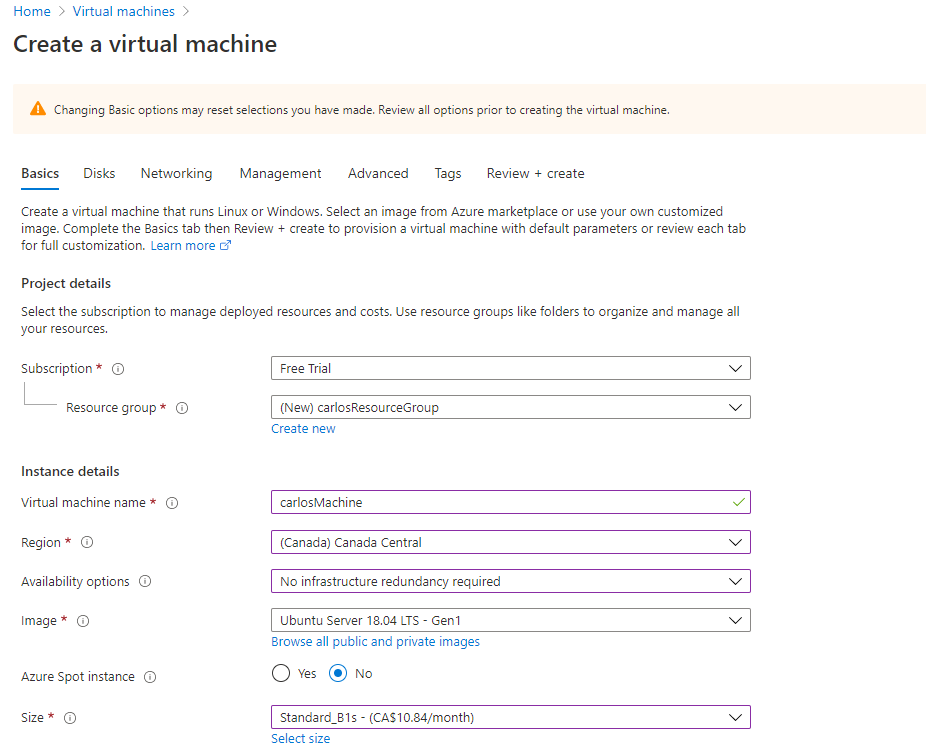
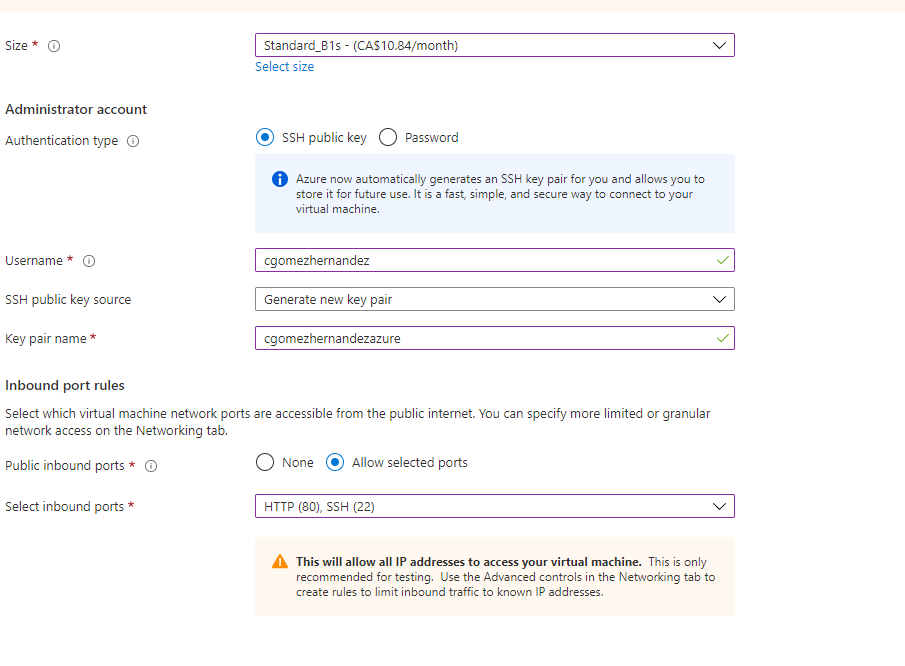
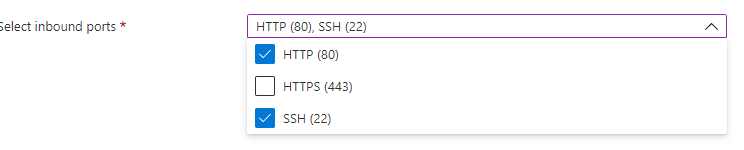
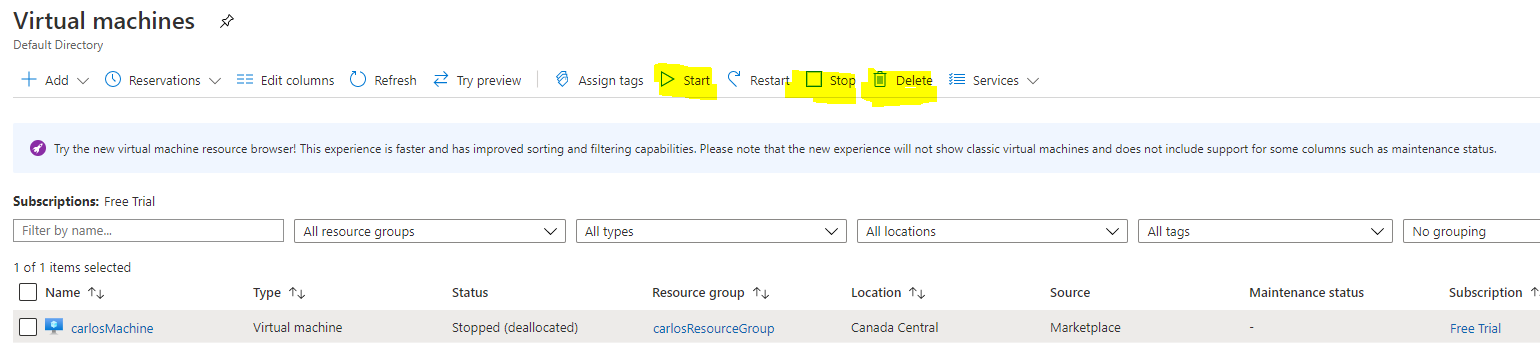
**CREATE A VIRTUAL MACHINE CONFIGURATIONS**

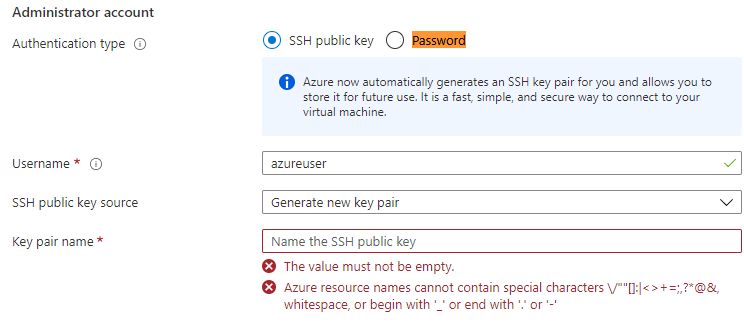






The rest of the tabs we left as Default

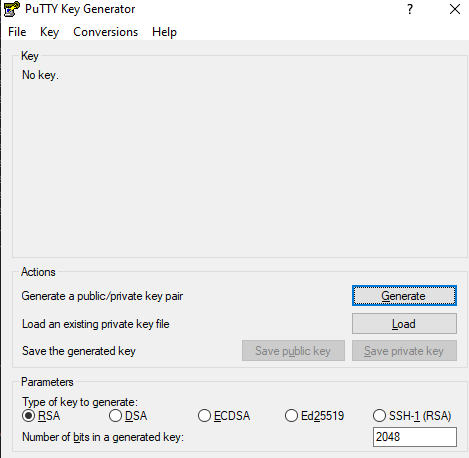
  
  
You are paying for:

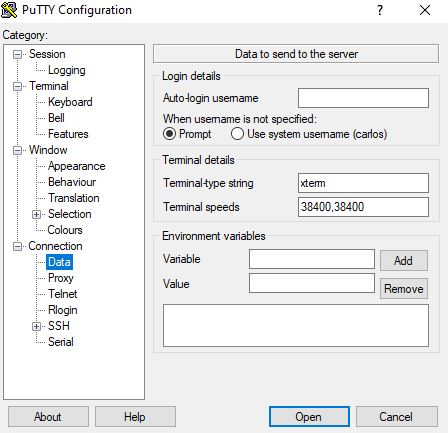
- The virtual Machine Running  
- The SSD containing your data  
  
If you want Azure to stop charging you for SSD you need to DELETE the machine.  
If you want Azure to stop charging you for runtime you need to stop the machine.  
  
Now that we have configured the machine we need to :

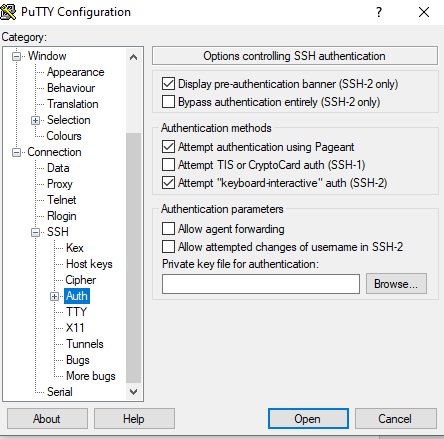
We created an account using an SSH public key and Azure gave us a key file.  

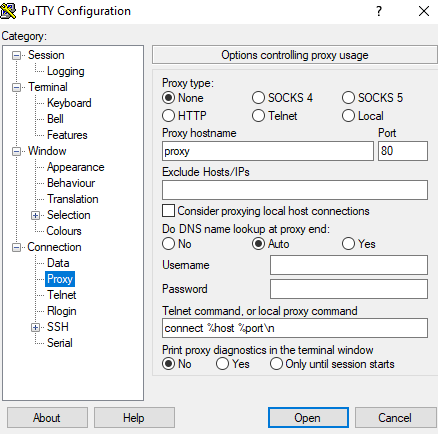

However Windows is really bad for connecting to the server directly:  

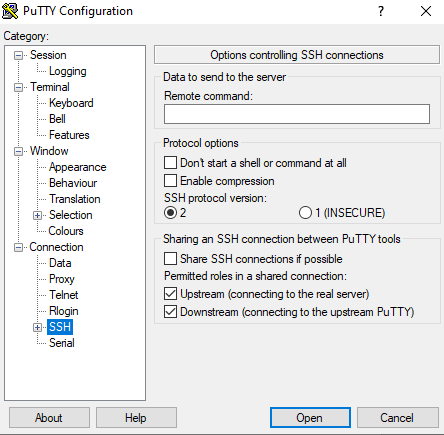

So you will need to download **Putty** and **puttygen here:**  
**https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html**



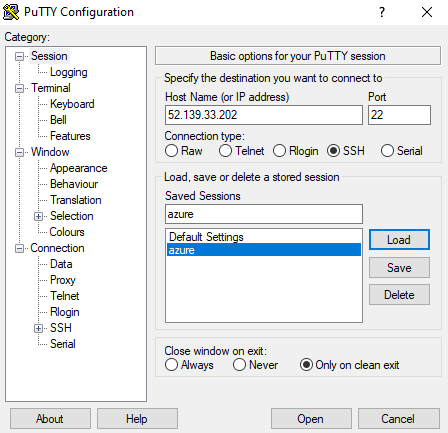
Load the key that you downloaded from Azure here, Generate and Save Private Key.   
  
**PUTTY CONFIGURATIONS:**







**LOGG INTO YOUR VIRTUAL MACHINE (REMEMBER TO HAVE THE MACHINE ON IN AZURE)**



**SETTUP YOUR VIRTUAL MACHINE and NGINX WEB SERVER**

sudo apt update -- these are the basic linux configurations

sudo apt upgrade -- this are the basic linux configurations python etc

sudo apt install nginx – this is the web server

**INSTALL GITHUB REPO**

git clone <https://github.com/Carlos-Gomez-Hernandez/git-public.git>

user: Carlos-Gomez-Hernandez

**CREATE YOUR VIRTUAL ENVIRORMENT:**

**Install pip**

sudo python3 -m pip install --user --upgrade pip

**Check python and pip are running**

python3 -m pip --version

python3 -m pip install --user --upgrade pip

python3 -m venv myNewWebApp

source myNewWebApp/bin/activate

**CREATE YOUR VIRTUAL ENVIRORMENT:**

python -m pip install Django

pip install django-extensions

pip install djangorestframework

**FOR CORS POLICY (OPEN API)**

python -m pip install django-cors-headers

**SETUP NGINX AS REVERSE PROXI**

[**https://linuxize.com/post/nginx-reverse-proxy/**](https://linuxize.com/post/nginx-reverse-proxy/)

cd /etc/nginx/sites-enabled

sudo vim default

The following screenshot displays the first attempt, to configure the reverse proxi. It workes however django rest framework won’t because ngnx do not like when applications call urls from different networks.

In order to fix this we needed to add the if statements in the second image to allow Access control **CORS policy settings**:

This code will tell NGNX to trust the Django-Rest Framework USLs. Later we will need to tell Django to trust **CORS**  
<https://www.ryadel.com/en/nginx-access-control-allow-origin-cors-policy-settings/>

server {

if ($request\_method = 'OPTIONS') {

add\_header 'Access-Control-Allow-Origin' '\*';

add\_header 'Access-Control-Allow-Credentials' 'true';

add\_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';

add\_header 'Access-Control-Allow-Headers' 'DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type';

add\_header 'Access-Control-Max-Age' 1728000;

add\_header 'Content-Type' 'text/plain charset=UTF-8';

add\_header 'Content-Length' 0;

return 204;

}

if ($request\_method = 'POST') {

add\_header 'Access-Control-Allow-Origin' '\*';

add\_header 'Access-Control-Allow-Credentials' 'true';

add\_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';

add\_header 'Access-Control-Allow-Headers' 'DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type';

}

if ($request\_method = 'GET') {

add\_header 'Access-Control-Allow-Origin' '\*';

add\_header 'Access-Control-Allow-Credentials' 'true';

add\_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS';

add\_header 'Access-Control-Allow-Headers' 'DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type';

}}





**SETUP NGINX AS REVERSE PROXI**  
At this point we should have NGINX configurations ready now we should be able to manage it.

**From Any Directory:**

**Every time you make a change in your nginx settings you need to restart**

sudo systemctl restart

**Active means all is good:**

sudo systemctl status

**Every time you make a change in your application**

sudo systemctl start carlosApp

sudo systemctl status carlosApp

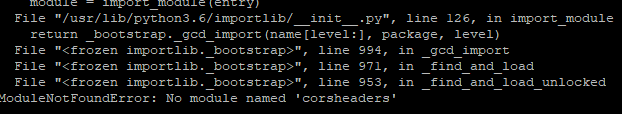
At this point you should be able to see your application in the internet using azure path.



**APLICATION DEBUGGING:  
  
To Test For Application Problems you need to first stop the server:**

sudo systemctl stop carlosApp

/home/cgomezhernandez/myNewWebApp/bin/python /home/cgomezhernandez/espacios/git-public/core/manage.py runserver



<https://medium.com/@dtkatz/3-ways-to-fix-the-cors-error-and-how-access-control-allow-origin-works-d97d55946d9>

<https://stackoverflow.com/questions/35760943/how-can-i-enable-cors-on-django-rest-framework>

<https://www.ryadel.com/en/nginx-access-control-allow-origin-cors-policy-settings/>

<https://github.com/adamchainz/django-cors-headers#configuration>

<https://stackoverflow.com/questions/58558804/django-cors-headers-and-nginx-config-preflight-response-missing-cors-headers>

**Automate NGINX to run the application even when the linux session cgomezhernandez is offline:  
  
Node Exporter Service File**https://www.digitalocean.com/community/tutorials/how-to-install-prometheus-on-ubuntu-16-04

**CHECK WHERE PYTHON IS LOCATED**

which python

*/usr/bin/python*

*(remember we have a virtual environment)*  
  
**GO TO**

cd etc/systemmd/system/

**CREATE AN APPLICATION CONFIGURATION FILE**

sudo vim carlosAPP.service

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  
ADD THE FOLLOWING CONFIGURATION SETTINGS AND ADJUST AS NEEDED  
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

[Unit]

Description=carlosApp

Wants=network-online.target

After=network-online.target

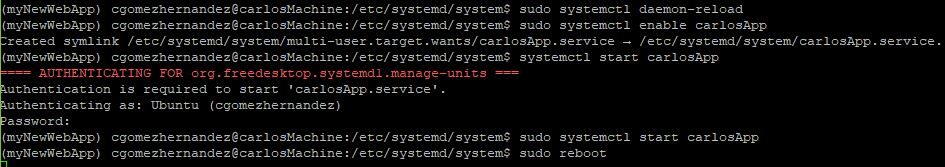
[Service]

Type=simple

ExecStart=**/home/cgomezhernandez/myNewWebApp/bin/python** **/home/cgomezhernandez/espacios/git-public/core/manage.py runserver**

[Install]

WantedBy=multi-user.target

TELL THE SYSTEM TTO READ THE UNIT FILE

**API CORS CANNOT FETCH PROBLEM:**



**INSTALL CORS HEADERS:**

**INSTALL THE DJANGO EXTENSION DJANGO-CORS-HEADERS**

python -m pip install django-cors-headers

*(remember we have a virtual environment)*  
  
**GO TO /git-public/core/core/settings.py**

INSTALLED\_APPS = [

...

'corsheaders',

...

]

MIDDLEWARE = [

'corsheaders.middleware.CorsMiddleware',

'django.middleware.common.CommonMiddleware',

...

]

*(THESE TWO SETTINGS MUST GO ON TOP OF EVERY OTHER SETTING)*

**OPEN THE PORT TO YOUR FRONT END SERVER**

CORS\_ORIGIN\_WHITELIST = [

'http://52.139.33.202'

]

python -m pip install django-cors-headers

**BACKUP:**

**Recursive Directory**

cp -r git-public/ git-public.back

**Single file**

cp fileName fileName.back