

Phase 5: Multi-Site Automation

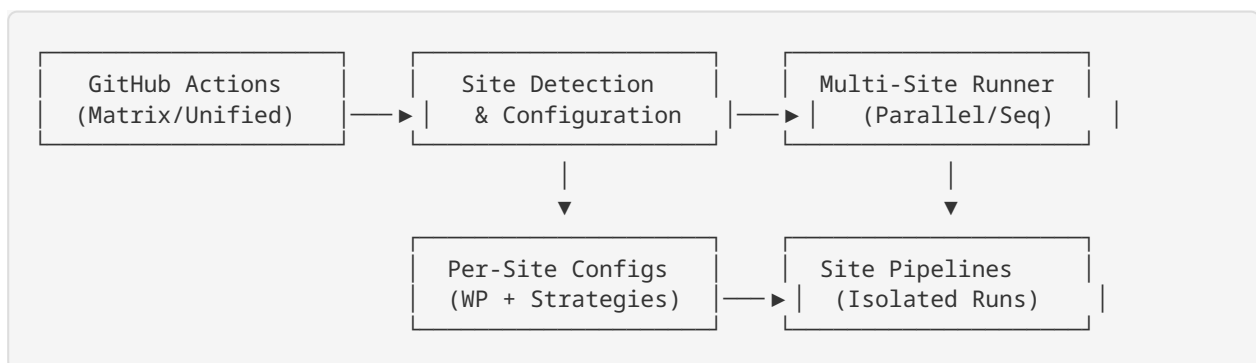
This documentation covers the enhanced multi-site automation system that allows running content pipelines across multiple WordPress sites with individual configurations and sophisticated orchestration options.

Overview

Phase 5 extends the automation system with:

- **✓ Per-Site Configuration:** Individual settings for each site via environment variables
- **✓ Multiple Execution Modes:** Parallel, sequential, or matrix job execution
- **✓ Enhanced Site Detection:** Auto-discovery of sites from environment variables
- **✓ Advanced Topic Management:** Site-specific topic counts with range support
- **✓ Flexible Prompt Strategies:** Per-site content generation strategies
- **✓ Comprehensive Logging:** Detailed logging per site with global summaries
- **✓ Matrix Job Support:** GitHub Actions matrix jobs for maximum parallelization
- **✓ Robust Configuration Validation:** Extensive validation and error handling

Architecture



Configuration

Site Detection

Sites can be configured through multiple methods:

Method 1: Explicit Site List

```
# Environment variable with comma-separated sites
ORION_SITES=travel,finance,health,technology
```

Method 2: Auto-Detection

Sites are automatically detected from environment variables with site-specific suffixes:

```
# These variables will auto-detect 'travel' and 'finance' sites
WP_URL__TRAVEL=https://travel.example.com
WP_USERNAME__FINANCE=finance_admin
TOPIC_COUNT__HEALTH=7
```

Per-Site Environment Variables

Each site can have individual configuration using the pattern `VARIABLE__SITE_KEY` :

WordPress Configuration

```
# Default WordPress credentials (used by all sites if no site-specific ones)
WP_BASE_URL=https://default.example.com
WP_USERNAME=default_admin
WP_APP_PASSWORD=default_secret

# Travel site specific
WP_URL__TRAVEL=https://travel.example.com
WP_USERNAME__TRAVEL=travel_admin
WP_APP_PASSWORD__TRAVEL=travel_secret

# Finance site specific
WP_URL__FINANCE=https://finance.example.com
WP_USERNAME__FINANCE=finance_admin
WP_APP_PASSWORD__FINANCE=finance_secret
```

Content Generation Configuration

```
# Default settings
TOPIC_COUNT=5
ENRICH_PROMPT_STRATEGY=default

# Site-specific overrides
TOPIC_COUNT__TRAVEL=3-7          # Range: 3-7 topics randomly selected
TOPIC_COUNT__FINANCE=10         # Fixed: exactly 10 topics
TOPIC_COUNT__HEALTH=2-5         # Range: 2-5 topics

ENRICH_PROMPT_STRATEGY__TRAVEL=listicle.md # Use specific prompt template
ENRICH_PROMPT_STRATEGY__FINANCE=random     # Random prompt selection
ENRICH_PROMPT_STRATEGY__HEALTH=default     # Use default strategy
```

GitHub Actions Configuration

Setting Up Secrets

In your GitHub repository settings, add secrets following the naming patterns:

Core Secrets

- `CONSOLE_BASE_URL` - Orion API base URL
- `CONSOLE_API_TOKEN` - Orion API token
- `ORION_SITES` - Comma-separated list of sites (optional if using auto-detection)

Default WordPress Credentials

- `WP_BASE_URL` - Default WordPress URL
- `WP_USERNAME` - Default WordPress username

- `WP_APP_PASSWORD` - Default WordPress app password

Site-Specific Secrets

For each site, add secrets with the pattern `VARIABLE__SITE_NAME` :

Travel Site Example:

- `WP_URL__TRAVEL`
- `WP_USERNAME__TRAVEL`
- `WP_APP_PASSWORD__TRAVEL`
- `TOPIC_COUNT__TRAVEL`
- `ENRICH_PROMPT_STRATEGY__TRAVEL`

Finance Site Example:

- `WP_URL__FINANCE`
- `WP_USERNAME__FINANCE`
- `WP_APP_PASSWORD__FINANCE`
- `TOPIC_COUNT__FINANCE`
- `ENRICH_PROMPT_STRATEGY__FINANCE`

Note: Use uppercase letters and underscores in secret names. Hyphens in site names become underscores in secret names (e.g., `health-blog` → `WP_URL__HEALTH_BLOG`).

Execution Modes

The workflow supports three execution modes:

1. Parallel Mode (Default)

Runs all sites within a single job using `ThreadPoolExecutor`:

- Faster execution
- Shared resource limits
- Combined logging

2. Sequential Mode

Runs sites one after another within a single job:

- Reliable execution
- Lower resource usage
- Easy debugging

3. Matrix Mode

Runs each site as a separate GitHub Actions job:

- Maximum parallelization
- Independent resource limits
- Isolated failure handling

Workflow Usage

Manual Trigger

Go to Actions → “Orion Multi-Site Automation” → “Run workflow”:

- **Sites:** Comma-separated list (leave empty for auto-detection)
- **Execution Mode:** `parallel`, `sequential`, or `matrix`
- **Topic Count Override:** Override all site configurations
- **Dry Run:** Test mode without WordPress publishing

Scheduled Execution

The workflow runs automatically twice daily (6 AM and 4 PM UTC) in parallel mode.

Command Line Usage

Multi-Site Pipeline Runner

```
# Run all configured sites in parallel (default)
python -m orion.automate.run_pipeline_multisite

# Run specific sites sequentially
python -m orion.automate.run_pipeline_multisite --sites travel,finance --sequential

# Override topic count for all sites
python -m orion.automate.run_pipeline_multisite --topics 10

# Dry-run mode (no WordPress publishing)
python -m orion.automate.run_pipeline_multisite --publish 0

# Control parallel execution
python -m orion.automate.run_pipeline_multisite --max-workers 5

# Skip random startup delay
python -m orion.automate.run_pipeline_multisite --no-jitter
```

Configuration Management

```
# List all detected sites
python -m orion.automate.multisite --list-sites

# Validate specific site configuration
python -m orion.automate.multisite --site-key travel

# Validate all sites
python -m orion.automate.multisite --validate-all

# Generate GitHub Actions matrix JSON
python -m orion.automate.multisite --matrix
```

Single Site Pipeline (Still Available)

```
# Run pipeline for single site
python -m orion.automate.run_pipeline --site-key travel --topics 7
```

Adding New Sites

Step 1: Configure Secrets in GitHub

Add the following secrets to your GitHub repository:

1. `WP_URL__NEWSITE` - WordPress URL for the new site
2. `WP_USERNAME__NEWSITE` - WordPress admin username
3. `WP_APP_PASSWORD__NEWSITE` - WordPress application password
4. `TOPIC_COUNT__NEWSITE` - Topic count (e.g., "5" or "3-8")

5. `ENRICH_PROMPT_STRATEGY__NEWSITE` - Content strategy (optional)

Step 2: Update Site List (Optional)

If not using auto-detection, add the new site to `ORION_SITES` :

```
ORION_SITES=existing1,existing2,newsite
```

Step 3: Verify Configuration




```
# Test the new site configuration
python -m orion.automate.multisite --site-key newsite

# Run a test pipeline
python -m orion.automate.run_pipeline --site-key newsite --topics 1 --dry-run-wp 1
```





Configuration Validation

The system includes comprehensive validation:




WordPress Configuration

-  Valid HTTP/HTTPS URLs
-  Required credentials present
-  Missing credentials → Dry-run mode

Topic Count Validation

-  Positive integers (1-20)
-  Valid ranges (e.g., “3-7”, “5-10”)
-  Invalid formats → Default to 5
-  High counts (>20) → Warning about rate limits

Prompt Strategy Validation

-  “default”, “random” strategies
-  Custom `.md` filenames
-  Invalid strategies → Error

Example Validation Output

```
Validating configuration for 3 sites: ['finance', 'health', 'travel']
finance: ✓ Configured, topics=5, strategy=random
health: ⚠ Dry-run only, topics=3-7, strategy=default
travel: ✓ Configured, topics=10, strategy=listicle.md
```

```
Summary: 2/3 sites fully configured for WordPress publishing
```

Logging and Monitoring

Multi-Site Logs

Global execution logs are stored in:

```

automation-logs/
├── multisite/           # Global multi-site logs
│   └── 2024-08-28.jsonl
├── travel/             # Site-specific logs
│   └── 2024-08-28.jsonl
├── finance/
│   └── 2024-08-28.jsonl
└── health/
    └── 2024-08-28.jsonl

```

Log Structure

Global Events

```

{
  "ts": "2024-08-28T14:30:00Z",
  "event_type": "multisite_start",
  "ok": true,
  "sites": ["travel", "finance", "health"],
  "parallel": true,
  "max_workers": 3
}

```

Site Events

```

{
  "ts": "2024-08-28T14:32:15Z",
  "site": "travel",
  "stage": "wordpress_publish",
  "ok": true,
  "topics_processed": 5,
  "drafts_created": 5,
  "errors": 0,
  "dry_run": false
}

```

GitHub Actions Artifacts

Each workflow run creates artifacts:

- automation-logs-unified-{run_id} (parallel/sequential mode)
- automation-logs-{site}-{run_id} (matrix mode)
- Retained for 30 days

Advanced Configuration

Topic Count Ranges

Use ranges to add natural variation:

```

TOPIC_COUNT__TRAVEL=3-7    # Random 3-7 topics per run
TOPIC_COUNT__FINANCE=5-12  # Random 5-12 topics per run
TOPIC_COUNT__HEALTH=2      # Always exactly 2 topics

```

Publishing Jitter

Random startup delays (0-180 seconds) when running multiple sites to avoid synchronized posting patterns:

```
# Disable jitter
python -m orion.automate.run_pipeline_multisite --no-jitter
```

Custom Prompt Strategies

Each site can use different content generation strategies:

```
ENRICH_PROMPT_STRATEGY__TRAVEL=listicle.md    # Travel listicles
ENRICH_PROMPT_STRATEGY__FINANCE=analysis.md    # Financial analysis
ENRICH_PROMPT_STRATEGY__HEALTH=how-to.md       # Health guides
ENRICH_PROMPT_STRATEGY__TECH=random            # Random prompts
```

Troubleshooting

Common Issues

“No sites configured” Error

Cause: No `ORION_SITES` variable and no site-specific environment variables detected.

Solution:

```
# Either set explicit list
ORION_SITES=mysite

# Or add site-specific variables
WP_URL__MYSITE=https://example.com
```

“Configuration issues” Warnings

Cause: Invalid URLs, topic counts, or prompt strategies.

Solution: Validate configuration:

```
python -m orion.automate.multisite --site-key problematic-site
```

Matrix Job Failures

Cause: GitHub secret naming doesn’t match site keys.

Solution: Ensure secret names match pattern:

- Site: `my-travel-blog`
- Secret: `WP_URL__MY_TRAVEL_BLOG` (hyphens → underscores, uppercase)

Debug Commands

```
# Check site detection
python -m orion.automate.multisite --list-sites

# Validate all configurations
python -m orion.automate.multisite --validate-all

# Test single site
python -m orion.automate.run_pipeline --site-key mysite --topics 1 --dry-run-wp 1

# Generate matrix JSON
python -m orion.automate.multisite --matrix
```

Log Analysis

```
# View global execution log
tail -f automation-logs/multisite/2024-08-28.jsonl

# View site-specific logs
tail -f automation-logs/travel/2024-08-28.jsonl

# Parse JSON logs
cat automation-logs/multisite/2024-08-28.jsonl | jq '.event_type'
```

Migration from Phase 4

Phase 5 is fully backward compatible. Existing single-site configurations continue to work:

Existing Setup (Still Works)

```
ORION_SITES=mysite
WP_BASE_URL=https://example.com
WP_USERNAME=admin
WP_APP_PASSWORD=secret
TOPIC_COUNT=5
```

Enhanced Setup (New in Phase 5)

```
# Auto-detected from site-specific variables
WP_URL__TRAVEL=https://travel.example.com
WP_USERNAME__TRAVEL=travel_admin
WP_APP_PASSWORD__TRAVEL=travel_secret
TOPIC_COUNT__TRAVEL=3-7
ENRICH_PROMPT_STRATEGY__TRAVEL=listicle.md

WP_URL__FINANCE=https://finance.example.com
WP_USERNAME__FINANCE=finance_admin
WP_APP_PASSWORD__FINANCE=finance_secret
TOPIC_COUNT__FINANCE=10
ENRICH_PROMPT_STRATEGY__FINANCE=random
```

The new multi-site workflow (`automation-multisite.yml`) can run alongside or replace the existing workflow.

Summary

Phase 5 transforms the Orion automation system into a sophisticated multi-site content management platform with:

- **Flexible Site Management:** Auto-detection and explicit configuration
- **Scalable Execution:** Parallel, sequential, and matrix job modes
- **Individual Site Control:** Per-site topics, strategies, and credentials
- **Production-Ready Monitoring:** Comprehensive logging and error handling
- **Easy GitHub Integration:** Matrix jobs and manual workflow triggers

This system can efficiently manage content generation across dozens of sites with minimal configuration overhead.