Automating Schema Migrations
with GitHub Actions, skeema & gh-ost

Shlomi Noach
GitHub

FOSDEM 2020
About me

@github/database-infrastructure

Author of orchestrator, gh-ost, freno, ccql and others.

Blog at http://openark.org

github.com/shlomi-noach
@ShlomiNoach
Built for developers

Busy and growing

40M+ developers
3M organizations
44M repositories created in the past year

Actions, Packages,
Security Advisories & Updates,
Code Navigation & Search,
Notifications, Sponsors,
Mobile, …
Incentive

With MySQL as the backend, new features imply schema changes: new tables, new columns, index changes, iterative schema changes, experiments.
What’s in a migration?

More than **CREATE, ALTER** or **DROP TABLE**
What’s in a migration?

Designing, coding, local testing, review process, queueing, scheduling, executing, controlling, auditing, versioning…
# Ownership

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer</td>
<td>Design code + schema change</td>
</tr>
<tr>
<td>Developer</td>
<td>Publish</td>
</tr>
<tr>
<td>Peer</td>
<td>Review</td>
</tr>
<tr>
<td>DBA</td>
<td>Review</td>
</tr>
<tr>
<td>DBA</td>
<td>Formalize statement/command</td>
</tr>
<tr>
<td>DBA</td>
<td>Locate</td>
</tr>
<tr>
<td>DBA</td>
<td>Schedule</td>
</tr>
<tr>
<td>DBA</td>
<td>Run</td>
</tr>
<tr>
<td>DBA/SRE</td>
<td>Audit/monitor/control</td>
</tr>
<tr>
<td>DBA</td>
<td>Cut-over/complete migration</td>
</tr>
<tr>
<td>DBA</td>
<td>Cleanup</td>
</tr>
<tr>
<td>DBA</td>
<td>Notify</td>
</tr>
<tr>
<td>Developer</td>
<td>Deploy</td>
</tr>
<tr>
<td>Developer</td>
<td>Merge</td>
</tr>
</tbody>
</table>
Ownership, our previous state

**Developer**
- Design code + schema change

**Developer**
- Publish

**Peer**
- Review

**DBA**
- Review

**DBA**
- Formalize statement/command

**DBA**
- Locate

**DBA**
- Schedule

**DBA**
- Run

**gh-ost/chatops**
- Audit/monitor/control

**DBA**
- Cut-over/complete migration

**DBA**
- Cleanup

**DBA**
- Notify

**DBA**
- Deploy

**DBA**
- Merge
A complex flow

Multiple domains (code, MySQL, production, communication).

Multiple environments (dev, production).

Multiple owners (devs, DBAs, SREs).
A combinatory solution

Loosely coupled, independent components.

Each solves an aspect of the problem.

Orchestrated to create an automated flow.
Code

A schema change should be presented as code.

Coupled with application code.

Versioned.
At GitHub, we use git.

We also happen to author GitHub for code hosting, versioning and management.
Pull Request

The change

Review

CI

Discussion
Demonstrating **skeefree** for FOSDEM 2020.

In a schema change PR we make no code changes; it’s just the schema. The developer should create a preceding/followup PR such that the code supports the change.
New PR

```
CREATE TABLE `some_table` (  
  `time_created` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `time_updated` datetime DEFAULT NULL,  
  `random_hash` char(48) CHARACTER SET ascii DEFAULT NULL, 
  PRIMARY KEY (`id`),  
  UNIQUE KEY `random_hash_key` (`random_hash`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```
What’s the migration?

From code to SQL statement
skeema

https://www.skeema.io/
https://github.com/skeema/skeema

Open source

Developed by Evan Elias
schema/
  .skeema
my_schema1/
  .skeema
  some_table.sql
  another_table.sql
my_schema2/
  .skeema
  foo.sql
  bar.sql
CREATE TABLE `some_table` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `hostname` varchar(128) NOT NULL,
  `time_created` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `time_updated` datetime DEFAULT NULL,
  `random_hash` char(40) CHARACTER SET ascii DEFAULT NULL,
  PRIMARY KEY (`id`) ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
$ skeema push env-name
# Connects to MySQL server, applies filesystem schema onto database

$ skeema pull env-name
# Imports schema definition from MySQL onto filesystem

$ skeema diff env-name
# Prints schema migrations changing the state of MySQL server to filesystem
GitHub Actions

Kick off workflows with GitHub events like push, issue creation, or a new release.

An action runs in a container on GitHub’s infrastructure (default).

Action has repository’s context and can operate on the repository.
GitHub Actions

Run skeema from within Action.

Fetch skeema as part of Action flow.
GitHub Action: skeema-diff, simplified

skeema-diff:
  runs-on: ubuntu-latest
  steps:
  - uses: actions/checkout@v2
    with:
      ref: master
  - name: push master schema to MySQL
    env:
      MYSQL_PWD: root
    run: |
      skeema push skeema-diff-ci
  - uses: actions/checkout@v2
  - name: skeema diff
    skeema diff skeema-diff-ci --allow-unsafe
GitHub Action: skeema-diff

- uses: actions/checkout@v2
  with:
    ref: ${{ github.event.pull_request.base.sha }}
- name: push master schema to MySQL
...
- uses: actions/checkout@v2
  with:
    ref: ${{ github.event.pull_request.head.sha }}
- name: skeema diff
Schema change demo for FOSDEM 2020 #64

Open shlomi-noach wants to merge 1 commit into master from fosdem-2020-demo

Conversation 0  Commits 1  Checks 2  Files changed 1

Schema change demo for FOSDEM 2020 a4b4b4

* skeema-diff
  - on: pull_request

  ✓ skeema-labels-validation

  ✓ skeema-diff

  skeema-diff / skeema-diff
  started 10s ago

  ✓ Set up job
  ✓ Run actions/checkout@v1
  ✓ Find skeema file
  ✓ Fetch skeema
  ✓ Fetch hub
  ✓ push master schema to MySQL
  ✓ Run actions/checkout@v1
  ✓ skeema diff
  ✓ validate diff
  ✓ check label
  ✓ validate migration:skeefree:queued status
  ✓ label PR
  ✓ comment PR
  ✓ comment suggestion for review
Demonstrating `skeefree` for FOSDEM 2020.

In a schema change PR we make no code changes; it's just the schema. The developer should create a preceding/followup PR such that the code supports the change.

```
-- skeema:diff
-- skeema:ddl:use skeefree_test;
-- skeema:ddl:begin
ALTER TABLE some_table ADD COLUMN fosdem_year INT(10) UNSIGNED NOT NULL DEFAULT '2020', DROP KEY random_hash_key;
-- skeema:ddl:end
```

@shlomi-noach it looks like you are making schema changes. When ready for production review & migration, please add the label `migration:for:review`. 
gh-ost

GitHub’s online schema migration tool.

Low-impact (nearly no-impact) in production.

Auditable, configurable, controllable.

Open source

https://github.com/github/gh-ost/
https://github.blog/2016-08-01-gh-ost-github-s-online-migration-tool-for-mysql/
https://speakerdeck.com/shlominoach/githubs-online-schema-migrations-for-mysql
skeefree

Name coined by Tom Krouper

A service to orchestrate the flow: PR, Actions/skeema, gh-ost.

Developed internally at GitHub
skeefree

Probes and detects schema change PRs

Analyzes schema changes

Initiates and follows up on review/approval status

Schedules the migration

Runs the migration (gh-ost/direct)

Follows up and reports on PR
Schema change demo for FOSDEM 2020 #64

shlomi-noach commented now

Demonstrating skeefree for FOSDEM 2020.

In a schema change PR we make no code changes; it's just the schema. The developer should create a preceding/followup PR such that the code supports the change.

Review required
At least 1 approving review is required by reviewers with write access. Learn more.

Some checks haven't completed yet
2 queued checks

- skeema-diff / skeema-labels-validation (pull_request)  Queued — Waitin...
- skeema-diff / skeema-diff (pull_request)  Queued — Waiting to run this c...

Merging is blocked
Merging can be performed automatically with 1 approving review.
Flow:

New PR

```
CREATE TABLE `some_table` (  
  `time_created` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `time_updated` datetime DEFAULT NULL,  
  `random_hash` char(40) CHARACTER SET ascii DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `random_hash_key` (`random_hash`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```
Flow: CI build

Schema change demo for FOSDEM 2020 #64

skeema-diff

skeema-diff / skeema-diff
started 10s ago

- Set up job
- Run actions/checkout@v1
- Find skeema file
- Fetch skeema
- Fetch hub
- push master schema to MySQL
- Run actions/checkout@v1
- skeema diff
- validate diff
- check label
- validate migration:skeefree:queued status
- label PR
- comment PR
- comment suggestion for review
Flow: skeema analysis

Schema change demo for FOSDEM 2020 #64

shlomi-noach commented 33 seconds ago

Demonstrating `skeefree` for FOSDEM 2020.

In a schema change PR we make no code changes; it's just the schema. The developer should create a preceding/followup PR such that the code supports the change.

github-actions bot added the `migration:skeema:diff` label now

github-actions bot commented now

```
-- skeema:diff
-- skeema:ddl:use skeefree_test;
-- skeema:ddl:begin
ALTER TABLE some_table ADD COLUMN fosdem_year int(10) unsigned NOT NULL DEFAULT '2020', DROP KEY random_hash_key;
-- skeema:ddl:end
```

github-actions bot commented now

@shlomi-noach it looks like you are making schema changes. When ready for production review & migration, please add the label `migration:for:review`.

1 participant
Flow: review & label

@shlomi-noach it looks like you are making schema changes. When ready for production review & migration, please add the label migration:for:review.

shlomi-noach added the migration:for:review label 1 minute ago

hubot approved these changes now
Flow: skeefree analysis

Migration instructions for @github/database-infrastructure:

```
:s "skeefree_test" -t "some_table" -d 'ADD COLUMN `fosdem_year` int(10) unsigned NOT NULL DEFAULT '}
```
Flow: Review request

Migration instructions for @github/database-infrastructure:

```
-s "skewfree_test" -t "some_table" -d 'ADD COLUMN `fosdem_year` int(18) unsigned NOT NULL DEFAULT '
```

- `hubot` commented 1 minute ago
  - added the `migration:skewfree:detected` label 1 minute ago
  - requested a review from `github/database-infrastructure` now

Reviewers
- `hubot`
- Database Infrastructure

Assignees
- No one—assign yourself

Labels
- `migration:for:review`
- `migration:skewfree:detected`
- `migration:skeema:diff`
Flow: migration execution

@shlomi-noach All migrations in this PR are in complete status. Please go ahead and follow your standard deploy/merge flow.
Flow: deploy & merge

Schema change demo for FOSDEM 2020 #64

shlomi-noach commented 10 minutes ago

Demonstrating skeefree for FOSDEM 2020.

In a schema change PR we make no code changes; it’s just the schema. The developer should create a preceding/followup PR such that the code supports the change.
Ownership: skeefree

- Developer: Design code + schema change
- Developer: Publish
- Peer: Review
- DBA: Review
- skeema/CI: Formalize statement/command
- skeefree: Locate
- skeefree: Schedule
- skeefree: Run
- gh-ost/chatops: Audit/monitor/control
- skeefree: Cut-over/complete migration
- GC: Cleanup
- skeefree: Notify
- Developer: Deploy
- Developer: Merge
Impact

Database team work reduced to minutes per week.

Developers have visibility into status. Get notified on their PR.

Better time utilization; migrations start executing as soon as possible, not based on a human availability.
skeefree

Uses internal services in GitHub’s infrastructure

- Inventory service
- MySQL discovery service
- Chatops integration
- Internal libraries (e.g. logging)
Open Source

skeefree is coupled with GitHub’s infrastructure:

- Inventory service
- MySQL discovery
- Chat/chatops

We nonetheless hope that the community finds it useful and are releasing it in partial state.

Release to be announced.
Thank you!

Questions?

[github.com/shlomi-noach](https://github.com/shlomi-noach)
@ShlomiNoach