

# Redis gebruiken

Redis is een soort website caching systeem. Te gebruiken om de website wat sneller te maken.

- [Installatie](#)

# Installatie

## 1: Redis installeren op lokale ddev

```
ddev get ddev/ddev-redis  
ddev restart
```

## 2: Drupal redis module installeren

```
ddev composer require drupal/redis  
ddev drush en redis
```

## 3: Redis instellen in My Combell

- Open project in Combell, ga naar 'Performance & Caching' en activeer Redis
- Genereer een wachtwoord, kopieer dit en sla dit op in projects.brandle.be. Klik dan op 'Activeer Redis'

## 4: Settings files aanpassen voor (development, staging, production)

Gebruik de host en poort die je op Combell ziet voor Redis bij `$settings['redis.connection']`.

Verwijder andere `$settings['cache']['bins']` uit de settings file.

Verwijder ook andere `$settings['container_yamls'][] = ...` onder het redis gedeelte!

```
<?php  
  
/**  
 * @file  
 */  
  
use Drupal\Core\Installer\InstallerKernel;  
  
/**
```

```

* Use Redis.
*/
if (!InstallerKernel::installationAttempted() && extension_loaded('redis') &&
class_exists('Drupal\redis\ClientFactory')) {
    // Set Redis as the default backend for any cache bin not otherwise specified.
    $settings['cache']['default'] = 'cache.backend.redis';
    $settings['redis.connection']['host'] = '176.62.170.240';
    $settings['redis.connection']['port'] = 10069;
    $settings['redis.connection']['password'] = 'xxxxxxx';

    // Apply changes to the container configuration to better leverage Redis.
    // This includes using Redis for the lock and flood control systems, as well
    // as the cache tag checksum. Alternatively, copy the contents of that file
    // to your project-specific services.yml file, modify as appropriate, and
    // remove this line.
    $settings['container_yamls'][] = 'modules/contrib/redis/example.services.yml';

    // Allow the services to work before the Redis module itself is enabled.
    $settings['container_yamls'][] = 'modules/contrib/redis/redis.services.yml';

    // Manually add the classloader path, this is required for the container cache bin
    definition below
    // and allows to use it without the redis module being enabled.
    $class_loader->addPsr4('Drupal\\redis\\', 'modules/contrib/redis/src');

    // Use redis for container cache.
    // The container cache is used to load the container definition itself, and
    // thus any configuration stored in the container itself is not available
    // yet. These lines force the container cache to use Redis rather than the
    // default SQL cache.
    $settings['bootstrap_container_definition'] = [
        'parameters' => [],
        'services' => [
            'redis.factory' => [
                'class' => 'Drupal\redis\ClientFactory',
            ],
            'cache.backend.redis' => [
                'class' => 'Drupal\redis\Cache\CacheBackendFactory',
                'arguments' => ['@redis.factory', '@cache_tags_provider.container',

```

```

'@serialization.phpserialize'],
],
'cache.container' => [
  'class' => '\Drupal\redis\Cache\PhpRedis',
  'factory' => ['@cache.backend.redis', 'get'],
  'arguments' => ['container'],
],
'cache_tags_provider.container' => [
  'class' => 'Drupal\redis\Cache\RedisCacheTagsChecksum',
  'arguments' => ['@redis.factory'],
],
'serialization.phpserialize' => [
  'class' => 'Drupal\Component\Serialization\PhpSerialize',
],
],
];
}

```

## 5: Wijzigingen deployen

Pas aan in .gitlab-ci.yml:

```
EXECUTE_INITIAL_COMMANDS: "true"
```

Comitten en deployen:

```

ddev drush cex -y
git add .
git commit -am "Using Redis as caching mechanism"
git push

```

## 6: controleren:

Er is een 'Redis' check bij admin/reports/status

## 7: cache tabellen leeg maken

Op de omgeving waar Redis geactiveerd is kan je de cachtags en cache\_\* tabellen leeg maken.