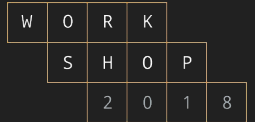


# VIP Go in 2018 précis and AMA

Simon Wheatley

simonw@automattic.com | @simonwheatley



Mo Jangda  
Josh Betz  
David Newman



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

*We would love to chat about your  
hopes for and experiences of VIP Go  
(and other platforms)*



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

# VIP is Go by default



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

# What's new?



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

# A new Cron infrastructure

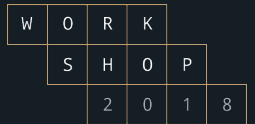
- Standard WordPress Cron events API
- Based on WP CLI
- Dedicated infrastructure
- Guaranteed scheduling
- With monitoring and alerting

# A pipeline for data sync

- Fast sync for VIP sized databases
- Handles operations needed to make the data safe
- Fast automated URL cleanup
- Media made available via our UnisonFS
- Per-application custom cleanup operations

# Jetpack Search

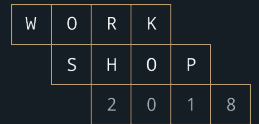
- Elasticsearch hosted by WordPress.com and Automattic
- Jetpack Sync
- Core site search
- Custom search queries





# Lets Encrypt

- Automated SSL renewals
- Custom certificates available
- HTTP to HTTPS redirection
- HSTS headers available
- SSL terminated at our edge

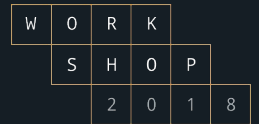


# GitHub PR Review

- Review on the GitHub Pull Request
- Conversation, collaboration, and approval
- Opportunities for CI integration... more on this later

# What else?

- IP Allow lists
- PHP 7.2
- (Nascent) NodeJS hosting capability



# What's next?



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

More power to you



W	O	R	K		
	S	H	O	P	
		2	0	1	8

# Platform CLI tooling

NEEDS SCREENGRAB



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

# Platform client dashboard

The screenshot shows a web dashboard for a client named 'testgutenberg'. On the left, there is a sidebar titled 'Applications' with a search bar at the bottom. It lists five applications: 'test-03 VIP', 'test-single-site-01 VIP', 'test-subdir-ms-01 VIP', 'testgutenberg VIP' (which is highlighted), and 'thedad Some Spider, LLC'. The main content area has a dark header with the client name 'testgutenberg', a help icon, and a user profile icon. Below the header, there is a large white box containing a 'Dashboard' title, a client logo, and a menu with 'Dashboard', 'Logs', 'Sync', and 'GitHub' options. At the bottom of the dashboard, there is the 'WVIP' logo and links for 'Terms Of Service' and 'Privacy Policy'.



W	O	R	K		
	S	H	O	P	
		2	0	1	8

# Code quality

plugins/index.php ✖ Hide outdated

```
...    ...    @@ -1,3 +1,4 @@
1      1      <?php
2      2      // Silence is golden.
3      3
4      4      +echo "Testing 123 " . time();
```

 **wpcmvip-vipgoci-bot** 22 days ago

 **Error:** All output should be run through an escaping function (see the Security sections in the WordPress Developer Handbooks), found 'time'.

 Reply...



# Self-service data sync



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

# Opportunities

- How can I confidently test new features on production infrastructure?
- How can I be sure I'm developing in the best local environment?
- How can I rapidly debug issues in production?
- How can I more directly control my applications and environments?
- How can I see and understand what usage I'm being billed for?
- How can I launch my site and deploy code faster and more confidently?
- How can I get the site live faster?
- How can I control support A/B testing?
- How can I develop or integrate with a paywall?

# Roadmap



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

More power to you



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

# Now, next, and later



W	O	R	K		
	S	H	O	P	
		2	0	1	8

# What we *are* working on now

- A client facing “platform CLI tool”
- A web-based client facing dashboard
- Self-service data sync
- Build and deploy functionality
- Linting and PHP\_CS integration with GitHub PRs

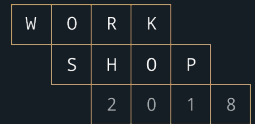
*(If you want to preview anything here, let me, Mo, or Josh know)*



W	O	R	K		
	S	H	O	P	
		2	0	1	8

# What we *plan* to work on next

- Making launching a site faster and easier
- Supporting more, and different, caching scenarios



# What we *plan* to work on later

- Running WP CLI commands
- Better supporting Git flow and QA testing
- Viewing error and access logs
- Improving local development for client teams



# AMA



W O R K  
S H O P  
2 0 1 8

*We would love to chat about your  
hopes for and experiences of VIP Go  
(and other platforms)*



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

# AMA

(Before you ask: a hundred duck sized horses, obviously)



W	O	R	K		
	S	H	O	P	
		2	0	1	8

# Topics



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

# How requests and responses flow



W	O	R	K		
	S	H	O	P	
		2	0	1	8

The Internet

Edge LB

Edge Cache

Edge LB

Edge Cache

DC LB

Files Service

Origin Site

Origin Site

DC LB

Files Service

Origin Site

Origin Site

DC LB

Files Service

Origin Site

Origin Site



W	O	R	K
S	H	O	P
	2	0	1 8

# How caching works on VIP Go

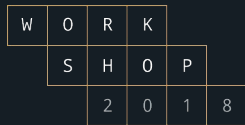


---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

# Varnish in outline

- GET a thing, cache a thing, serve the thing
- ...repeat until done
- Set the Time To Live (TTL)
- PURGE individual cache objects
- BAN certain requests with a regex
- `/mu-plugins/vip-cache-manager/`





# HTTP Response Headers

```
$ curl -sI https://www.denverpost.com/  
HTTP/2 200  
server: nginx  
date: Wed, 11 Apr 2018 14:59:04 GMT  
x-rq: lhr1 87 107 3089  
cache-control: max-age=300, must-revalidate  
age: 32  
x-cache: hit  
vary: Accept-Encoding  
accept-ranges: bytes
```



W	O	R	K		
	S	H	O	P	
		2	0	1	8

# London calling

```
$ curl -sI https://www.denverpost.com/  
HTTP/2 200  
server: nginx  
date: Wed, 11 Apr 2018 14:59:04 GMT  
x-rq: lhr1 87 107 3089  
cache-control: max-age=300, must-revalidate  
age: 32  
x-cache: hit  
vary: Accept-Encoding  
accept-ranges: bytes
```



W	O	R	K		
	S	H	O	P	
		2	0	1	8

# We hit a cache object (in London)

- `$ curl -sI https://www.denverpost.com/`
- `HTTP/2 200`
- `server: nginx`
- `date: Wed, 11 Apr 2018 14:59:04 GMT`
- `x-rq: lhr1 87 107 3089`
- `cache-control: max-age=300, must-revalidate`
- `age: 32`
- `x-cache: hit`
- `vary: Accept-Encoding`
- `accept-ranges: bytes`

# 32 seconds old

```
$ curl -sI https://www.denverpost.com/  
HTTP/2 200  
server: nginx  
date: Wed, 11 Apr 2018 14:59:04 GMT  
x-rq: lhr1 87 107 3089  
cache-control: max-age=300, must-revalidate  
age: 32  
x-cache: hit  
vary: Accept-Encoding  
accept-ranges: bytes
```



W	O	R	K		
	S	H	O	P	
		2	0	1	8

# Max-Age != Age != TTL

```
$ curl -sI https://www.denverpost.com/  
HTTP/2 200  
server: nginx  
date: Wed, 11 Apr 2018 14:59:04 GMT  
x-rq: lhr1 87 107 3089  
cache-control: max-age=300, must-revalidate  
age: 32  
x-cache: hit  
vary: Accept-Encoding  
accept-ranges: bytes
```

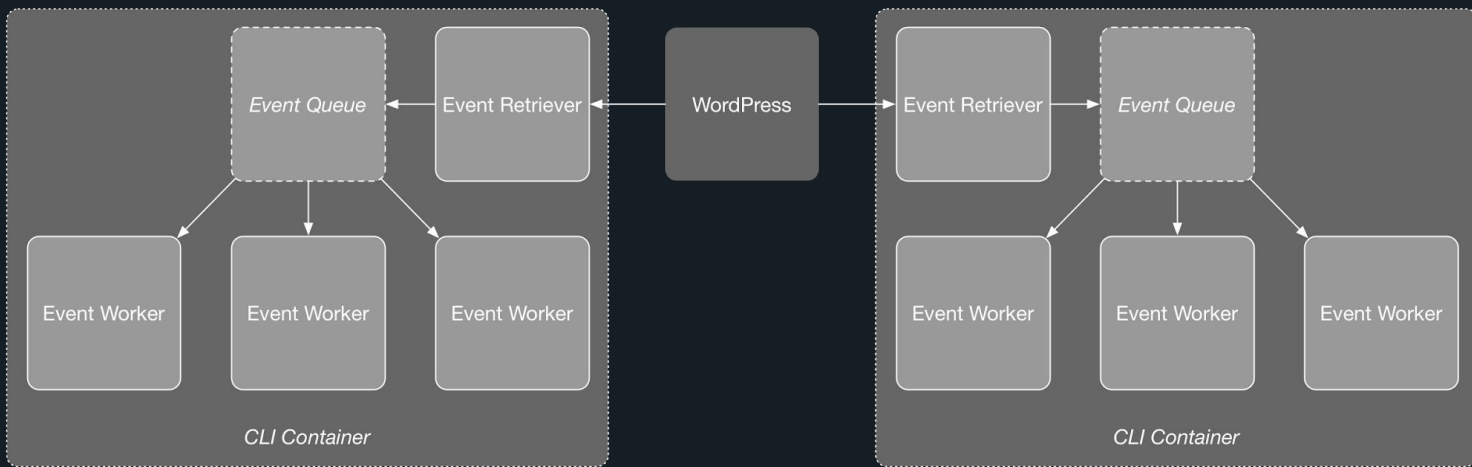


W	O	R	K		
	S	H	O	P	
		2	0	1	8

# How Cron works on VIP Go



W	O	R	K		
	S	H	O	P	
		2	0	1	8



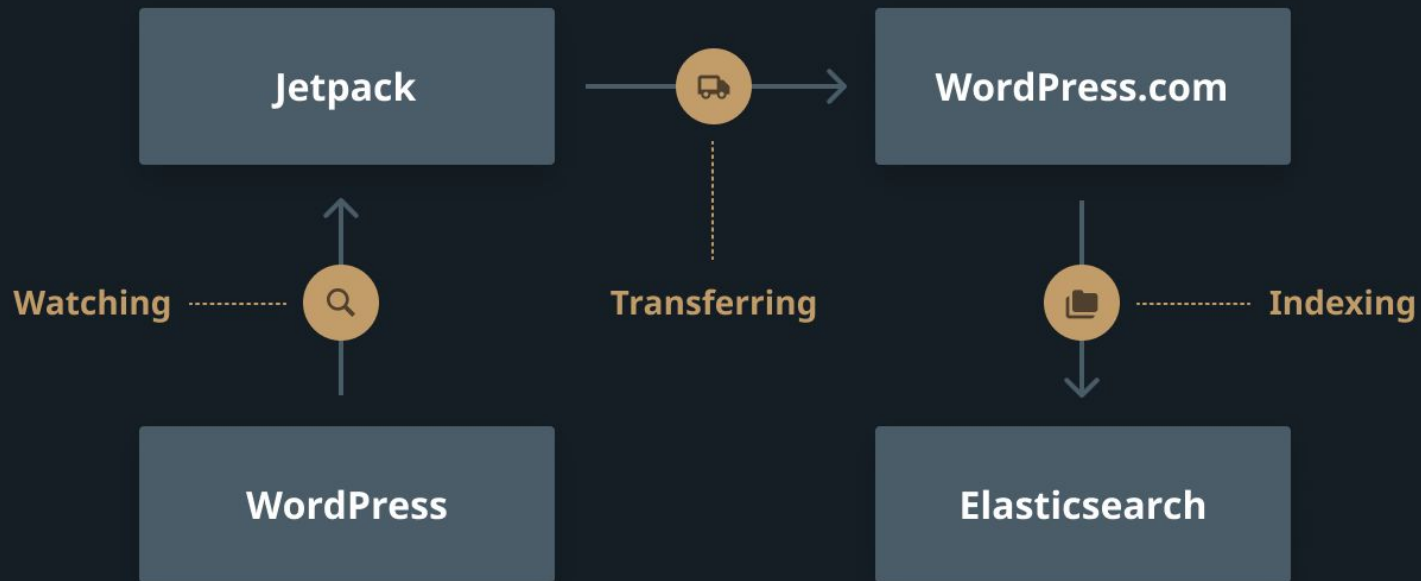
# Jetpack Search on VIP Go



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8

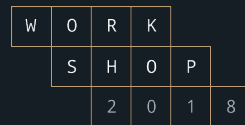




# How deployment works on VIP Go

1. Push to the master branch
2. GitHub notifies VIP Go via a webhook
3. VIP Go extracts the commit SHA
4. VIPd deploys that SHA to all containers

*Note that we can manually specify a SHA at Step 3 to set the site to a particular commit*



# Thank you!

Simon Wheatley  
@simonwheatley



---

W	O	R	K		
	S	H	O	P	
		2	0	1	8