

VisLab App Deployment Guide

Machines

We currently have 3 machines:

1. nyuvis-web.poly.edu: This is our main HTTP server for the lab projects.
2. nyuvis-web2.poly.edu: This is an HTTP server for students projects
3. nyuvis-db.poly.edu: This is our main Database server, it is also the server to use when online high demanding computations is needed. For off-line computations we recommend the [NYU HPC](#) or the [VGC Hadoop Cluster](#)

Accessing the machines

1. To access the machines it is required to be on the NYU network, if working from outside the campus you can connect using the [NYU VPN](#) service.
2. Access to the machines can be done using ssh
3. File transfer is also possible through the SFTP protocol allowing the use of softwares like [ForkLift](#) or [Cyberduck](#) to transfer file, as well the scp command

App types

- a. **Static:** static pages are the ones that don't require an exclusive server running, this includes apps totally developed in HTML and Javascript
- b. **Service:** This type of deployment requires some sort of server running on an specific port. This is common for Python and NodeJS apps.
 - i. **Port:** This type of app requires a dedicated port, the ports will be assigned on intervals of 10 growing from 15000. This gives some room for additional ports in case the application needs to run a second network service.

Deploying the app

The projects should be deployed to the nyuvis-web machine unless high demanding computation is required.

The source code for each project should be stored on the /projects folder on its own folder.

We recommend:

1. Keep the development code on our GitHub organization.
2. Clone the code to a folder with the same name on the server .
3. Publish the app to a path on the server with the same name.

Using script

To facilitate the process and guarantee consistency, on the path `/projects` there is an script that adds the application to the server.

To execute it just type `sudo ./add.sh`

The script will perform the following operations that is:

1. Ask for:
 - a. Name: A presentable name for your application.
 - b. Description: A short description to help others to know what the app is about
 - c. URLPath: this is the name that will be part of the URL for your applications, this name will also be used to create the folder, as repository name to clone from GitHub and in case of running a service, as name of the service.
2. Ask if it should just create a folder or clone from a GitHub repository.
3. Ask if it is static or a service
 - a. If a **service**:
 - i. Select a port for the service
 - ii. Create a file named `index.php` to route calls to the service port
 - iii. Create an `.htaccess` file to protect the code and redirect calls
 - iv. If selected to create the service file (sudo required):
 1. As for an initialization command
 2. Create a file named `app.service` with the configuration for a `systemd` service
 3. Create a symbolic link from `/etc/systemd/system/{appUrl}.service` to the `app.service` file
 4. A file named `app.readme` will be created containing information about the app including port and how to start the service
 5. And entrance will be created on the `/projects` page pointing to the app. If you don't want the app to be indexed. Just remove the file with your application name from inside `/projects/.apps`

Btw use `systemctl start yourapp` and other subcommands to manually control the app starting / restarting etc.

Computing demanding apps.

The machine hosting our proxy server has only 1GB of ram, for this reason apps that require high memory and computation power should run on `nyuvis-db.poly.edu` server.

In this case follow these steps

- On **nyuvis-web.poly.edu**
 - Run the script:
 - When asked to clone or create folder, select 2) Create folder
 - When asked if the app runs as a server, select 1) Run as a service
 - When asked if it will have automatic initialization, select 2) No
 - On the end of the script take note of the **PORT** selected for your app.
- On **nyuvis-db.poly.edu**
 - Go to `/projects`
 - Create a folder with the same name you used on the `nyuvis-web` server
 - Clone, download or upload your code or software
 - Configure the application to use the PORT provided on the initial script
 - Start the app

Authentication

In case where you want to protect your app with user and password, it is possible to activate authentication using `htaccess`.

Use this link: <http://www.htaccesstools.com/htaccess-authentication/> to create a configuration and put it on the `.htaccess` on the project folder.

And this link: <http://www.htaccesstools.com/httpasswd-generator/> to create a file with users and passwords.

Reverse Proxy with password

Use `sudo` as needed and replace `yourproject` and the port with your info
Add line in `/etc/httpd/conf.d/vhost.conf`

Include `./conf.d/projects/yourproject.conf`

And add the conf file in `/etc/httpd/conf.d/projects/`

```
<Location /projects/yourproject>
  AuthType Basic
  AuthName "Explanation Explorer"
  AuthBasicProvider file
  AuthUserFile "/projects/yourproject/passwords"
  Require user nyuvis
  ProxyPass http://localhost:15200
```

</Location>

Or in case of no user/password

<Location /projects/yourproject>

ProxyPass http://localhost:15200

</Location>

Create passwords with htpasswd -c /projects/yourproject/passwords nyuvis

Restart server with httpd -k restart