

OPERATIONS TEMPLATE

WordPress Maintenance and Security Runbook

The monthly run, the security baseline, and the incident procedure, written so someone else can execute them at 2am.

Maintenance is only worth paying for if it is repeatable and evidenced. This runbook covers the monthly update pass, a security baseline you can audit against, a backup restore test most sites have never actually run, and an incident procedure for the day something does go wrong.

WHAT'S INSIDE

- | | |
|---------------------------------|-------------------------------------|
| 01 How to use this runbook | 05 Monitoring and thresholds |
| 02 The monthly maintenance run | 06 Incident response |
| 03 The security baseline | 07 The client-facing monthly report |
| 04 Backups and the restore test | |

How to use this runbook

A maintenance plan is a promise about what happens on a bad day. This is the document that makes the promise checkable.

- 1 **Staging first, always.** Updating production directly saves ten minutes and eventually costs a day.
- 2 **Test the restore, not the backup.** A backup you have never restored is a hypothesis. Section 4 turns it into a fact.
- 3 **Log every run.** The log is what a client is actually buying, and it is what makes a retainer renewable.
- 4 **Set thresholds before you need them.** An alert with no agreed threshold gets muted within a fortnight.
- 5 **Keep the incident procedure short.** Nobody reads a twelve-page plan while a site is down.

SITE	RUN CADENCE	OWNER
	Monthly, plus critical releases	

SECTION 1

The monthly maintenance run

About 20 minutes per site once the environment is set up.

- 1 Confirm a backup completed within the last 24 hours. If not, run one and wait for it to finish before doing anything else.
- 2 Pull production to staging so you are testing against real data.
- 3 Update core, then plugins, then theme, in that order.
- 4 Update anything handling payments, forms, or security one at a time, checking between each.
- 5 On staging: load the homepage, one interior page, and one template of each type.
- 6 Submit a form and confirm the notification arrives.
- 7 For ecommerce, add to cart and reach the payment step without completing it.
- 8 Check the error log for anything new since the last run.
- 9 Deploy to production and repeat the same checks within 10 minutes.
- 10 Confirm the uptime monitor shows no incident, then record the run in the log.

Run log

DATE	CORE	PLUGINS	ISSUES FOUND	DEPLOYED BY

WHEN NOT TO UPDATE

Never run a routine update during a client's peak trading window, on a Friday, or while a campaign is mid-flight. A postponed update is a decision; a broken checkout on Black Friday is an incident.

The security baseline

Audit against this quarterly. Anything unticked is either fixed or written down as an accepted risk with a reason.

- ☐ No shared logins: every human has a named account
- ☐ Two-factor authentication on all admin accounts
- ☐ Strong password policy enforced, not just advised
- ☐ File editing disabled in the dashboard
- ☐ Automatic security updates enabled for core
- ☐ No plugin abandoned by its author still in use
- ☐ HSTS and baseline security headers set
- ☐ XML-RPC disabled or restricted if unused
- ☐ Database prefix not the default
- ☐ Malware and file-integrity scanning scheduled
- ☐ Administrator accounts reviewed and ex-staff removed
- ☐ Default 'admin' username absent
- ☐ Login attempts rate limited
- ☐ PHP version supported and receiving security patches
- ☐ Unused plugins and themes deleted, not just deactivated
- ☐ SSL valid, with auto-renewal confirmed
- ☐ Directory listing disabled
- ☐ REST API user enumeration restricted
- ☐ Backups stored off the same server
- ☐ Hosting firewall or WAF active

Accepted risks

ITEM NOT MET	WHY	REVIEWED ON

Backups and the restore test

Backup configuration

QUESTION	ANSWER
Frequency of file backups	
Frequency of database backups	
Retention period	
Stored where, off-server	
Who can trigger a restore	
Recovery point objective	
Recovery time objective	

The restore test

- 1 Take the most recent backup and restore it to a scratch environment.
- 2 Time the restore end to end and record it. That number is your real recovery time, not the one in the proposal.
- 3 Confirm the restored site loads, the database is intact, and recent content is present.
- 4 Confirm uploads and media restored, not just the database.
- 5 Record the result below and destroy the scratch environment.

TESTED ON	BACKUP AGE	RESTORE TIME	RESULT	BY

RUN THIS TWICE A YEAR, MINIMUM

Most sites have never had a restore tested. The failure mode is discovering during a real incident that the backup excluded uploads, or that nobody has the credentials to the backup store.

Monitoring and thresholds

WHAT	THRESHOLD	ALERTS WHOM	CHANNEL
Uptime	<u>2</u> failed checks		
Response time	over <u>3</u> s sustained		
SSL expiry	<u>21</u> days out		
Domain expiry	<u>60</u> days out		
Disk usage	over <u>85</u> percent		
PHP or 500 errors	any spike		
Failed logins	over <u>20</u> per hour		
File integrity change	any in core		
Backup failure	any		

AN ALERT NOBODY OWNS GETS MUTED

Every row needs a person, not a shared inbox. The fastest way to make monitoring useless is to route it somewhere that generates noise nobody is accountable for.

Incident response

Short on purpose. Nobody reads a long plan while a site is down.

Severity

LEVEL	MEANS	RESPONSE STARTS	CLIENT TOLD
1	Site down, or checkout failing	Immediately	Within 30 min
2	Major feature broken, site up	Same business day	Same day
3	Minor defect, workaround exists	Next run	In monthly report

First 30 minutes

- 1 Confirm the problem is real and reproduce it from outside your own network.
- 2 Check whether anything changed: a deploy, an update, a DNS or plugin change.
- 3 If a change caused it, revert that change first. Do not debug forward under pressure.
- 4 If the cause is unknown and severity is 1, restore from the last known good backup.
- 5 Tell the client what is happening before they ask, even without a cause yet.
- 6 Preserve the logs before they rotate.

If it is a compromise

- ☐ Take the site offline or into maintenance mode rather than leaving it serving malware
- ☐ Rotate every credential: hosting, database, admin users, API keys, SFTP
- ☐ Restore from a backup predating the compromise, not the most recent one
- ☐ Patch the entry point before restoring, or it happens again within days
- ☐ Scan for injected users, scheduled tasks, and modified core files
- ☐ Check whether personal data was exposed, and take advice on notification duties
- ☐ Write the timeline down the same day, while it is still accurate

SECTION 6

The client-facing monthly report

This page is what the retainer is actually selling. Send it every month, whether or not anything interesting happened.

SITE	PERIOD	PREPARED BY

AREA	THIS MONTH	NOTES
Uptime		
Core, plugin, theme updates applied		
Backups taken and verified		
Security events blocked		
Average load time		
Issues found and fixed		
Hours used against allocation		

What we recommend next

ONE OR TWO ITEMS, WITH A REASON AND A ROUGH COST

ALWAYS RECOMMEND SOMETHING

A report that says everything is fine every month trains the client to stop reading it, and eventually to question the invoice. There is always a next improvement worth naming, even if the recommendation is to do nothing yet and why.

We build the systems behind the paperwork

Gatilab is an independent WordPress and search agency. We build sites, rank them, and stay long enough to prove the work moved something.

2014 Founded	850+ Clients served	2,000+ Projects completed	10,000+ Plugin installs
------------------------	-------------------------------	-------------------------------------	-----------------------------------

What we do

WordPress engineering

Custom themes, blocks, plugins, and migrations built to stay fast and maintainable after handoff.

SEO strategy

Technical fixes, topical coverage, and search visibility work measured against clicks, not scores.

Content marketing

Research-led articles and editorial systems your team can keep publishing without us.

Conversion design

Landing pages, service pages, and forms designed around the decision the reader is actually making.

Need this handled instead of templated?

Send the problem, not a brief. We will tell you whether it is worth building, what it depends on, and what we would deliberately skip. If the honest answer is that you do not need us, you will get that too.

[Start a project](#)

gatilab.com/connect

USE IT FREELY

Free to use, edit, and adapt for your own business and your own clients. No attribution required, no sign-up. The one thing we ask: do not resell it as your own product.