Redis-Einrichtung

"Der Redis Cache Store ist eine der besten Optionen, um Session- und Anwendungs-Caches zu verwalten, denn der Redis Cache Store bietet Datensicherheit, Locking, Key Awareness. Außerdem kann er für das Caching der Nutzersitzungen in der Moodle-Konfigurationsdatei config.php verwendet werden."

Docs:

https://docs.moodle.org/de/Redis_Cache

https://docs.moodle.org/en/Redis_cache_store#Installing_Redis_server

Redis-Installation und Einrichtung:

Auf **Linux-Server** (z.B. dedizierter Redis-Server oder auf Moodle-Server):
- sudo dnf install redis
- Redis starten: sudo systemctl start redis.service
- Autostart: sudo systemctl enable redis
- sudo sysctl vm.overcommit_memory=1
- config-Anpassung in `/etc/redis.conf` :

> appendonly yes (default: no)
> bind auskommentiert `#bind 127.0.0.1 (außer wenn auf gleichem Server wie Moodle)`
> *protected-mode no* (default: yes)
> *Wichtig: für öffentliche Instanzen ein Passwort nutzen! (requirepass foobar)*
> siehe auch: https://redis.io/topics/security
>
> Speicherlimit auf z.B. 8 GB setzen (je nach Server-Konstellation, Limit wird auch in tool_redis angezeigt) + policy "allkey-lru" als Replacement-Strategie
> (sonst werden ggf. keine neuen Keys mehr angenommen / gespeichert, wenn der Store voll ist)
> *maxmemory 8590000000*
> *„maxmemory_policy“ auf „allkeys-lru“ (maxmemory-policy allkeys-lru)*

- sudo systemctl restart redis.service

PHP-Paket installieren

Auf **Webservern** PHP-Paket installieren:
- sudo yum install php-pecl-redis
- sudo systemctl reload php-fpm
- *php -m* muss "redis" anzeigen

PHP-Info prüfen

Wurde Redis sauber geladen? z.B. `/admin/phpinfo.php`
Es sollte einen Absatz "redis" geben, zudem eine redis.ini unter "Additional .ini files parsed"

Moodle-Frontend

"Testserver" einrichten unter `/admin/settings.php?section=cachestore_redis_settings`
z.B.:
`RedisServer:6379`
wenn Redis auf dem gleichen Server wie Moodle selbst läuft:
127.0.0.1:6379

- Danach ist Redis unter `/cache/testperformance.php` mit Werten gelistet
  neue Redis-Instanz hinzufügen unter `/cache/admin.php` (ebenfalls mit obiger Adresse und Port 6379) und "*Key-Prefix*" definieren, z.B. "mdlred_": "Neue Instanz hinzufügen" (`/cache/admin.php?action=addstore&plugin=redis`)
- gleiche Seite ganz unten "*Speicher-Zuordnungen*" auf "*Bearbeiten*" klicken, für "*Anwendung*" und "*Session*" zuvor konfigurierten RedisCache einstellen
- unter "*Speicher-Zuordnungen*" sollte nun eine zwei- oder dreistellige Zahl stehen, z.B. "*92* " oder "*117*".

Empfehlung: Testserver nach erfolgreichen Tests deaktivieren (Eintragung löschen), um ungewünschte Beeinflussungen bei etwaigen Tests darüber im Betrieb auszuschließen.

Speicherort Konfiguration

Die Redis-**Konfiguration** des jeweiligen Systems wird in **moodledata** (`/vol/moodledata/muc/config.php`) gespeichert, nach DB-Importen von z.B. prod nach test ist also nichts anzupassen:
sudo cat /vol/moodledata/muc/config.php | grep Redis

Deaktivierung der Redis-Einbindung

"Speicherort wenn keine Definition erstellt wurde" (`/cache/admin.php` ganz unten) wieder auf "*Standard*" Speicher setzen
wenn kein Zugriff auf admin-Frontend: ***/vol/moodledata/muc/config.php***
Zeile 69 ändern von "'*store*' => '*Redis-mdl*'," auf "'*store*' => '*default_application*',"

/var/lib/redis: dump.rdb und appendonly.aof (persistence)

Die Datei dump.rdb dient der Speicherung des gesamten Snapshots der Redis-Datenbank ("Point-in-Time" Darstellung der Datenbank). Redis erstellt regelmäßig Snapshots der Datenbank und speichert diese als `dump.rdb`. Diese Schnappschüsse enthalten den Zustand der Datenbank zu dem Zeitpunkt, zu dem der Snapshot erstellt wurde. Im Falle eines Absturzes oder Neustarts des Redis-Servers verwendet Redis die `dump.rdb`-Datei, um den letzten gespeicherten Zustand der Datenbank wiederherzustellen. Redis erstellt die `dump.rdb`-Datei automatisch basierend auf den in der Konfigurationsdatei (`redis.conf`) festgelegten Regeln. Diese Regeln definieren, wie oft und unter welchen Bedingungen Snapshots erstellt werden (z.B. nach einer bestimmten Anzahl von Schreiboperationen oder nach einer bestimmten Zeitspanne). Redis läuft also auch ohne dump.rdb (erstellt Redis selbst) und ohne appendonly.aof in `/var/lib/redis`, sprich mit leerem Verzeichnis (bei sauberem Service-Neustart via sudo *systemctl start redis.service*).

appendonly.aof:

- Speichert jede einzelne Schreiboperation in der Reihenfolge, in der sie auftritt.
- Ermöglicht eine genauere Wiederherstellung der Datenbank, da nahezu keine Daten verloren gehen, ist aber ressourcenintensiver.
- Kann mehrere GB diskspace verbrauchen

Siehe auch https://redis.io/docs/latest/operate/oss_and_stack/management/persistence/ und

https://www.memurai.com/blog/redis-persistence-deep-dive

Autor: Klaus Steitz, Technische Universität Darmstadt

Version #8
Erstellt: 2025-09-09 05:10:32 UTC von Klaus Steitz
Zuletzt aktualisiert: 2026-06-05 10:46:00 UTC von Klaus Steitz