Phase 1: Lock down the infected site

1. **Put the current site in maintenance / offline mode**
    - If possible, block public access at server or firewall level.
    - At minimum, use .htaccess to restrict access by IP while you work.
2. **Disable all logins**
    - Change passwords for:
        - cPanel or EC2 / SSH user accounts
        - WordPress admin users
        - Database users
        - FTP / SFTP users
3. **Take a full backup of the infected server**
    - Even if it's infected, you need a snapshot for reference and for pulling content:
        - Whole web root (e.g. /var/www/html or /public_html)
        - Database (via mysqldump or phpMyAdmin export)
    - Store this backup **offline** (NOT on the new server) for analysis only.

Phase 2: Prepare a brand new clean server

Since you said you will host on a new server that's clean, start here:

1. **Set up a fresh environment**
    - New EC2 instance / VPS with:
        - Latest stable PHP version supported by WordPress
        - Latest MySQL / MariaDB
        - Web server: Apache or Nginx
2. **Install WordPress fresh**
    - Download *only* from the official site (wordpress.org).
    - Create a **new** database and database user with a strong password.
    - Complete the fresh install so you have a clean wp-core + wp-config.php.
3. **Harden the base install**
    - Change the default database table prefix (not wp_).
    - Use strong, unique passwords for:
        - WordPress admin

        - Database user
        - System users
    - Ensure file permissions are sensible:
        - Folders: 755
        - Files: 644
        - No 777 anywhere.

Phase 3: Move *only clean data* from old to new

The golden rule:

You only migrate content and trusted assets, never code from the infected server.

3.1 Export and import database content safely

1. **Export the old database**
    - Use mysqldump or phpMyAdmin: export as .sql file.
2. **Inspect the SQL file quickly**
    - Open it in a text editor.
    - Look for obviously injected junk in:
        - wp_options table (especially siteurl, home, active_plugins, suspicious base64 strings).
        - wp_posts (hidden iframes, <script> tags, strange encoded blocks).
    - If you see very obvious malware payload blocks, remove them before import if you are comfortable. If not, you can still clean within WordPress later using security plugins.
3. **Import content into the new database**
    - Option 1 (cleanest):
        - Use WP's **Tools → Export** on the old site (Posts, Pages, Media) then **Tools → Import** on the new site. This only brings content, not plugins/themes/options.
    - Option 2 (if export tool is broken):
        - Import the SQL into the new DB, then be ready to **scan and clean via security plugin** and possibly manually.

If you can, **prefer Option 1 (Export/Import)**. It minimizes risk.

3.2 Reinstall themes and plugins from trusted sources

This is crucial.

1. **On the new server, delete default wp-content contents (except index.php)**
    - Keep the structure, but **do not copy wp-content from the infected site**.
2. **On the new server, reinstall:**
    - The theme:
        - Download from original source (ThemeForest, developer site, or WordPress theme repo).
        - If it's a custom theme, get it from your original clean copy / developer, not from the hacked server.
    - All plugins:
        - Install one by one from the official WordPress repo or original vendor.
        - Do not upload plugins from the infected server.
3. **Only copy safe assets from the old server**
    - From old wp-content/uploads, copy **only media files**:
        - Images: .jpg, .jpeg, .png, .gif, .webp
        - PDFs or docs **if you trust them**
    - Do **not** copy .php, .js, .html from uploads.
        - Malware often hides in random .php files inside uploads (red flag).
4. **If you had custom code**
    - Ask your developer for the latest clean version, or manually review each file before copying.
    - Never blindly copy custom .php from the infected server.

Phase 4: Clean and check the new WordPress install

Now we assume:

- Core is fresh
- Themes/plugins are reinstalled from trusted sources
- Content is imported

Now you deep-scan the *new* site so you know it is clean.

1. **Install a reputable security plugin**
    - Examples: Wordfence, iThemes Security, or similar.
    - Run a full scan:

        - It should verify core files against originals.
        - It should flag modified or suspicious files.

2. **Fix what the scan finds**
    - For each flagged file:
        - If it is a WordPress core file: restore from official core.
        - If it is a plugin/theme file: delete and reinstall that plugin/theme from the original source.
        - If it is in uploads or custom folders: open it:
            - If you see base64_decode, eval, gzuncompress, shell_exec, strange long encoded strings, etc, and it is not expected: delete the file.
3. **Check user accounts**
    - Go to Users in WordPress admin:
        - Delete any suspicious users, especially admins you did not create.
        - Change passwords for all admins.
    - Make sure there is only **one** real Administrator role (max 2 if truly needed).
4. **Check critical options**
    - In Settings → General:
        - WordPress Address (URL) and Site Address (URL) must be correct.
    - In your security plugin or wp_options, check:
        - There are no suspicious auto-load options with encoded code.

Phase 5: Switch domain to the new server

Once you are confident the **new server is clean**, then:

1. **Update DNS**
    - Point the domain's A record to the new server's IP.
    - Lower DNS TTL beforehand if you want faster propagation.
2. **Set up HTTPS**
    - Install an SSL certificate (Let's Encrypt or hosting provider).
    - Enforce HTTPS via .htaccess or web server config.
3. **Monitor logs closely for the first week**

- Web server access logs: see if there are repeated hits to old known backdoor paths (e.g., wp-content/uploads/somefile.php).
- If you see 404s to strange old paths, that is actually good: the files are gone.

Phase 6: Hardening so it doesn't happen again

To reduce the chance of future backdoors:

1. **Keep everything updated**
    - Enable automatic minor updates for WordPress core.
    - Regularly update:
        - Themes
        - Plugins
        - PHP version
2. **Remove what you don't use**
    - Delete inactive themes and plugins entirely.
    - Fewer components means fewer vulnerabilities.
3. **Limit write access and file editing**
    - In wp-config.php add:
    - define('DISALLOW_FILE_EDIT', true);
    - Ensure correct file permissions:
        - No world-writable directories or files.
    - Restrict direct PHP execution in uploads and similar folders by adding an .htaccess like:
    - <Files *.php>
    - Deny from all
    - </Files>

(Or equivalent rules for Nginx.)

4. **Use a Web Application Firewall**
    - If BitNinja or another WAF is configured, keep it active on the **new** server.
    - Configure rules to:
        - Block known exploit patterns
        - Limit login attempts
5. **Secure logins**

- Enforce strong passwords using a password manager.
- Implement 2FA (two factor auth) on all admin accounts.
- Consider changing /wp-login.php URL using a security plugin and limit login by IP if possible.

6. **Regular backups**
    - Automated daily backups of:
        - Files
        - Database
    - Store them off server (S3, remote storage, etc).
    - Test restoring occasionally so you know backups actually work.

Quick summary checklist for you

You can literally tick these off:

1. Lock down old server, change all passwords, take offline backup
2. Deploy brand new server with clean WordPress install
3. Migrate content via WP Export/Import or carefully cleaned SQL
4. Reinstall all themes/plugins from official sources
5. Copy only media files, no PHP from old server
6. Run security scan on new site and clean anything flagged
7. Verify users, options, URLs, and file permissions
8. Point domain to new server and ensure HTTPS
9. Harden security: WAF, 2FA, disable file editing, regular updates
10. Set up offsite, automated backups and monitoring