

# Configuring s2p in jasmin cloud

## Getting Started

This follows all the steps to [deploy in the jasmin cloud](#).

We need to login to the server to do this. It's helpful to have two terminal windows. One looking at the nginx logs:

```
1 ssh -A root@192.171.139.78
2 tail -F /var/log/nginx/error.log
```

and one running as the django user

```
1 ssh -A root@192.171.139.78
2 su YOUR_USER
3 cd
4 source django/bin/activate
```

## Configuring Static Files

You may or may not have set up static files properly in your app, but they need to end up in your static directory. After configuration, I was faced with, e.g. this:

```
1 2022/07/24 07:27:52 [error] 28292#28292: *11 open()
   "/home/s2p/staff2proj/static/static/projdb/style.css" failed (2: No such file or directory),
   client: 87.115.254.169, server: _, request: "GET /static/projdb/style.css HTTP/1.1", host:
   "192.171.139.78", referer: "http://192.171.139.78/accounts/login/?next=/projdb/staff/"
2 (2: No such file or directory), client:
```

At this point what I needed to do was go back and change my static file for nginx to this:

```
1 location /static/
2     {
3         root /home/s2p;
4     }
```

(and fixed now in the ansible template) and modified my `settings.py` to have a 'STATIC\_ROOT' section:

```
1 STATIC_URL = '/static/'
2 STATICFILES_DIRS = (os.path.join(BASE_DIR, "static"),)
3 STATIC_ROOT='../static/'
```

noting that we needed the "deployment" static root (STATIC\_ROOT) to be not the same as STATIC\_URL, so we could run the following command:

```
1 python manage.py collectstatic
```

which put all the static files in the right place. They are now being served ok, but don't appear to be doing anything!

Ah, it's an nginx problem, I'd removed an important line about mime types.

led to an error like:

```
1 Did not parse stylesheet at ... because non CSS MIME types are not allowed in strict mode
```

Fixed in the template now.

## Setting up users

At this point we just have a super user, we need me and Grenville to have access! We also need access for some automatic uploads.

Login to `http:192.171.139.78/admin` with the credentials you used in step 5 of the centos ansible stuff.

Bugger. CSRF problem??? Ah, ok, we're not on https:

Check that you're either on https or you have `CSRF_COOKIE_SECURE=False` (which is the default) in settings, otherwise your csrf cookie exists but won't be sent. Purge your cookies after changing `CSRF_COOKIE_SECURE`.

To fix this I added the following lines to the [settings.py](#), and restarted gunicorn.

```
1 # While we're on https, we may need this:
2 CSRF_COOKIE_SECURE=False
3 CSRF_TRUSTED_ORIGINS = ['http://192.171.139.78']
```

```
1 systemctl restart gunicorn
```