

How to move SPIRAL to a new Azure Ubuntu 22.04 server?

A helpful tutorial: <https://www.rosehosting.com/blog/how-to-install-flask-on-ubuntu-22-04-with-apache-and-wsgi/>.

Note that we store the app at `/var/www/html/`.

Create a VM and open the 80 (http) and 443 (https) ports (and 5000)

This tutorial may be helpful for this step: https://www.youtube.com/watch?v=HmKUCPRN0os&ab_channel=VRKsolutions

In the Azure portal:

- Follow the tutorial to create a VM. The current deployment uses a Standard E20as v5 (20 vcpus, 160 GB memory).
- In the Azure portal: VM -> Networking -> Add inbound security rule -> port 80, name it "allowHTTP"
- In the Azure portal: VM -> Networking -> Add inbound security rule -> port 443, name it "allowHTTPS"
- In the Azure portal: VM -> Networking -> Add inbound security rule -> port 5000, name it "allow5000" (only for now)
- In the Azure portal: VM -> Networking -> Make sure that SSH is open for everywhere (only for now)

Important: before beginning the installation, make sure that all the disks are allocated and mounted. The current deployment uses a 1T disk that is used for both the OS and data. However, the more "correct" way is to use a small (32G) OS disk, and a larger (let's say 1T) Data disk. In this case, you should mount the Data disk on some folder, let's say `/var2`, and then create a symbolic link from `/var` to `/var2`.

Some useful commands:

- `df -h` shows both the physical and attached disks.
- `lsblk` shows the physical disks and whether they are partitioned and mounted. So if there is "sdb" for example but it is not divided to "sdb1", "sdb1a", etc., then it is not partitioned. If it is divided, but for example "sdb1" has nothing under "MOUNTPOINTS", then it is not mounted.
- `df -lh` shows the actual attached disks (and the available space). So make sure that the "SIZE" fits what you expect.

Apache2

Open PuTTY and connect to the VM (public IP) with you name and password.

As in the tutorial:

Before installing the software, we need to update the system packages to the latest versions available.

```
sudo apt-get update -y && sudo apt-get upgrade -y
```

To install the Apache Web server execute the following command:

```
sudo apt install apache2 -y
```

Once installed, start and enable the service.

```
sudo systemctl enable apache2 && sudo systemctl start apache2
```

Check if the service is up and running:

```
sudo systemctl status apache2
```

Open the server IP in a browser. You should see the apache2 ubuntu default page.

Python

Verify Python installation:

```
python3 -V
```

you should see Python 3.10.X

Install python3-pip, python3-venv, python3-dev:

```
sudo apt-get install python3-pip python3-venv python3-dev
```

Clone SPIRAL repository, install packages and add needed files

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

As explained in the readme of https://github.com/hadasbi/SPIRAL_web_tool (steps 2-7):

```
sudo git clone https://github.com/hadasbi/SPIRAL_web_tool.git
cd SPIRAL_web_tool
sudo python3 -m venv spiral_venv
```

Activate the virtual environment:

```
source ./spiral_venv/bin/activate
```

Install required Python packages:

```
sudo ./spiral_venv/bin/pip install -r ./requirements.txt
```

Create analysis directory:

```
cd static_analysis
sudo mkdir analysis
cd ..
```

Move required files (that are not in Github for obvious reasons...) to the server:

The files are `FLASK-WTF_encryption_key.txt` and `mail_password.txt`.

In the SCP, move the files from your computer to `/home/SpiralAdmin` (since you can only write to `/var/www/html` in sudo). Then move them to their final location by:

```
sudo mv /home/SpiralAdmin/mail_password.txt ./mail_password.txt
sudo mv /home/SpiralAdmin/Flask-WTF_encryption_key.txt ./Flask-WTF_encryption_key.txt
```

Add the server name to list of servers running in production mode (in `main.py`)

In `main.py`, look for the line:

```
if hostname in ['bi-fiona', 'Spin']: # production (Linux)
```

And add `<SERVERNAME>`:

```
if hostname in ['bi-fiona', 'Spin', '<SERVERNAME>']: # production (Linux)
```

Check that the flask app is working:

A very important note: `app.config['SERVER_NAME']` in `/var/www/html/SPIRAL_web_tool/main.py` and `ServerName` in the configuration file `/etc/apache2/sites-available/spiral.technion.ac.il.conf` (see next section) have to match.

For now, in order to test the app, change `app.config['SERVER_NAME']` in `\var\www\html\SPIRAL_web_tool\main.py` to `<VM_PUBLIC_IP>:5000`, for example:

```
app.config['SERVER_NAME'] = '52.240.143.213:5000'
```

You can do this with the command:

```
sudo nano /var/www/html/SPIRAL_web_tool/main.py
```

Navigate the file and make the change.

To save, type `ctrl+X`, then `Y`, then enter.

Then you can test the app:

Test 1: make sure you have the venv activated and type:

```
python main.py
```

Test 2:

make sure you have the venv activated and type:

```
export FLASK_APP=main.py
flask run --host=0.0.0.0
```

For both tests you should see:

```
Running on all addresses.
WARNING: This is a development server. Do not use it in a production deployment.
Running on http://19.0.0.8:5000/ (Press CTRL+C to quit)
```

Open the browser and type: `<VM_PUBLIC_IP>:5000`

You should see SPIRAL's homepage.

Go back to the terminal, press `ctrl+C` and:

```
deactivate
```

What if it doesn't work?

Go to the last line in `main.py` and add `debug=True` to `app.run()`.

Then:

```
python file.py runserver -d
```

Pre-configuration step just for Windows

Skip this step. It is here in case SPIRAL would have to work from Windows one day... For Windows, you have to move `activate_this.py` file to the right location:

```
sudo cp activate_this.py ./spiral_venv/bin/activate_this.py
```

WSGI configuration for HTTP

Install `mod_wsgi`:

```
sudo apt-get install libapache2-mod-wsgi-py3
```

Create SPIRAL configuration file:

```
sudo touch /etc/apache2/sites-available/spiral.technion.ac.il.conf
```

To edit the file, open it with `nano`:

```
sudo nano /etc/apache2/sites-available/spiral.technion.ac.il.conf
```

And paste this:

```
<VirtualHost *:80>
    ServerName spiral.technion.ac.il
    ServerAlias <VM_PUBLIC_IP>
    DocumentRoot "/var/www/html/SPIRAL_web_tool/"

    WSGIDaemonProcess spiral user=www-data group=www-data threads=28 python-home="/var/www/html/SPIRAL_web_tool/spiral_venv"
    WSGIScriptAlias / "/var/www/html/SPIRAL_web_tool/"
    WSGIScriptAlias / "/var/www/html/SPIRAL_web_tool/app.py"

    ErrorLog /var/log/apache2/spiral.technion.ac.il-error_log
    CustomLog /var/log/apache2/spiral.technion.ac.il-access_log combined

    <Directory "/var/www/html/SPIRAL_web_tool">
        WSGIProcessGroup spiral
        WSGIApplicationGroup %GLOBAL%
        Order deny,allow
        Require all granted
    </Directory>
    LogLevel warn
    <VirtualHost>

Don't forget to replace <VM_PUBLIC_IP> with the IP.
```

Also, go over all the lines and make sure that nothing weird happened in the pasting process: lines out short, characters added...

To save, type `ctrl+X`, then `Y`, then enter.

Then, enable the Apache2 configuration file:

```
sudo a2ensite spiral.technion.ac.il.conf
```

Check the syntax of the Apache2 configuration:

```
sudo apachectl -t
```

You should see:

```
Syntax OK
```

Restart Apache:

```
sudo systemctl restart apache2
```

Remove `index.html` (the default Apache page) from `/var/www/html/`:

```
sudo mv /var/www/html/index.html /home/SpiralAdmin/index.html
```

Change the ownership of the relevant folder to the user that is supposed to run the app:

```
sudo chown -R www-data:www-data /var/www/
```

Change server name:

Change both `app.config['SERVER_NAME']` in `/var/www/html/SPIRAL_web_tool/main.py` and `ServerName` in the configuration file `/etc/apache2/sites-available/spiral.technion.ac.il.conf` to `<VM_PUBLIC_IP>` (the later one should be done, just make sure).

Check if it's working:

Restart Apache:

```
sudo systemctl restart apache2
```

Open http://<VM_PUBLIC_IP>/ in a browser. SPIRAL's homepage should appear.

WSGI configuration for HTTPS

Follow this tutorial: <https://www.digitalocean.com/community/tutorials/how-to-create-a-self-signed-ssl-certificate-for-apache-in-ubuntu-22-04>. In step 2, under `common name`, put `<VM_PUBLIC_IP>`.

Change the conf file `/etc/apache2/sites-available/spiral.technion.ac.il.conf` to:

```
<VirtualHost *:80>
    ServerName <VM_PUBLIC_IP>
    ServerAlias www.spiral.technion.ac.il
    Redirect / https://<VM_PUBLIC_IP>/
    <VirtualHost>

<VirtualHost *:443>
    ServerName spiral.technion.ac.il
    ServerAlias <VM_PUBLIC_IP>
    ServerAlias www.spiral.technion.ac.il
    DocumentRoot "/var/www/html/SPIRAL_web_tool/"

    WSGIDaemonProcess spiral user=www-data group=www-data processes=19 maximum-requests=100 python-home="/var/www/html/SPIRAL_web_tool/spiral_venv"
    WSGIScriptAlias / "/var/www/html/SPIRAL_web_tool/"
    WSGIScriptAlias / "/var/www/html/SPIRAL_web_tool/app.py"

    ErrorLog /var/log/apache2/spiral.technion.ac.il-error_log
    CustomLog /var/log/apache2/spiral.technion.ac.il-access_log combined

    <Directory "/var/www/html/SPIRAL_web_tool">
        WSGIProcessGroup spiral
        WSGIApplicationGroup %GLOBAL%
        Order deny,allow
        Require all granted
    </Directory>
    LogLevel warn

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
    <VirtualHost>
```

Don't forget to replace `<VM_PUBLIC_IP>` with the IP.

Restart Apache:

```
sudo systemctl restart apache2
```

At this point the website should be working when you type in your browser either http://<VM_PUBLIC_IP>/ or https://<VM_PUBLIC_IP>/. If this is not the case, try to debug (take a look at "Common issues" at the end of this document). Do not proceed until it is working.

Send an email to the DNS record

Send an email to `dns@technion.ac.il` and ask to point `www.spiral.technion.ac.il` and `spiral.technion.ac.il` to the IP of the new server (`<VM_PUBLIC_IP>`).

Generate a Technion SSL certificate

To issue an SSL certificate, you first have to generate a CSR file. Use this command:

```
openssl req -new -newkey rsa:2048 -nodes -keyout spiral.technion.ac.il.key -out spiral.technion.ac.il.csr
```

(This tutorial is helpful: <https://phoenixnap.com/kb/generate-openssl-certificate-signing-request>)

- Country Name - use a 2-letter country code (US for the United States): `IL`
- State - the state in which the domain owner is incorporated: `ISRAEL` (not important)
- Locality - the city in which the domain owner is incorporated: `Haifa` (not important)
- Organization name - the legal entity that owns the domain: `Technion` (not important)
- Organizational unit name - the name of the department or group in your organization that deals with certificates: `CIS` (not important)
- Common name - typically the fully qualified domain name (FQDN), i.e. what the users type in a web browser to navigate to your website: `*.spiral.technion.ac.il` (SUPER important)

Email address - the webmaster's email address: `hadas.noriah@gmail.com`

Challenge password: `123456789`

This process creates two files:

- a private key: `spiral.technion.ac.il.key`
- a csr file: `spiral.technion.ac.il.csr`.

You have to send the csr file to `CyberSecurity@technion.ac.il` (or: `cis@technion.ac.il`) and ask them to issue an SSL certificate for you. Another option is to ask Dan Bilberg (`ardan@technion.ac.il`) to issue a certificate since he has access to the SECTIGO certificate system (<https://ds.technion.ac.il/sectigo/>). They don't give access to students. The certificate you need is the "Certificate only, PEM encoded".

When you have the Technion certificate and the DNS has been set:

- Copy the private key to `/etc/ssl/private/spiral.technion.ac.il.key`
- Copy the certificate that was issued for you (a 'cer' or 'crt' file) to `/etc/ssl/certs/_spiral.technion.ac.il.cert.cer`
- Change in `/var/www/html/SPIRAL_web_tool/main.py` the configuration of `ServerName` to `spiral.technion.ac.il`:

```
sudo nano /var/www/html/SPIRAL_web_tool/main.py
```

and then change so that:

```
app.config['SERVER_NAME'] = 'spiral.technion.ac.il'
```

Make some changes in the SPIRAL configuration file:

```
sudo nano /etc/apache2/sites-available/spiral.technion.ac.il.conf
```

and then change the values of `ServerName`, `ServerAlias`, `Redirect`, `SSLCertificateFile`, `SSLCertificateKeyFile` in both port 80 and port 443:

```
ServerName spiral.technion.ac.il
```

```
ServerAlias www.spiral.technion.ac.il
```

```
Redirect / https://spiral.technion.ac.il/
```

```
SSLCertificateFile /etc/ssl/certs/_spiral.technion.ac.il.cert.cer
```

```
SSLCertificateKeyFile /etc/ssl/private/spiral.technion.ac.il.key
```

That should be enough... But since `ServerAlias` in port 443 did not work for me (that is, <https://www.spiral.technion.ac.il> did not find the website), I resorted to multiplying the port 443 section, ultimately creating the following final SPIRAL configuration file:

```
<VirtualHost *:80>
    ServerName spiral.technion.ac.il
    ServerAlias -spiral.technion.ac.il
    Redirect / https://spiral.technion.ac.il/
    <VirtualHost>

<VirtualHost *:443>
    ServerName spiral.technion.ac.il
    ServerAlias www.spiral.technion.ac.il

    DocumentRoot "/var/www/html/SPIRAL_web_tool/"

    WSGIDaemonProcess spiral user=www-data group=www-data processes=19 maximum-requests=100 python-home="/var/www/html/SPIRAL_web_tool/spiral_venv"
    WSGIScriptAlias / "/var/www/html/SPIRAL_web_tool/"
    WSGIScriptAlias / "/var/www/html/SPIRAL_web_tool/app.py"

    ErrorLog /var/log/apache2/spiral.technion.ac.il-error_log
    CustomLog /var/log/apache2/spiral.technion.ac.il-access_log combined

    <Directory "/var/www/html/SPIRAL_web_tool">
        WSGIProcessGroup spiral
        WSGIApplicationGroup %GLOBAL%
        Order deny,allow
        Require all granted
    </Directory>
    LogLevel warn
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/_spiral.technion.ac.il.cert.cer
    SSLCertificateKeyFile /etc/ssl/private/wildcard8.spiral.technion.ac.il.key
    <VirtualHost>
```

- Sometimes you need to add the row

```
ServerName 127.0.0.1
```

to the end of the Apache2 configuration file `/etc/apache2/apache2.conf`.

- Restart Apache2:

```
sudo systemctl restart apache2
```

Some last tasks...

- In the Azure portal: remove the port 5000 rule.

In the Azure portal: Make sure that SSH is only open for a list of IP's (enter your IP). Remember this... next time you want to connect via SSH you'll need to first go to the Azure portal and enter your IP in the list of allowed IP's.

- And you're done! Hallelujah!

Common issues...

What to do if something isn't working?

Check the error logs (both SPIRAL's and Apache's):

```
cat /var/log/apache2/spiral.technion.ac.il-error_log
```

```
cat /var/log/apache2/error_log
```

Azure portal:

make sure that your NIC and your subnet are under the same NSG.

Don't install `nginx`, as it can create issues by making the ports unavailable for Apache2.

If `nginx` is installed you have to completely remove it (see <https://www.cyberciti.biz/faq/remove-uninstall-nginx-from-ubuntu-debian-linux-apt-get-command/>).

```
In [9]: # Check response time
import requests, time

for url in ['http://www.spiral.technion.ac.il/', 'https://spiral.technion.ac.il/', 'https://52.240.143.213/',
           'https://www.spiral.technion.ac.il/', 'https://spiral.technion.ac.il/', 'https://52.240.143.213/']:
    t0 = time.time()
    response = requests.get(url, verify=False)
    t1 = time.time()
    total = t1-t0
    print(url, response.get_request('total'))

c:\program files (x86)\microsoft visual studio\shared\python37_64\lib\site-packages\urllib3\connectionpool.py:1020: InsecureRequestWarning: Unverified HTTPS request is being made to host 'spiral.technion.ac.il'. Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
InsecureRequestWarning:
https://spiral.technion.ac.il
Single get request took 0.73133447860718

c:\program files (x86)\microsoft visual studio\shared\python37_64\lib\site-packages\urllib3\connectionpool.py:1020: InsecureRequestWarning: Unverified HTTPS request is being made to host 'spiral.technion.ac.il'. Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
InsecureRequestWarning:
https://spiral.technion.ac.il
Single get request took 0.4449386596796875

c:\program files (x86)\microsoft visual studio\shared\python37_64\lib\site-packages\urllib3\connectionpool.py:1020: InsecureRequestWarning: Unverified HTTPS request is being made to host 'spiral.technion.ac.il'. Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
InsecureRequestWarning:
https://spiral.technion.ac.il
Single get request took 0.9989325803745117

c:\program files (x86)\microsoft visual studio\shared\python37_64\lib\site-packages\urllib3\connectionpool.py:1020: InsecureRequestWarning: Unverified HTTPS request is being made to host 'www.spiral.technion.ac.il'. Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
InsecureRequestWarning:
https://spiral.technion.ac.il
Single get request took 0.4449386596796875

c:\program files (x86)\microsoft visual studio\shared\python37_64\lib\site-packages\urllib3\connectionpool.py:1020: InsecureRequestWarning: Unverified HTTPS request is being made to host '4.227.224.198'. Adding certificate verification is strongly advised. See: https://urllib3.readthedocs.io/en/1.26.x/advanced-usage.html#ssl-warnings
InsecureRequestWarning:
```

In []: