Επιλογή και αλλαγή PHP version
Τι πρέπει να γνωρίζετε

Compatibility, έλεγχος πριν και μετά, extensions, staging και rollback - intraweb.host

Η PHP είναι βασικό μέρος της λειτουργίας πολλών ιστοσελίδων και εφαρμογών. Η αλλαγή έκδοσης μπορεί να βελτιώσει ασφάλεια, συμβατότητα και απόδοση, αλλά μια εφαρμογή, ένα plugin ή ένα theme που δεν υποστηρίζει τη νέα έκδοση μπορεί να παρουσιάσει warnings, errors ή πλήρη διακοπή λειτουργίας.

Η νεότερη διαθέσιμη PHP δεν είναι αυτομάτως η σωστή επιλογή για κάθε site. Η σωστή έκδοση είναι μία ενεργά υποστηριζόμενη έκδοση που υποστηρίζεται και από την εφαρμογή και από τα κρίσιμα components της.

1. Γιατί έχει σημασία η PHP version

• Κάθε έκδοση PHP έχει συγκεκριμένο κύκλο υποστήριξης και λαμβάνει διορθώσεις για περιορισμένο χρονικό διάστημα.

• Παλαιότερος κώδικας μπορεί να χρησιμοποιεί λειτουργίες που έχουν deprecated ή αφαιρεθεί σε νεότερες εκδόσεις.

• Νεότερες εφαρμογές και plugins μπορεί αντίθετα να απαιτούν ελάχιστη PHP version για να εγκατασταθούν ή να λειτουργήσουν.

2. Μην επιλέγετε έκδοση μόνο επειδή είναι η νεότερη

• Ελέγξτε πρώτα τις απαιτήσεις του CMS ή της εφαρμογής.

• Ελέγξτε τα κρίσιμα plugins, themes, modules και integrations.

• Σε custom ή παλιές εφαρμογές, η συμβατότητα πρέπει να επιβεβαιώνεται πριν από αλλαγή σε production.

• Στόχος είναι η νεότερη κατάλληλη και υποστηριζόμενη έκδοση, όχι απλώς ο μεγαλύτερος αριθμός.

3. Πριν από την αλλαγή

• Διατηρήστε πρόσφατο backup ή κατάλληλο restore point.

• Καταγράψτε την τρέχουσα PHP version ώστε να γνωρίζετε σε ποια έκδοση θα επιστρέψετε αν χρειαστεί.

• Ελέγξτε αν υπάρχουν ήδη PHP/application errors πριν την αλλαγή.

• Για σημαντικά sites, προτιμήστε πρώτα δοκιμή σε staging ή clone.

4. WordPress / WooCommerce

• Ελέγξτε τις απαιτήσεις της εγκατεστημένης έκδοσης WordPress και των σημαντικών plugins/themes.

• Ενημερώστε εγκαίρως παλιά components που εμποδίζουν τη μετάβαση σε υποστηριζόμενη PHP.

• Σε WooCommerce ελέγξτε checkout, payment gateway, emails, scheduled/background actions και admin λειτουργίες μετά την αλλαγή.

• Το ότι ανοίγει η homepage δεν αποδεικνύει πλήρη συμβατότητα.

5. Άλλα CMS και custom εφαρμογές

• Joomla, Drupal, frameworks και custom PHP εφαρμογές έχουν δικές τους απαιτήσεις ανά έκδοση.

• Παλιές εφαρμογές μπορεί να εξαρτώνται από deprecated functions ή παλιά libraries.

• Encoded/commercial components μπορεί να απαιτούν συμβατό loader ή συγκεκριμένες εκδόσεις PHP.

• Μην αλλάζετε PHP σε legacy εφαρμογή χωρίς να υπάρχει σχέδιο επιστροφής.

6. PHP extensions

• Μία εφαρμογή μπορεί να χρειάζεται extensions όπως mbstring, intl, zip, gd, imagick, ionCube ή άλλα.

• Μετά από αλλαγή PHP version, επιβεβαιώστε ότι τα απαραίτητα extensions είναι διαθέσιμα για τη νέα έκδοση.

• Missing extension μπορεί να εμφανιστεί ως application error παρότι η ίδια η PHP λειτουργεί.

• Μην ενεργοποιείτε άγνωστα extensions χωρίς πραγματική απαίτηση της εφαρμογής.

7. PHP configuration και limits

• Ρυθμίσεις όπως memory_limit, max_execution_time, upload_max_filesize και post_max_size μπορεί να διαφέρουν ανά περιβάλλον/έκδοση.

• Ελέγξτε αν η εφαρμογή έχει ειδικές απαιτήσεις.

• Μην χρησιμοποιείτε υπερβολικά limits ως μόνιμη λύση για inefficient code ή βαριές εργασίες.

• Καταγράψτε custom PHP settings πριν από σημαντική αλλαγή.

8. Τι να ελέγξετε αμέσως μετά

• Αρχική σελίδα και βασικές εσωτερικές σελίδες.

• Login και διαχειριστικό περιβάλλον.

• Forms, search, uploads και άλλες δυναμικές λειτουργίες.

• Σε e-shop: cart, checkout και payment flow με ασφαλή δοκιμή.

• PHP/application logs για νέα warnings, deprecated notices ή fatal errors.

9. «Η σελίδα ανοίγει» δεν αρκεί

• Ένα site μπορεί να εμφανίζει σωστά cached/public pages αλλά να αποτυγχάνει σε admin, cron ή background processes.

• Μπορεί να λειτουργεί το frontend αλλά να αποτυγχάνει η αποστολή email, ένα API ή συγκεκριμένο plugin.

• Ελέγξτε αντιπροσωπευτικές λειτουργίες και όχι μόνο την homepage.

• Παρακολουθήστε τα logs και μετά την αρχική δοκιμή.

10. Αν εμφανιστεί error

• Καταγράψτε το ακριβές μήνυμα και την ώρα.

• Ελέγξτε αν το log αναφέρει συγκεκριμένο plugin, theme, module ή custom file.

• Αν το site λειτουργούσε αμέσως πριν την αλλαγή, η επιστροφή στην προηγούμενη PHP version μπορεί να είναι το ασφαλέστερο προσωρινό rollback.

• Στη συνέχεια διορθώστε ή αναβαθμίστε το ασύμβατο component πριν επιχειρήσετε ξανά.

11. Πότε χρειάζεται rollback

• Όταν εμφανίζεται fatal/critical error που επηρεάζει ουσιαστική λειτουργία.

• Όταν checkout, login, admin ή κρίσιμη integration σταματήσει να λειτουργεί.

• Όταν δεν μπορεί να διορθωθεί άμεσα το ασύμβατο component.

• Το rollback είναι προσωρινή αποκατάσταση, όχι λόγος να παραμείνει επ' αόριστον μια μη υποστηριζόμενη PHP.

12. Παλιές / EOL εκδόσεις PHP

• Μία εφαρμογή που απαιτεί παλιά PHP χρειάζεται σχέδιο αναβάθμισης, όχι μόνιμη αναβολή.

• Η μακροχρόνια χρήση μη υποστηριζόμενης έκδοσης αυξάνει τεχνικό και security risk.

• Εξετάστε update της εφαρμογής, αντικατάσταση ασύμβατων plugins/modules ή migration σε νεότερη πλατφόρμα.

• Μην περιμένετε μέχρι η παλιά έκδοση να γίνει το μοναδικό εμπόδιο σε hosting ή application upgrade.

13. Καλή διαδικασία αλλαγής

• Backup → καταγραφή υπάρχουσας κατάστασης → compatibility check → staging/test → αλλαγή → λειτουργικός έλεγχος → logs.

• Κάντε μία σημαντική αλλαγή κάθε φορά ώστε να είναι σαφές τι προκάλεσε τυχόν πρόβλημα.

• Κρατήστε σημείωση της τελικής PHP version και τυχόν ειδικών settings.

• Σε κρίσιμο production site, προγραμματίστε την αλλαγή σε ώρα που επιτρέπει άμεσο έλεγχο.

Πριν και μετά την αλλαγή – σύντομο checklist

Πριν	Μετά
Πρόσφατο backup / restore point	Homepage και βασικές σελίδες
Καταγραφή τρέχουσας PHP version	Login / admin
Compatibility CMS/plugins/themes	Forms / uploads / dynamic functions
Απαραίτητα PHP extensions	Checkout / integrations όπου υπάρχουν
Custom PHP settings	PHP/application logs
Staging test όπου χρειάζεται	Rollback αν υπάρχει σοβαρή ασυμβατότητα

Κανόνας που αξίζει να θυμάστε: δεν κρατάμε μια παλιά PHP επειδή «το site ακόμη ανοίγει», αλλά ούτε αλλάζουμε βιαστικά σε νέα έκδοση χωρίς compatibility check. Η σωστή πορεία είναι να διατηρούμε την εφαρμογή αρκετά ενημερωμένη ώστε να μπορεί να λειτουργεί σε σύγχρονη, υποστηριζόμενη PHP.