Legacy DFS cluster

5.4 / 2014.11

Use of DFS clustering is a requirement for use in Clustering setup, for overview of clustering feature see Clustering.

What it is meant for

The `legacy_dfs_cluster` IO metadata handler can be used to store images on NFS, while remaining compatible with legacy, using the DFS Cluster file handler. It stores metadata in the `ezdfsfile` table from the legacy database. It is meant to be used to write binarydata to a locally mounted NFS server.

Configuration

> **Note for use with legacy**
> This handler requires legacy to be configured to use DFS clustering.

Assuming that your database is named `ezdfs`, configure it, for instance in `ezpublish.yml`:

```
# set the handlers
ezpublish:
    system:
        default:
            io:
                metadata_handler: dfs
                binarydata_handler: nfs

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

# 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

# new flysystem adapter for the nfs metadata handler
oneup_flysystem:
    adapters:
        nfs_adapter:
            local:
                # The last part, $var_dir$/$storage_dir$, is required for legacy
compatibility
                directory: "/path/to/nfs/$var_dir$/$storage_dir$"
```

> **Important**: take good note of the `$var_dir$/$storage_dir$` part for the NFS path. Legacy expects this path to exist on the NFS mount in order to be able to read and write files.

Web server rewrite rules.

The default eZ Publish rewrite rules will let image requests be served directly from disk. With native support, files matching `^/var/([^/]+/)?storage/images(-versioned)?/.*` have to be sent to the normal index.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)?/.* /index.php [L]
```

nginx

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