

# WordPress

Fast 40 % des gesamten Internets läuft auf Wordpress...

(und fast 40 % des gesamten Internets hängt hinter Fortigate Firewalls... Falls ihr Hackingziele braucht)

- [manuelles Update](#)
- [uralt-Versionen hochziehen](#)
- [Maintenance Mode](#)
- [bedrock-Wordpress](#)
- [Admins in der Datenbank anlegen](#)

# manuelles Update

- SSH aufschalten
- Im Wordpress-Backend alle Plugins deaktivieren oder:

```
cd wp-content && mv ./plugins ./plugins.disable
```

- Alle Files löschen, außer ./wp\_config.php und ./wp-content
- Neue Wordpress Files hochladen
- ./wp-admin/upgrade.php als URL im Browser aufrufen und durchmachen

Sollte mit allen Versionen gehen. Es entsteht Sprachenmix, wenn ihr andere Sprachen übereinander bügelt.

Inkompatibilität mit PHP 8.1 kann aber auch anderswo her resultieren.

# uralt-Versionen hochziehen

1. Alles außer WP-config.php und .\wp-content via SSH löschen
2. .\wp-content\plugins umbenennen
3. Neue Wordpress Files hochladen ohne Dateiüberschreibung
4. .\login.php als TCP 443 aufrufen und dann wird man entweder vor oder nach Login auf die Aktualisierungs-Seite weitergeleitet.

# Maintenance Mode

Maintenance Mode schaltet sich beim Plugin und Theme updaten an.

Falls was fehlschlägt sollte er nach 10 Minuten automatisch wieder ausgehen.

# bedrock-WordPress

WordPress mit [Bedrock](#) auf [Composer](#) geworfen mag vielleicht besser sein, ist aber realistisch gesehen vielleicht nicht ganz so geeignet.

Weil:

Ich sehe es als wichtiger an, Wordpress und Plugins stets aktuell zu halten – weil Frontend und so. Das funktioniert mit Bedrock halt nur auf der Shell, und wer wählt sich alleine deswegen schon auf die Shell ein?

Im Endeffekt wirds also so sein, dass die meisten Bedrock-WordPress-Instanzen diejenigen sind, die eher veraltet sind.

Das ist kritischer als der Vorteil den Bedrock eventuell bringt.

Bedrock lagert eben vieles in Richtung composer aus. Sprich: vieles geht zurück auf Shell-Ebene (vielleicht auch zum Glück). Das machts erforderlich dann dort mit Scripts oder Cronjobs zu arbeiten, will man es halbwegs aktuell halten, oder das Wordpress-Auto-Update für Plugins kopieren.

Aber bevor man an [WP-CLI](#) denkt, vielleicht doch eher an Bedrock denken?

# Admins in der Datenbank anlegen

1. phpMyAdmin öffnen

## Tabelle wp\_users

1. User in Tabelle wp\_users anlegen. [Passworthasher hier](#)
2. user\_id aufsteigend
3. user\_email füllen, damit zur Not Passwortreset funktioniert

## Tabelle wp\_usermeta

1. Neuer Eintrag
2. gleiche user\_id angeben
3. meta\_value

```
a:1:{s:13:"administrator";s:1:"1";}
```

4. meta\_key: wp\_user\_level
5. meta\_value: 10

Wenn jetzt irgendwas am Hash nicht stimmt ist ein Passwortreset mit der Mail-Adresse möglich. Nach der ersten Anmeldung im Backend einmal in die Benutzeroptionen für sich selbst öffnen und speichern.

Wenn was grundlegendes falsch ist und sich jetzt gar keiner mehr anmelden kann, einfach die zwei erstellten Tabellenzeilen wieder löschen und gut ist...