Session

Sessions are handled by the Symfony2 framework, specifically API and underlying session handlers provided by HTTP Fundation component[1][2], this is further enhanced in eZ Publish with support for siteaccess aware session cookie settings.

Session handlers

In Symfony, session handler is configured using `framework.session.handler_id.` By default Symfony's *NativeFileSessionHandler*[3] is configured in eZ Publish, but Symfony can also be configured to use custom handlers[2], or just fallback to what is configured in PHP by setting it to null (~).

Single server setup

For single server, default settings should be preferred. As implied above they setup PHP's builtin "files" session save handler, and specifically configures it to use `session.save_path` set to `ezpublish/sessions` by default (`ezpublish/cache/<env>` before 5.3).

> **Session Garbage collection on Debian & Ubuntu**
> Debian based distros disables session.gc_probability by default an uses cronjob instead to clear sessions files. As we use custom save_path for sessions that would normally be a problem, however default Symfony settings makes sure to re enable this in `framework.session.gc_probability` so it should work by garbage collecting sessions for each 100th request out of the box.

Cluster setup

For Cluster setup we need to configure Sessions to use a backend that is shared between web servers, and supports locking. Only options out of the box supporting this is PHP memcached session save handler provided by php-memcached extension, and Symfony session handler for PDO (database).

Storing sessions in Memcached using php-memcached

For setting up eZ Publish using this memcached you'll need to configure the session save handler settings in php.ini as documented here, optionally tweak php-memcached session settings, and lastly disable Symfony session handler like:

```
framework:
    session:
        # handler_id set to null like here will use default session handler from
php.ini
        handler_id: ~
```

Alternative storing sessions in database using PDO

While not currently our recommendation from performance perspective, for setups where where Database is preferred for storing Sessions, you may use Symfony's PdoSessionHandler.
Below is an configuration example for eZ Publish, but please refer to documented in Symfony Cookbook documentation for full documentation.

```
framework:
    session:
        # ...
        handler_id: session.handler.pdo

parameters:
    pdo.db_options:
        db_table:    session
        db_id_col:   session_id
        db_data_col: session_value
        db_time_col: session_time

services:
    pdo:
        class: PDO
        arguments:
            dsn:      "mysql:dbname=<mysql_database>"
            user:     <mysql_user>
            password: <mysql_password>

    session.handler.pdo:
        class:
Symfony\Component\HttpFoundation\Session\Storage\Handler\PdoSessionHandler
        arguments: ["@pdo", "%pdo.db_options%"]
```

References

1. Cookbook Session recipes (symfony.com)
2. HTTP Fundation Component documentation (symfony.com)
3. Source code of NativeFileSessionHandler (github.com), aka `session.handler.native_file` service
4. Cookbook Configuration recipe for setting-up PdoSessionHandler (symfony.com), aka `session.handler.pdo` service