

Specific Cells parameters

During your Pydio Cells installation (actually at the beginning) you will be prompted for 2 important parameters:

- **Internal Url:** is the url used to bind your cells to the server.
- **External Url:** is the url used to access your cells from outside.

In this case for the reverse proxy, we must pay extra attention to the **External URL** as it has to match the address of the reverse proxy endpoint that is going to be used to access Cells.

For instance if **Cells** is running under **192.168.0.12** and the **Reverse proxy 192.168.1.201** both running on 2 completely different networks and/or publicly exposed.

Note: in the examples below we use ip addresses but you can also use domains (make sure that they are reachable)

Cells Server	Reverse proxy Server
192.168.0.12	192.168.1.201

Hence on **Cells Server** we use the following values upon installation, we also want to access Cells through **https**.

(assuming that we are binding Cells on port 8080, internal_url)

Internal URL	External URL
192.168.0.12:8080	https://192.168.1.201

To resume, we will access **Cells** through `https://192.168.1.201` while Cells is actually running on another server (192.168.0.12).

Basic NGINX reverse proxy configuration

```

server {
    client_max_body_size 200M;
    server_name example.pydio.com;

    location / {
        proxy_buffering off;
        proxy_pass https://192.168.0.1$request_uri;
        #proxy_pass_request_headers on;
        #proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    location /ws {
        proxy_buffering off;
        proxy_pass https://192.168.0.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_read_timeout 86400;
    }

    error_log /var/log/nginx/cells-proxy-error.log;
    access_log /var/log/nginx/cells-proxy-access.log;

    listen [::]:443 ssl http2;
    listen 443 ssl http2;
    ssl_certificate www.example.com.crt;
    ssl_certificate_key www.example.com.key;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!MD5;
}

server {
    if ($host = example.pydio.com) {
        return 301 https://$host$request_uri;
    }

    listen 80 http2;
    listen [::]:80 http2;
    server_name example.pydio.com;
    return 404;
}

```

Cells Sync

Mandatory section for the Sync Client to work behind a Nginx reverse proxy.

If your Cells Server is running behind a Nginx reverse proxy you must meet 2 requirements and then add the config below to your main nginx reverse proxy configuration.

TLS and HTTP2 meaning that the reverse proxy and Cells must communicate with SSL (you can use the self signed option during installation).

Once that is done you must set a port for **grpc** in this example it's **33060**,

to set it you have the following env variable, `PYDIO_GRPC_EXTERNAL=33060`

otherwise you can set it when running the binary with the following flag `--grpc_external`, for instance;

```
./cells start --grpc_external=33060
```

In all those examples you can substitute the port 33060 by the port of your choice

Also make sure to put the **address/domain** on which your **Cells Server** is running (refer to the arrays above) line **grpc_pass**.

```
server {
    listen 33060 ssl http2;
    listen [::]:33060 ssl http2;
    ssl_certificate      www.example.com.crt;
    ssl_certificate_key  www.example.com.key;
    ssl_protocols        TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers           HIGH:!aNULL:!MD5;

    location / {
        grpc_pass grpc://192.168.0.1:33060;
    }

    error_log /var/log/nginx/proxy-grpc-error.log;
    access_log /var/log/nginx/proxy-grpc-access.log;
}
```

Below you can have a look at the complete file

Finale note

Make sure to substitute the values of the **certificates** and **ip/domains**.

The complete configuration

cells.conf

```
server {
    client_max_body_size 200M;
    server_name example.pydio.com;
```

```

    location / {
        proxy_buffering off;
        proxy_pass https://192.168.0.1$request_uri;
        #proxy_pass_request_headers on;
        #proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }

    location /ws {
        proxy_buffering off;
        proxy_pass https://192.168.0.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_read_timeout 86400;
    }

error_log /var/log/nginx/cells-proxy-error.log;
access_log /var/log/nginx/cells-proxy-access.log;

listen [::]:443 ssl http2;
listen 443 ssl http2;
ssl_certificate www.example.com.crt;
ssl_certificate_key www.example.com.key;
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_ciphers HIGH:!aNULL:!MD5;
}

server {
    if ($host = example.pydio.com) {
        return 301 https://$host$request_uri;
    }

    listen 80 http2;
    listen [::]:80 http2;
    server_name example.pydio.com;
    return 404;
}

server {
    listen 33060 ssl http2;
    listen [::]:33060 ssl http2;
    ssl_certificate www.example.com.crt;
    ssl_certificate_key www.example.com.key;
    ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!MD5;

    location / {
        grpc_pass grpcs://192.168.0.1:33060;
    }
}

```

```
}  
  
error_log /var/log/nginx/proxy-grpc-error.log;  
access_log /var/log/nginx/proxy-grpc-access.log;  
}
```