

# Using Varnish

eZ Platform being built on top of Symfony, it uses standard HTTP cache headers. By default the Symfony reverse proxy, written in PHP, is used to handle cache, but it can be easily replaced with any other reverse proxy like Varnish.

Use of Varnish is a requirement for use in Clustering setup, for overview of clustering feature see [Clustering](#).

- Prerequisites
- Recommended VCL base files
- Configure eZ Publish
  - Update your Virtual Host
    - On apache:
    - On nginx:
  - Update YML configuration

## Prerequisites

- A working Varnish 3 or Varnish 4 setup.

## Recommended VCL base files

For Varnish to work properly with eZ, you'll need to use one of the provided files as a basis:

- [Varnish 3 VCL example](#)
- [Varnish 4 VCL example](#)

**Note:** *Http cache management is done with the help of [FOSHttpCacheBundle](#). You may need to tweak your VCL further on according to [FOSHttpCache documentation](#) in order to use features supported by it.*

## Configure eZ Publish

### Update your Virtual Host

#### On apache:

##### my\_virtualhost.conf

```
<VirtualHost *:80>
    # Configure your VirtualHost with rewrite rules and stuff

    # Force front controller NOT to use built-in reverse proxy.
    SetEnv USE_HTTP_CACHE 0

    # Configure IP of your Varnish server to be trusted proxy
    # Replace fake IP address below by your Varnish IP address
    SetEnv TRUSTED_PROXIES "193.22.44.22"
</VirtualHost>
```

#### On nginx:

## mysite.com

```
fastcgi_param USE_HTTP_CACHE 0;
# Configure IP of your Varnish server to be trusted proxy
# Replace fake IP address below by your Varnish IP address
fastcgi_param TRUSTED_PROXIES "193.22.44.22";
```

## Update YML configuration

### ezplatform.yml

```
ezpublish:
  http_cache:
    purge_type: http

system:
  # Assuming that my_siteaccess_group your frontend AND backend siteaccesses
  my_siteaccess_group:
    http_cache:
      # Fill in your Varnish server(s) address(es).
      purge_servers: [http://my.varnish.server:6081]
```