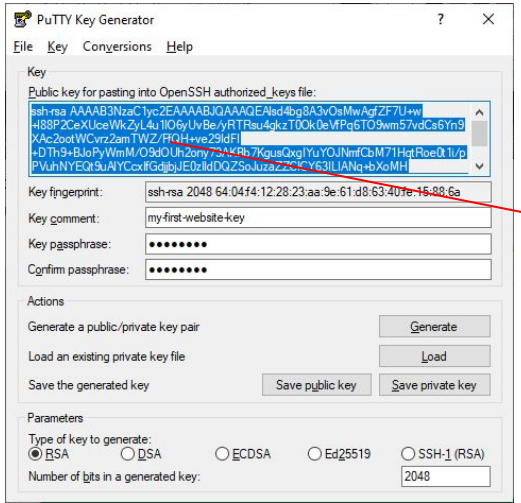


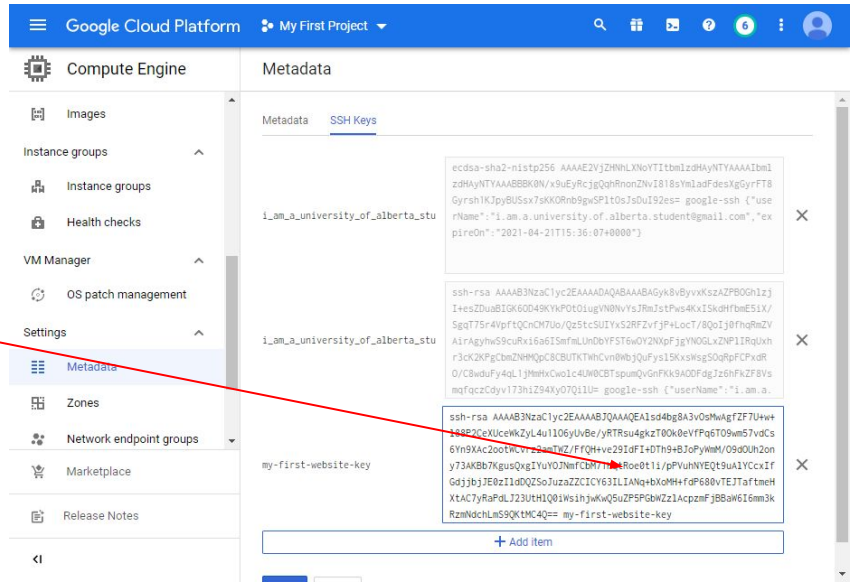
Add the Public Key to the Server

Click the “Add item” button to add a new textarea field that you can add information to.

Paste the public key data into the new field.

Don’t forget to click “Save” when you’re done, for these changes to be recorded!



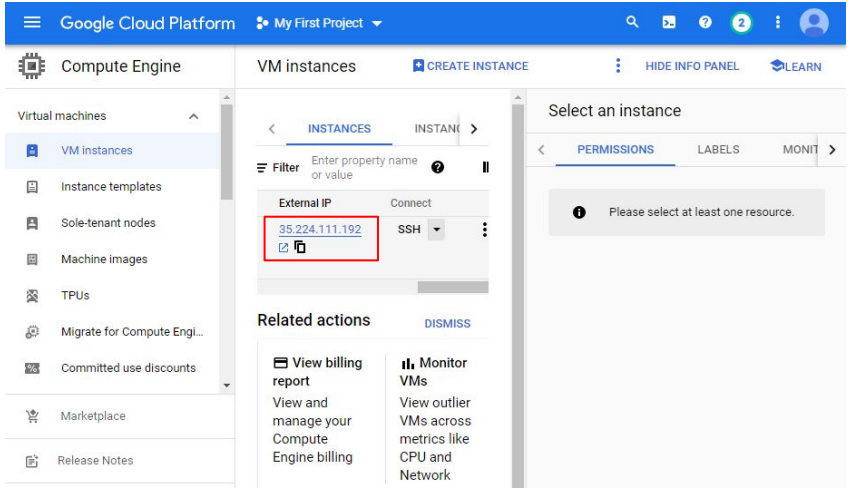


Connect to an FTP-Enabled Server using FileZilla

Gather Required Information

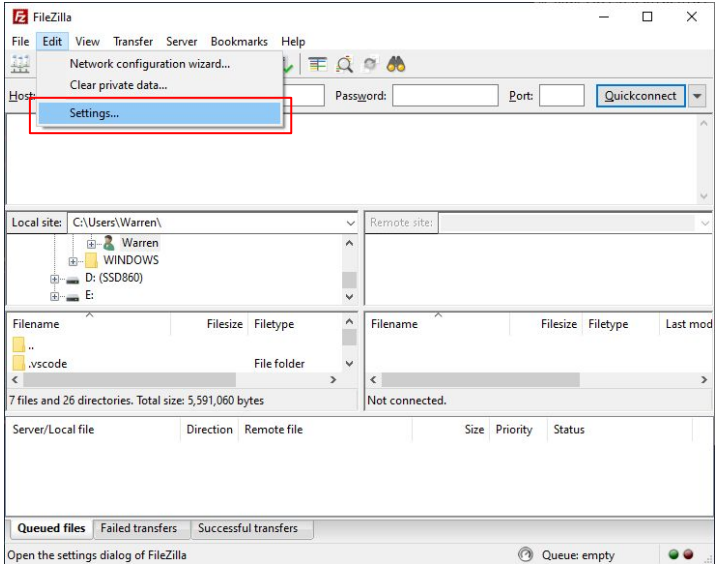
You'll need the address of the website server. Unless, or until, you have a domain name setup for the server—this will be the **external IP address**.

That aside, you'll need to make ensure you know the FTP account **username** and **password** that you set up earlier.

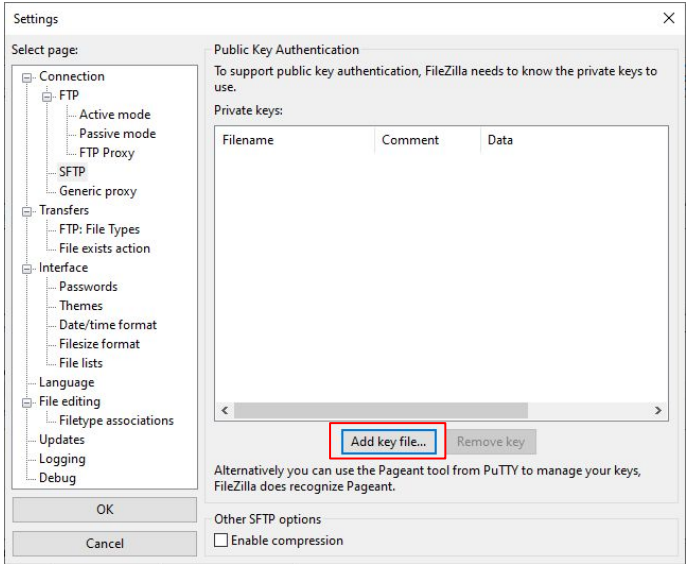


Add the Private Key to your Filezilla

In FileZilla, click the “Edit” dropdown and then the “Settings” option.



Navigate to “Connection→SFTP” page. Click the “Add key file…” button and find the key file(s) saved earlier. Click “OK” when done.



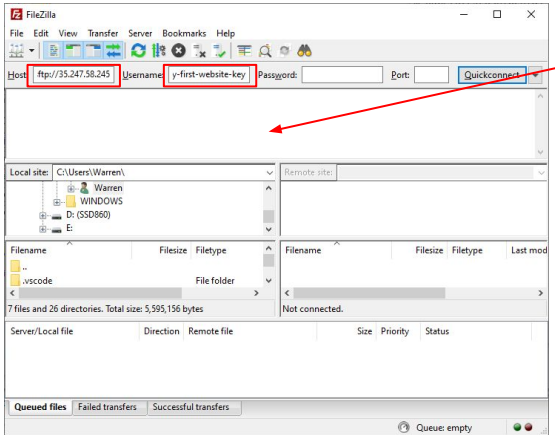
Enter the SSH Information

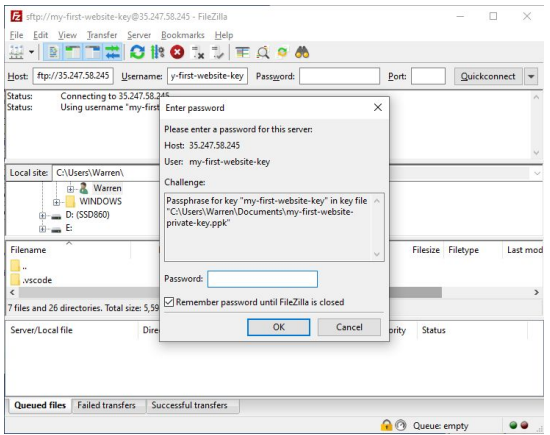
Enter “sftp://” into the “Host” field (sans quotations), followed by the external IP address of the server.

The Username should be the name of your key (as seen in your Google Cloud VM Instance “Metadata” SSH tab.)

It will ask for a password to connect to the server—use the password we associated with your key in this field.

Pay attention to the logging pane to see if the connection is successful.





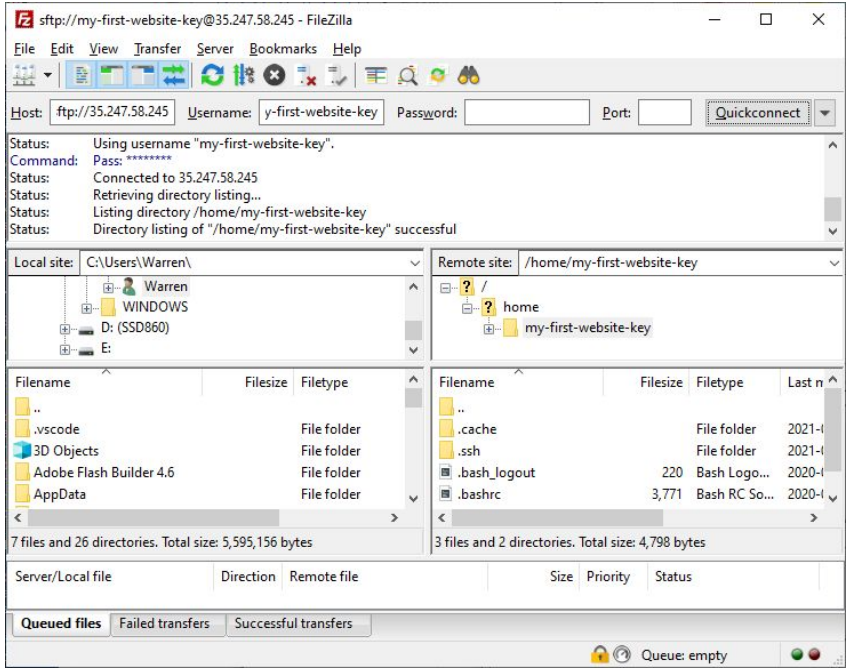
Practice Navigating in FileZilla

If all went well, you'll gain access to the filesystem on the server.

Note the **left pane contains the files and folders on your computer**, and the **pane on the right contains the files and folders on the server**.

You can click through the folders on the server, much like you would on your own PC using File Explorer.

***Beware:** Double-clicking or pressing the enter key on a file will transfer the file in the current pane to the open directory in the other pane, it will not open the file!



Setting up Apache (Preparing the Server for Web Traffic)

Installing Apache

First, install Apache:

```
sudo apt update
```

```
sudo apt install apache2
```

```
sudo systemctl start apache2
```

The default installation should come with a publicly available folder; its contents are served to web browser clients making requests to this server.

The public folder is where you can place HTML, CSS, JavaScript, and multimedia files for consumption by users via the internet.

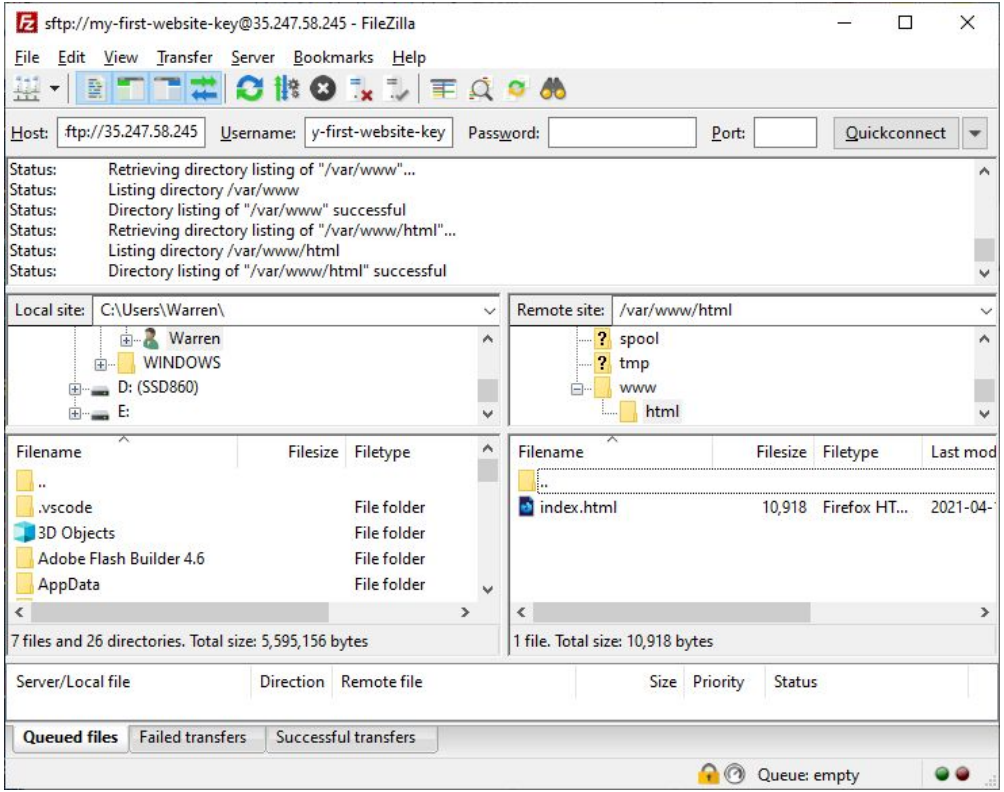
Typically, the directory in question can be found in…

```
/var/www/html
```

A simple test you can place in here might be a file titled “index.html” with the following contents:

```
<!DOCTYPE html>
<html>
  <head><title>Hello, World!</title></head>
  <body><h1>Hello, World!</h1></body>
</html>
```

If you navigate to the /var/www/html folder in FileZilla, you can drag files into the server pane to upload them.Replace the default “index.html” file.


sftp://my-first-website-key@35.247.58.245 - FileZilla
File Edit View Transfer Server Bookmarks Help
Host: ftp://35.247.58.245 Username: y-first-website-key Password: Port: Quickconnect
Status: Retrieving directory listing of "/var/www"...
Status: Listing directory /var/www
Status: Directory listing of "/var/www" successful
Status: Retrieving directory listing of "/var/www/html"...
Status: Listing directory /var/www/html
Status: Directory listing of "/var/www/html" successful
Local site: C:\Users\Warren\
Warren
WINDOWS
D: (SSD860)
E:
Remote site: /var/www/html
spool
tmp
www
html
Filename Filesize Filetype
..
.vscode File folder
3D Objects File folder
Adobe Flash Builder 4.6 File folder
AppData File folder
7 files and 26 directories. Total size: 5,595,156 bytes
Filename Filesize Filetype Last mod
..
index.html 10,918 Firefox HT... 2021-04-
1 file. Total size: 10,918 bytes
Server/Local file Direction Remote file Size Priority Status
Queued files Failed transfers Successful transfers
Queue: empty

Helpful Tutorials and Documentation

Ensure you familiarise yourself with the documentation for the tools offered by Google (or any other host you try out) as you'll likely need to refer to these if you want to try new things, forget steps you've carried out before, or want to explore what the product is capable of.

Some articles good to keep on hand to get started with include...

- All pricing
- Connecting using third-party tools
- gcloud compute (command-line tool)
- Using IAP for TCP forwarding
- Google Cloud Free Program
- Free Tier usage limits
- Google Cloud Pricing Calculator
- Get started with Google Cloud
- Install and Setup FTP on Ubuntu (Hostinger)
- Install Apache on Ubuntu

Recommended Readings

If you'd like a more in-depth look at hosting and configuration:

- Geewax, J. J. (September 2018). Google Cloud Platform in Action. Manning Publications.