SEV 1 Party Migration Plan

Re-architect of SEV 1 Party from Amazon Lightsail to Headless WordPress

Accountable: Molly Sheets **Date**: Aug 10, 2025

Executive Summary

SEV 1 Party (https://mollysheets.com) is currently hosted on Amazon Lightsail using a Bitnami image for WordPress on a single host. While it has daily snapshots and a separate warm standby for disaster recovery, this model's simplicity-by-design presents challenges. Downtime is < 30 seconds during maintenance; however, the architecture does not support upgrading core software with 0 downtime for end-users without adding additional components, is not horizontally scalable today, and cannot grow with its audience.

Additionally, current architecture, due to coupling the CMS (Content Management System), with the frontend user-facing website (the web pages the public sees) is no longer a forward-thinking security model. The site presents performance and security risks that if rearchitected could be avoided with minimal cost increase.

Moving to a headless WordPress model, where the CMS is hosted either locally or on a separate instance and the frontend is hosted in Amazon S3 as static pages, not only makes upgrades and management of the site safer and easier, it makes the frontend scalable in the event of a DDoS (distributed denial of service), enables the architecture to have robust deployment that avoids downtime, and ensures the system is more secure.

Full Timeline	**Cost/Resource**
3 Months	10 Days

Project Requirements

- Migrate SEV 1 Party from Amazon Lightsail to a static WordPress model
- Identify, remove, and replace plugins that no longer make sense in this model
- Investigate non-local hosting for the CMS and database backup
- Setup WordPress with GitHub and Simply Static CI/CD to deploy to Amazon S3
- Practice upgrades of WordPress Core, PHP, Apache, MySQL in new model with 0 downtime
- After 3 months of maintaining new architecture, decommission Amazon Lightsail

Out of Scope

- CDN & caching improvements - This is a future desire and nice to have
- Web Redesign - Assumes no PHP conflicts with existing theme. If encountered, re-estimate required.

Executive Sign-Off

I agree to implement this in the name of performance & security.

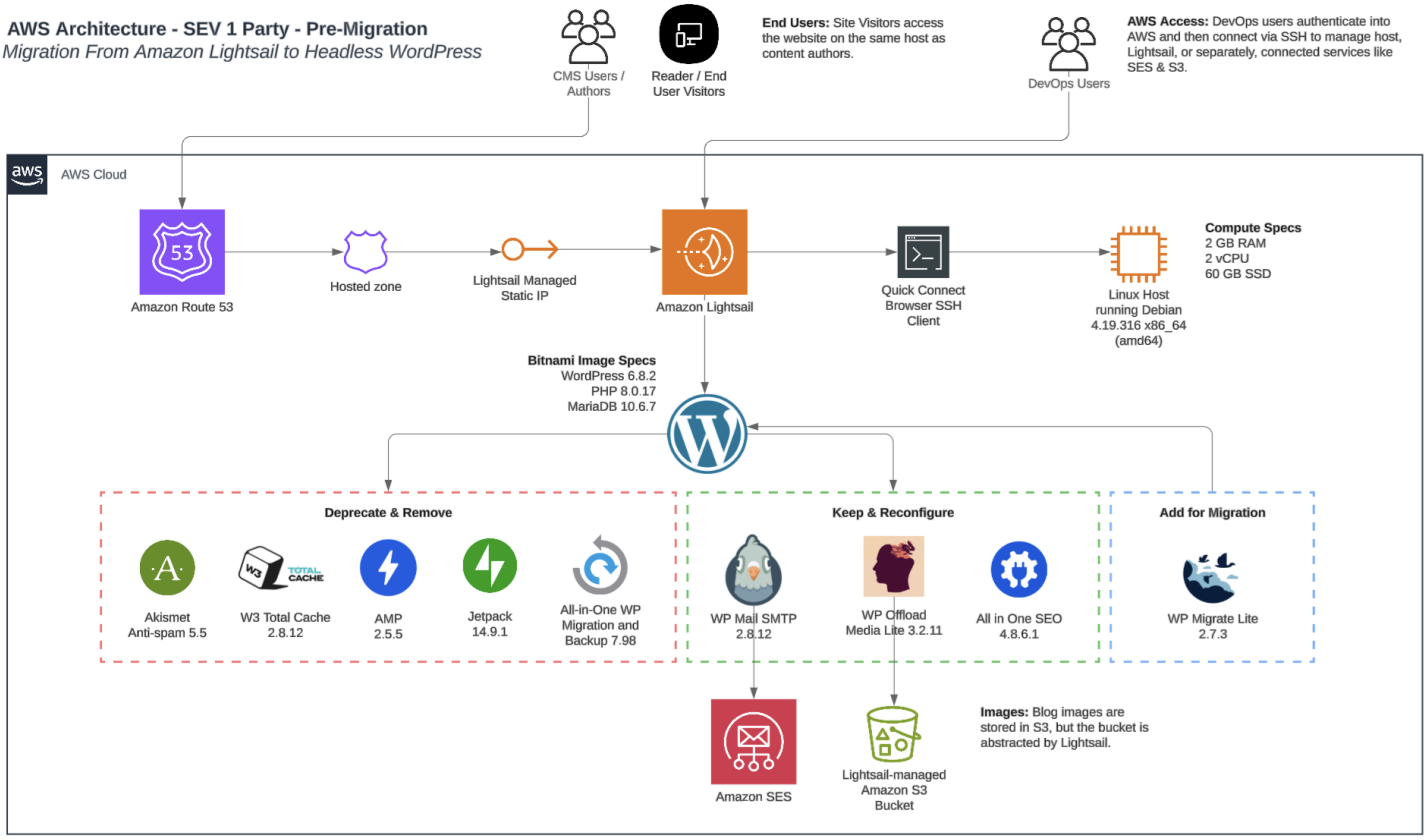
Copyright Molly Sheets © 2025. All Rights Reserved.

Pre-Migration Assessment: Current State & History

SEV 1 Party was created in 2022 as a move from Medium to self-hosted WordPress. Amazon Lightsail presented a great managed service solution in AWS then as it was easy to get started if you were an existing Amazon Web Services user and provided AMI (Amazon Machine Images) for a Bitnami-managed WordPress coupled with cheap compute out of the box. These were provided as “Linux/Unix” blueprints that also supported IPv4 and IPv6 dual stack networking. While this was great, and WordPress core and plugin installs were easy, as PHP and MariaDB needed updates, it became clear that updating other parts of the site were not. Additionally, while this author is okay with 30 seconds of downtime due to SEV 1 Party’s audience being small and the website not revenue-generating, downtime is not ideal if the website were to grow. Specifically,

(1) Upgrades: When needing to upgrade PHP, Apache, and MariaDB the website must migrate from one Lightsail instance to a new host with a fresh WordPress install. Plugin and core updates are done on host from a pre-existing image, with a warm standby on older versions should an update fail. In this current model, 0 downtime requires more infrastructure for load balancing, more complex Route53 DNS setup, routine changes to wpconfig.php and overall does not increase security, while increasing costs.

(2) High Level Coupling: The CMS (Content Management System) is coupled on the same host as the frontend, user-facing website. This means if someone were to DDoS (Distributed Denial of Service) SEV 1 Party, not only would it not scale to absorb the attack, once it is brought down, the CMS to manage the website would be brought down with it, making recovery painful and long. The frontend should operate independently of the CMS, decoupled.



Copyright Molly Sheets © 2025. All Rights Reserved.

Copyright Molly Sheets © 2025. All Rights Reserved.

The current architecture runs on a 2 GB RAM, 2 vCPUs, 60 GB SSD host in us-west-2a (Oregon) complete with automatic daily snapshots and manual snapshots during the upgrades process. However, there are more aspects to this architecture to support it today. For example, the architecture is also supported by a series of plugins to handle media, page caching, analytics, and emails.

Plugins Inventory: The full plugin inventory for the Amazon Lightsail deployment is documented below.

Plugin Name	Use Case	Keep?	Suggested Replacement
Akismet Anti-spam: Spam Protection	Spam protection and management for comments.	No	The website does not support comments today and has no plans to in the immediate future.
All in One SEO	SEO plugin for WordPress - includes XML Sitemaps, SEO features for blogs.	Yes	Compatible with Simply Static
All-in-One WP Migration and Backup	Pre-installed with Lightsail migration plugin for moving entire WordPress site including database, media, plugins, and themes.	No	Not being used for this migration or future. Will be using WP Migrate Lite.
AMP	Provides optimizations for pages.	No	N/A
Jetpack	Security, performance and marketing tools. Currently used for uptime tracking and analytics (“Jetpack Stats”). Jetpack also blocks brute force attacks by blocking malicious IPs and bots.	No - Replace	Replace with new uptime and analytics solutions. Decoupling prevents Jetpack Stats from interacting cohesively. Brute force login attacks hardened in new architecture.
W3 Total Cache	Caching solution for pages, objects, browser, and databases. Also provides minify features, reverse proxy, and Bunny CDN. Used by SEV 1 Party for page caching, browser caching, minify and caching php code via Opcode.	No	Some of these features can be replaced by Simply Static as well as CloudFront CDN.
WP Mail SMTP	Send mail management integrated with Amazon SES.	Yes	Compatible with Simply Static, but will require reconfiguration.
WP Offload Media Lite	Used to copy media uploads to be stored in S3 instead of on disk.	Yes - Rework	Will require rework - see “Offloaded Images”

Offloaded Images: It is important to note that there is an Amazon S3 bucket abstracted by Amazon Lightsail and managed in Amazon Lightsail as part of the existing architecture that was created using this method in Lightsail instead of the Amazon S3 method. This bucket is used to offload uploaded images, connected to the WP Offload Media Lite plugin, so they are not stored on the 60 GB SSD disk. Unlike buckets created in Amazon S3 itself, it has a 5 GB storage limit. The abstraction is significant as the bucket did not originally appear directly accessible in Amazon S3 via the console. Previously uploaded images to the Lightsail bucket, while their URLs will remain the same when migrated, will need a new place to be stored in a new S3 bucket that can be accessed without abstraction and old images will need to be moved to that bucket before Amazon Lightsail is decommissioned completely.

Copyright Molly Sheets © 2025. All Rights Reserved.

However testing revealed that when specifying the bucket directly via command line using `aws s3 ls s3://bucket-name/` the contents are not only able to be listed, but also downloaded via command line meaning they can be migrated without a scraping script. This means that while the bucket is abstracted in Lightsail, and even though with `aws s3 ls` the bucket does not show in the main Amazon S3 list of buckets because of the Lightsail abstraction, when the migrator knows the specific bucket name, the migrator is able to bypass the abstraction instead of using `aws lightsail` cli functions, use `aws s3` functions.

The following command can be used to create a temp directory for downloading images from the Lightsail bucket to move to a non-Lightsail managed bucket.

```
mkdir ~/Desktop/bucket-2qnbwu-download && aws s3 sync s3://bucket-2qnbwu
~/Desktop/bucket-2qnbwu-download/ && cd ~/Desktop && zip -r
bucket-2qnbwu-complete-backup-$(date +%Y%m%d).zip bucket-2qnbwu-download/
```

Search: Included WordPress search functionality will no longer work out of the box as it is inherently dependent on the database and queries that database matching the string provided in the search criteria against post titles, excerpts, tags, and blog body text. While this is not a plugin today in the pre-migration architecture, this will need a new setup in the new model. Categories are not affected as they do not use database queries to display.

Static IP: There is also a static IPv4 IP, those familiar with AWS would recognize this as an Elastic IP, managed via Amazon Lightsail that is, similar to the S3 bucket, not accessible outside of Lightsail to attach. This is an easier situation to remedy and would become irrelevant anyway in the new model so it would only require deletion once ready to decommission Lightsail.

General Networking: Other aspects to consider are that with Amazon Lightsail in this current architecture, users do not have to worry about multiple networking aspects of AWS that one would with self-hosting, including, but not limited to setting up their own VPCs (Virtual Private Clouds), routing tables, security groups, and network access control lists (NACLs). When moving to an updated architecture, it would be preferred to keep the networking simple in AWS so that hosting complexity is still low in this regard.

TLS Certificate Management: Certificate management, which is required for https connections, is handled via the bncert-tool which is pre-installed as part of the Bitnami stack. With configuration, it creates a cron job (in this case for every 90 days) that renews TLS certificates orchestrated with Let's Encrypt. While the local version of the website, on a developer machine, for preview in Local WP will use a locally created TLS certificates to avoid trust issues in the developer's browser; the end user website hosted in Amazon S3 will need to have a different solution for certificate management to allow secure connections via https.

The current configuration supports modern protocols like TLS 1.2 and TLS 1.3 and is considered a "Grade A" (strong commercial security) against the Qualys SSL Labs test. While this is acceptable, the reason it falls short of A+ is because it does not have CAA configuration, certification authority authorization, which is required since 2017 by the CA/Browser Forum as a baseline for issuing certificates. CAA allows those who manage a domain to allowlist certificate authorities that can create certificates for their hostnames. In the new model, for the public hosted pages it would be ideal to have a certificate solution that meets this baseline.

Copyright Molly Sheets © 2025. All Rights Reserved.

Post Migration Architecture & Content Strategy

The post migration architecture keeps the existing spirit of “simplicity by design” so that it is easy to maintain, while giving the overall website more room to grow, more security, and better best practices for continuous integration and continuous deployment. The new architecture is a robust foundation to build new features and capabilities upon as well for testing, monitoring, and analytics by moving these aspects to separate systems.

There are 4 key differentiators that profoundly change the way the website operates.

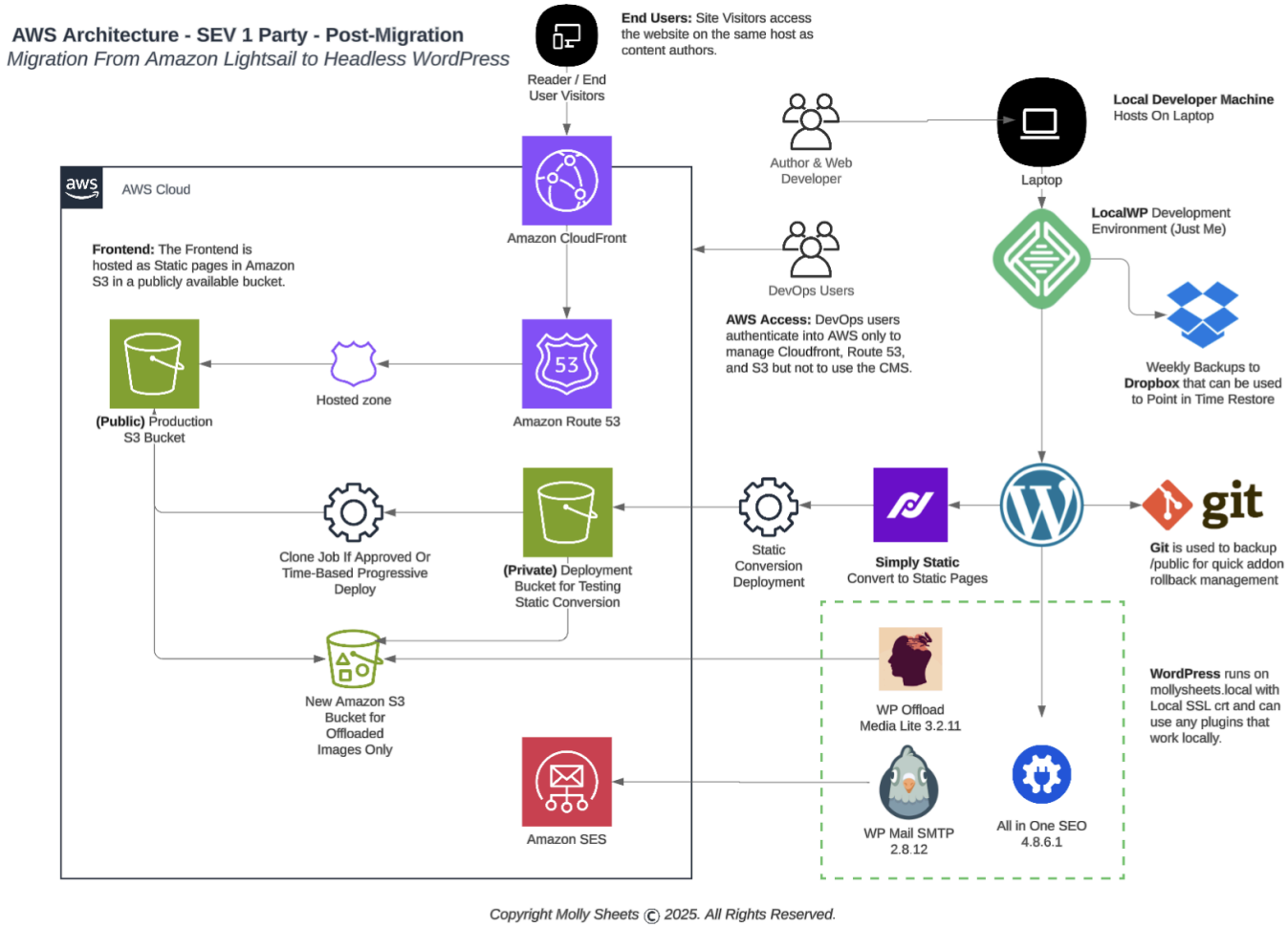
(1) End User Front-End Hosting is Static: The full visitor frontend will now be hosted in Amazon S3, not only images. This means all content MUST be static - content delivered does not change based on the specific user and is mainly HTML, CSS, and JavaScript. If in the future dynamic capabilities are required (such as would be for e-commerce, newsletters, login, or general forms), those can be distinct solutions not hosted in S3. With some exceptions as JavaScript can handle database returns from API calls processed locally in browser, for the most part content cannot modify for different users or update with real-time information and cannot rely on sever-side scripting.

(2) Author Writing and Development is Local: Post-migration, local development of the site will happen on the author’s own computer who is the only person who needs to access the CMS. This includes website maintenance, code updates, and writing of blogs and content creation. While the CMS portion of WordPress was not accessible to outside users in Lightsail, because it was physically hosted on the same machine as the website users visited, this meant if the website grew in traffic eventually readership (and honestly AI scraping bots) would overpower the compute of the website requiring either vertically scaling or more instances and complexity. Traffic overload would also bring down both the CMS and the end user facing portions in tandem. Now, both the CMS and the end-user frontend can scale independently of each other with the CMS not needing to scale beyond the needs of 1 user and multiple testing environments.

(3) End User Front-End Scalability is Virtually Unlimited: With the visitor frontend hosted in S3, the website benefits from Amazon S3’s massively parallel architecture which means it can “infinitely” scale without concern for compute sizing (vertical) or multi-instance (horizontal) scaling. This developer will no longer need to care about capacity. While WordPress can be hosted in the cloud separate from the Front-End hosting (a headless setup that is a variant of this proposed architecture), this author is not interested paying for another box without a broader team needing access to the CMS that would require more complex multi-environment setups.

(4) Maintenance & Backups is Cheaper: To test new versions of PHP, WordPress, and Plugins the website now will be fully integrated with GitHub, be able to create virtualized environments on the fly, and each website can separately be backed-up for full site DR to Dropbox. This is similar to the original flow of creating daily snapshots in Lightsail except cheaper as new virtualized environments on the local host can be started and stopped as needed without costing money even when stopped and inaccessible. I also already pay for Dropbox and have a good amount of space available I am not using.

Copyright Molly Sheets © 2025. All Rights Reserved.



Copyright Molly Sheets © 2025. All Rights Reserved.

New Plugins Inventory: The following plugins will either be used as a replacement of existing functionality that does not work in the static architecture format or installed to perform the migration.

Plugin Name	Use Case
WP Migrate Lite	Used to migrate the website to Local WP. Local WP as a tool is installed separately on a local host and not on the website itself.
Simply Static (Pro)	Used to convert, via scraping the local environment, to static pages to be hosted in Amazon S3. Pro is $99 year/site and includes support for deployment to AWS, search and forms support, as well as the WP-CLI.
Analytics Solution TBD	

LocalWP: Development and blog writing will now happen in LocalWP which can create multiple WordPress virtualized environments on the machine it is installed on. Local is an open source solution created by WP Engine that supports one-click admin and WordPress installation as well as Cloud backups via Google Drive and Dropbox for comprehensive, point in time, full-site disaster recovery. For the management of website files and version control, code will also be committed to GitHub.

WP Offload Media Lite - Rework: Because this design will be backing up and uploading the “public”

Copyright Molly Sheets © 2025. All Rights Reserved.

WordPress repo to git, anything stored in `wp-content/uploads/` will also be uploaded unless `.gitignore` is applied to the folder. If this folder is ignored as part of version control, it will not be backed up nor will it correctly publish to Amazon S3 when `public` is converted to a static site in deployment. The alternative is to still use an offload media plugin, like WP Offload Media Lite, but host all media and images in a separate bucket from the site deployment where only the URLs are referenced from the site. This model bifurcates image and media content and storage of that content from the storage and management of the static pages.

Jetpack - Replace Gallery Functionality: A few blog posts depended on aspects of Jetpack's media features, specifically the `jetpack/tiled-gallery` block and the `jetpack/image-compare` block. These must be replaced with either a new plugin or a CSS class that modifies the existing base block gallery functionality to create a similar visual look and feel. The posts using these blocks must have their gallery content re-created and re-embedded using the base block.

As this is not a photography website and many gallery plugins are bloated with features for that use case, it is preferred to use CSS via the customizer instead of installing another plugin to create the same look which can later be added to the theme's stylesheets; however, for the completion of migration, this proposes simply replacing with the base gallery features. Known posts affected are listed below.

Jetpack Dependency	Post Title	Blog Date
tiled-gallery	"Those Who Wander Might Actually Be Lost"	12/5/2022
tiled-gallery	"An Example Migration Plan & Architecture for Game Studio Archival Domains"	6/3/2023
tiled-gallery	"I Want to See the Mountains Again…And then Find Somewhere I Can Rest"	6/24/2023
image-compare	"Hilarious Cloud & DevOps T-Shirts That Don't Exist (Yet)"	10/28/2023
tiled-gallery	"On Thanksgiving I Attempted to be a "Prompt Engineer" for my Family"	11/26/2023
tiled-gallery	"East Coast Wanderer: Screen Detoxing the Atlanta Way"	5/12/2024

Jetpack - Replace Uptime Tooling: Because Jetpack provided support for uptime checks but was bloated with features, this must be replaced with another tool. Jetpack's free tier provided one check at 5 minute intervals. The two tools this migration proposal investigated were Uptime Robot and OpenStatus.Dev. Uptime Robot is popular with startups and provides 5 minute interval monitoring for free for up to 50 monitors and is owned by Pale Fire Capital. OpenStatus was started by 2 devs in 2023 to be opensource and provides 1 monitor for 10 minute intervals for free, but is completely self-funded. Neither require installation as plugins, but do require monitor setup. Both target startups / mid-market as opposed to enterprise solutions like Datadog, Grafana Cloud, and Honeycomb which are more robust observability tools. After investigation, this proposal will implement Uptime Robot.

Search - Replacement: Search will not require a separate plugin and can be setup in Simply Static with Fuse JS and Algolia API. For this implementation the migration will follow the Fuse JS implementation.

Emoji Support: Certain emojis do not work in the new model. For example, the music icon has disappeared

Copyright Molly Sheets © 2025. All Rights Reserved.

from all posts after migration into LocalWP but before static conversion.

Jetpack - Legacy Sharing Buttons: The current version of the website uses a legacy feature for social buttons on each post. Jetpack recommends replacing this with built in Sharing Buttons Block, another Jetpack feature, but for this migration these will be removed.

Cost Comparison

Per Month	Old Model	New Model
Amazon Route 53 - Hosted Zones	$0.50 (1 Hosted Zone)	$0.50 (1 Hosted Zone)
Amazon Route 53 - Registrar	$2.30 ($28/year)	$2.30 ($28/year)
Amazon CloudFront	N/A	Free Tier Eligible
Amazon S3	N/A	TBD
CMS Hosting	$20.85 - AWS Lightsail (2 Hosts, Daily backups)	$0.00 - LocalWP
GitHub	N/A	$0.00 (Free Tier)
Simply Static (Pro)	N/A	$8.25 ($99/year)
Uptime Monitoring	$0.00 (Free Tier - Jetpack)	$0.00 (Free Tier - Uptime Robot)
Analytics	$0.00 (Free Tier - Jetpack)	TBD
Dropbox (1GB Site Bi-Weekly Backups)	$0.00 (Unused)	$0.02/per month
Total Per Month	**$23.65**	**$X.XX**
Total Per Year	**$284.20**	**$X.XX**

Migration & Decommission Checkpoints

Phase 1 [3 Days] - Migrate to Local WP and Update Core Dependencies

- Install required migration plugins - WP Migrate Lite
- Export the website file as a .zip
- Create new local site in Local WP on Apache, updated version of PHP, and conversion of MariaDB to MySQL
- Tests
    - Check all blog data migrated over to new database and look for inconsistencies
    - Post a local test blog to ensure theme is still properly functioning
    - Test local search
- Site backup to GitHub of WordPress “public” files to private repo
- Remove specific plugins: AMP, Akismet, Jetpack and identify dependencies. Update migration plan with known issues and identify any additional solutions required.

Copyright Molly Sheets © 2025. All Rights Reserved.

Phase 2 [1 Day] - Setup Continuous Deployment with Simply Static Pro

- Purchase Simply Static Pro, Install, and configure in Local WordPress site
- Create a new production Amazon S3 Bucket & setup new deployment pipeline to Amazon S3 from Local
- Publish a working version of the website hitting a unique S3 URL & retest page displays served from S3
- Implement new search configuration and test in S3
- Setup Local WP backups with Dropbox

Phase 3 [1 Day] - DNS Networking & CDN

- Replace Jetpack's uptime functionality with Uptime Robot prior to DNS swap
- Setup Cloudfront with Amazon S3 for the new static website
- Setup new certificate solution
- Route53 re-routing of hosted zones and domain configuration to instead point to Amazon S3
- Perform cache busting exercise

Phase 4 [2 Days] - Replace Functionality for Removed or Adjusted Plugins

- Replace Analytics Plugin (Jetpack) with X and learn the new tool
- Implement alternative bucket for WP Offload Media Lite plugin & test with new images
- Create public blog post with new image hosted in alternative bucket

Phase 5 [2 Days] - Lightsail Decommission

- Pull all images from Amazon Lightsail bucket
- Migrate all old images to new Amazon S3 bucket
- Replace URLs in older blog posts pointing to newly moved images
- Save 2 snapshots from pre-migration for 1 year
- Stop and delete old instances in Lightsail, delete old StaticIP, and Lightsail bucket

Optional Follow-up

- Create a Stage Bucket and reconfigure deployment to deploy static files to stage first. Production deployments would clone the stage bucket instead.

Resources

- Simply Static: Is WordPress a Headless CMS? (And How to Make It One)
- Simply Static: LocalWP: Migrate an Existing Website
- Simply Static (Pro): Deploy to Amazon S3
- Simply Static: How to Minify Static Files
- WP Migrate: Import a WordPress Site to Local

Copyright Molly Sheets © 2025. All Rights Reserved.