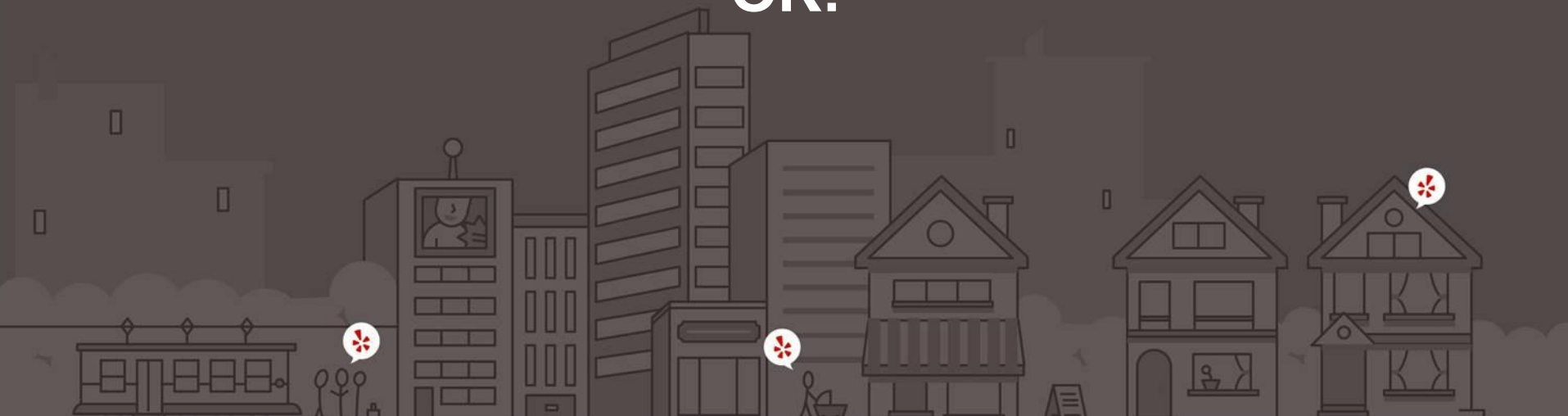




**Schema Changes Multiple Times a Day?
OK!**



Who am I?

MySQL DBA ~10 years

1st DBA at Yelp, been here ~4 yrs

(yes, slides will be online)



A Bit About Yelp

- *"Yelp is a website and mobile app that connects people with great local businesses"*
- Founded 2004
- More than just reviews!



Yelp's DBs

- Many logical shards, hundreds of tables
- 100+ of db servers
- Most small, several 10+ GB tables
- No horizontal sharding yet
- No Foreign Keys



More About Yelp

- Yelp pushes all the time
 - Main application 3-hecka times a day
 - Services pushed independently
- We now have 300+ engineers + interns



Culture

- Yelp + developer productivity = <3
- Ops & Devs aligned
- But, schema changes, ALL the time?



This Freaked Me Out

- Developer DB access
 - `GRANT ALTER,CREATE,DROP ON yelp.* TO 'yournamehere'@'10.xyz.%'`
- No scheduled downtime windows

... but DBAs are now free to do more fun stuff



Spoiler Alert

1. We have to make schema changes
2. Make them a regular, *safe* process
3. Use/write tools



You Need Schema Changes!

- New features
- Fix bugs



Why Are Schema Changes Scary?

- Lock table, create copy
- I/O slow, degraded performance
- Replication delay

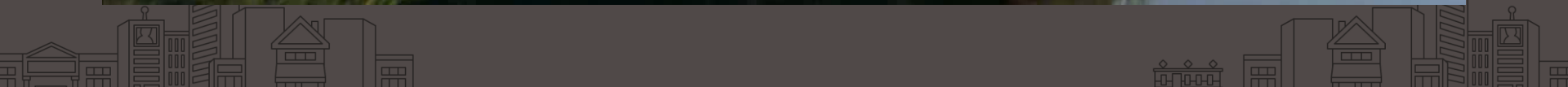


Online DDL?

- Concerns about I/O
- Performance
- Exclusive lock on table



You Can't Just Keep Adding Tables



So Then What?

We needed to support schema changes:

- That could be run any time
- Backwards/forwards compatible with app
- No site/performance impact



The DevOps Part

1 part culture, 1 part technology:

Culture:

- Empower, trust

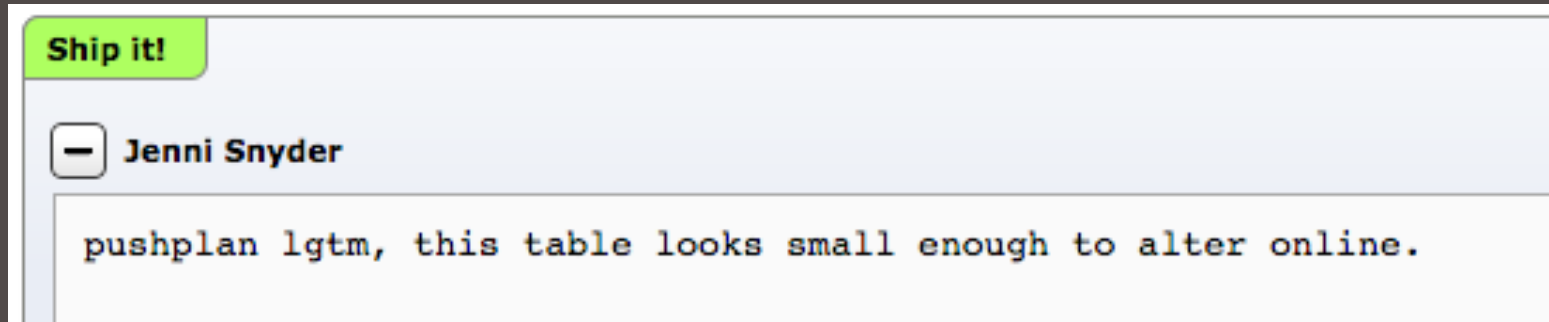
Technology:

- Tools that create confidence



About That Trust

If you can't trust trained team members, who can you?



How We Do This

Implement process around schema changes

- Rollback safety
- DBAs and Pushmasters
- pt-online-schema-change
- Other tools



Rollback Safety

Applications aren't deployed in a heartbeat

Schema changes must be safe

- NOT NULL
- Dropping/Renaming columns
- Modifying types/length



DBAs

DBAs are available

DBAs are are CC'd on code reviews w/ DDL

- Actual DDL is committed in text file
- Automation ensures DBAs get eyes on CR
- Process checks for DBA "Ship it!"



Pushmasters

"Pushmasters are Role Models"

"Pushmasters are Guides"

"Pushmasters are not Robots"

killer documentation for process*



pt-online-schema-change

- How pt-online-schema-change works
- Gotchas
 - triggers
 - table metadata lock



pt-online-schema-change

- Modified to query our repl_delay_reporter
- We use these, some other arguments:
 - chunk-time
 - max-lag
 - critical-load
 - no-drop-old-table
 - dry-run



Other Tools

Development

- Unit testing*
- Gross Query Checker*
- Reviewboard

*Blog post on Yelp Engineering Blog



(More) Other Tools

Pushing/Prod

- Pushmanager*
- pt-kill, query mortuary
- Monitoring, graphing

*blog post on Yelp Engineering Blog



Replication Delay

Central service repl_delay_reporter



When Are Schema Changes Made?

- Schema changes are run before associated code push
- Announced in IRC
- Send signal to graphing/monitoring



How Can YOU Do This?

- Work with your devs
- Separate schema changes from code
- Push bulk DML into code
- pt-online-schema-change
- Know your replication delay limits



The Future

- More automation
- All changes made w/ pt-online-schema-change
- Liquibase?



Conclusion

1. You have to make schema changes
2. Scary? Make it safe
3. Write/Use tools



Follow me! [@jcsuperstar](#)

Related GitHub Projects:

<https://github.com/Yelp/pushmanager>


Related Yelp Engineering Blog Posts:
[Now, Testify!](#)

[GrossQueryChecker: Don't get surprised by query performance in production!](#)

 YelpEngineers

 @YelpEngineering

 engineeringblog.yelp.com

 github.com/yelp

Come see us at booth 402!

