

# DFS IO handler

DFS is a requirement for use in Clustering setup. See [Clustering](#) for an overview of the feature.

## What it is meant for

The DFS IO handler (`legacy_dfs_cluster`) can be used to store binary files on an NFS server. It will use a database to manipulate metadata, making up for the potential inconsistency of network based filesystems.

## Configuration

You need to configure both metadata and binarydata handlers.

As the binarydata handler, create a new Flysystem local adapter configured to read/write to the NFS mount point on each local server. As metadata handler handler, create a dfs one, configured with a doctrine connection. We recommend that a dedicated database is used for DFS metadata. In our example, we will use one named `dfs`.

```
# new doctrine connection for the dfs legacy_dfs_cluster metadata handler.
doctrine:
  dbal:
    connections:
      dfs:
        driver: pdo_mysql
        host: 127.0.0.1
        port: 3306
        dbname: ezdfs
        user: root
        password: "rootpassword"
        charset: UTF8

# declare the handlers
ez_io:
  binarydata_handlers:
    nfs:
      flysystem:
        adapter: nfs_adapter
  metadata_handlers:transp
  dfs:
    legacy_dfs_cluster:
      connection: doctrine.dbal.dfs_connection

# set the handlers

ezpublish:
  system:
    default:
      io:
        metadata_handler: dfs
        binarydata_handler: nfs
```

## Customizing the storage directory

eZ Publish 5.x required the NFS adapter directory to be set to `$var_dir/$storage_dir` part for the NFS path. This is no longer required with eZ Platform, but the default prefix used to serve binary files will still match this expectation.

If you decide to change this setting, make sure you also set `io.url_prefix` to a matching value. If you set the NFS adapter's directory to `"/path/to/nfs/storage"`, use this configuration so that the files can be served by Symfony:

```
ezpublish:
  system:
    default:
      io:
        url_prefix: "storage"
```

As an alternative, you may serve images from NFS using a dedicated web server. If in the example above, this server listens on <http://static.example.com> and uses `/path/to/nfs/storage` as the document root, configure `io.url_prefix` as follows:

```
ezpublish:
  system:
    default:
      io:
        url_prefix: "http://static.example.com/"
```

You can read more about that on [Binary files URL handling](#).

## Web server rewrite rules.

The default eZ Platform rewrite rules will let image requests be served directly from disk. With native support, files matching `^/var/([^/]+)/storage/images(-versioned)?/.*` have to be passed through `/web/app.php`.

In any case, this specific rewrite rule must be placed without the ones that "ignore" image files and just let the web server serve the files.

### Apache

```
RewriteRule ^/var/([^/]+)/storage/images(-versioned)?/.* /app.php [L]
```

### nginx

```
rewrite "^/var/([^/]+)/storage/images(-versioned)?/(.*)" "/app.php" break;
```