



# Content testing and deployment done the right way

---

By Shy Ruparel



# Content model changes done the right way part 2.

---

By Shy Ruparel



# Shy Ruparel

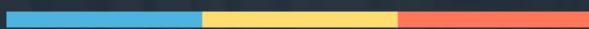
---

Developer Evangelist

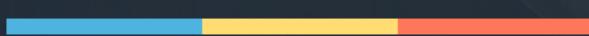
Twitter: [@ShyRuparel](https://twitter.com/ShyRuparel)

Talks: [GitHub/Shy/Talks](https://github.com/Shy/Talks)

# Building Software



*Nobody gets it right the first time*





When you can't have  
perfection, the next best thing  
is change.

---

# What could go wrong?



In a past meetup we talked about the  
Contentful Migration CLI 

---

# TL;DR

## What you can do

- Create content type
- Delete content type
- Edit content type
- Create/edit/delete fields
- Change field ID

# All in Javascript

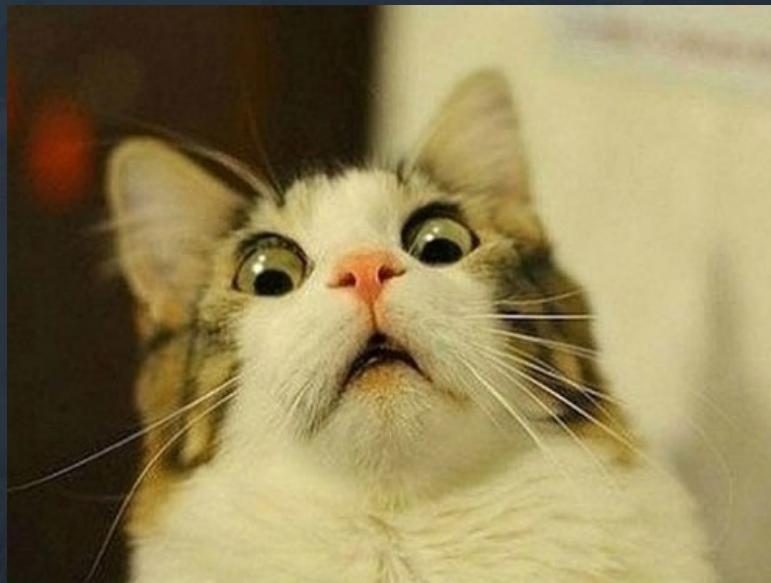
```
contentful-migration
  --space-id $YOUR_SPACE_ID
  --access-token $CONTENTFUL_MANAGEMENT_ACCESS_TOKEN
migration-demo.js
```

# Advantages

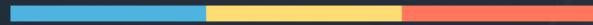
- Repeatable
- Dry-runs
- Can be kept in version control
- Sanity checks
- Use CI to apply.

# That's nice

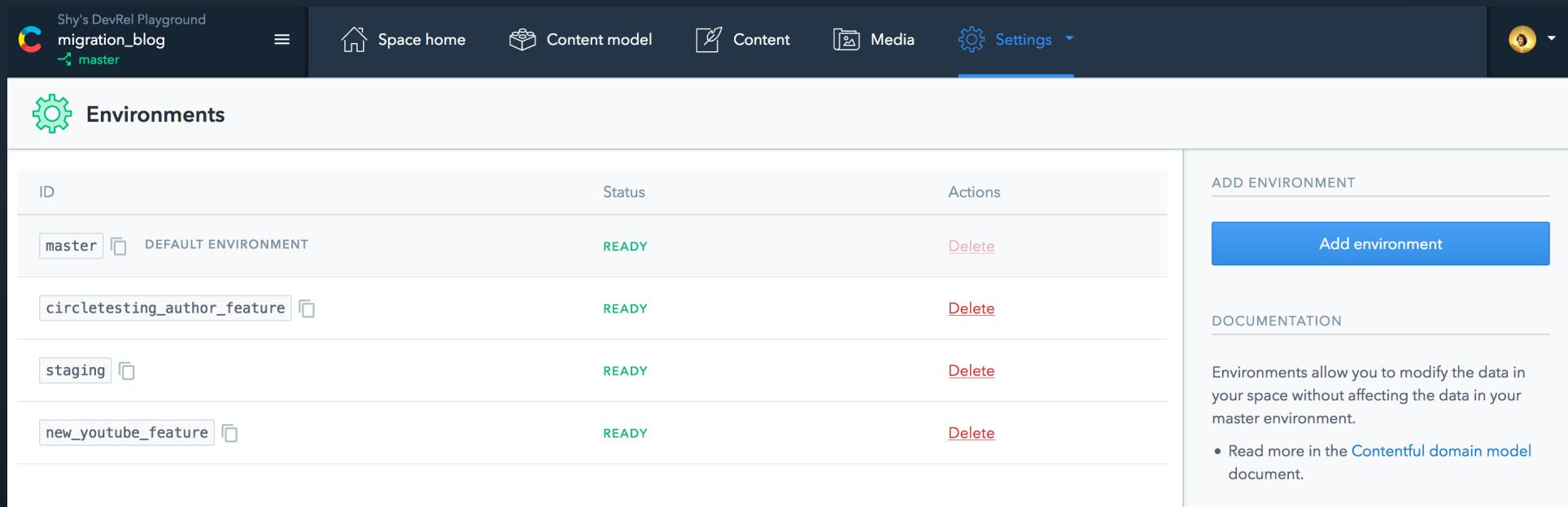
But what if I want to discard, test or preview a migration?



# Space Environments



# Space environments allow you to create multiple versions of the space, and change them in isolation.



The screenshot shows the Contentful Space environments interface. At the top, there's a navigation bar with icons for Home, Content model, Content, Media, and Settings (which is currently selected). Below the navigation bar is a header with the space name "Shy's DevRel Playground" and a "migration\_blog" environment. The "master" environment is highlighted as the default. On the left, there's a sidebar titled "Environments" with a gear icon. The main content area displays a table of environments:

ID	Status	Actions
master	READY	<a href="#">Delete</a>
circletesting_author_feature	READY	<a href="#">Delete</a>
staging	READY	<a href="#">Delete</a>
new_youtube_feature	READY	<a href="#">Delete</a>

To the right of the table, there's a sidebar with sections for "ADD ENVIRONMENT" (containing a blue "Add environment" button) and "DOCUMENTATION". The documentation section contains text about environments and a list item pointing to the "Contentful domain model" document.

**ADD ENVIRONMENT**

Add environment

**DOCUMENTATION**

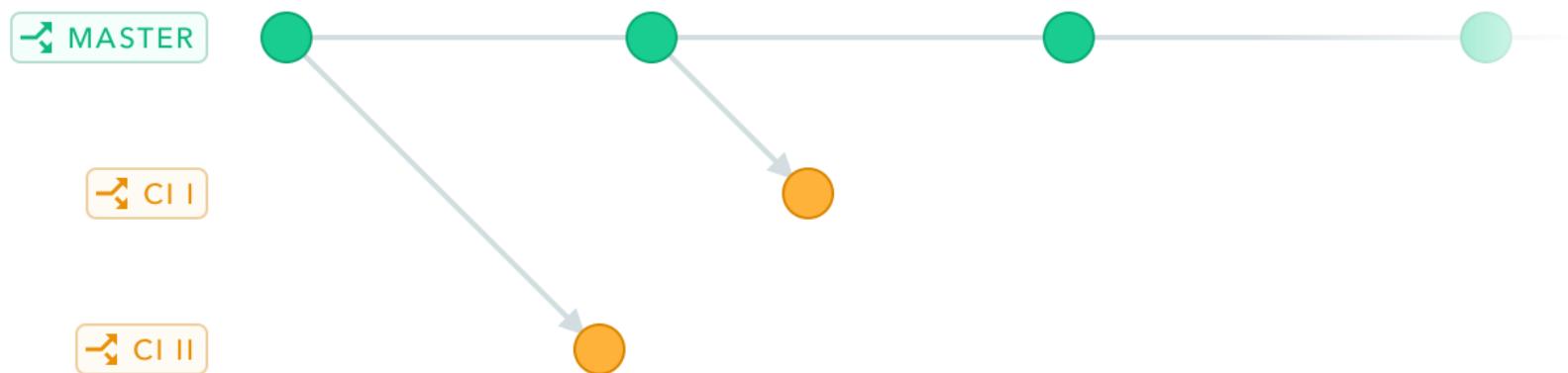
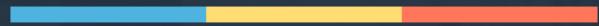
Environments allow you to modify the data in your space without affecting the data in your master environment.

- Read more in the [Contentful domain model](#) document.

# Common uses for space environments

- Local development
- Staging / QA
- Continuous integration

# Let's talk about Continuous Integration



# A simple Continuous Integration pipeline:

- Build
- Test
- Deploy

# Integrating Migrations into CI Pipeline:

- Build
  - *Spin up a copy of the master space*
  - *Migrate copy*
- Test
- Deploy
  - *Migrate master space*



circleci

# Build & Test

Shy

Builds » Shy » migration-env-demo » author\_feature » 74 (build) 2.0

Rerun job with SSH

Test Summary Queue (00:01) Artifacts Configuration Build Timing Build Parameters

Show containers: All (1) Successful (1) Failed (0)

0 (00:45) TEST

Spin up Environment 00:18

Checkout code 00:00

Restoring Cache 00:01

install dependencies 00:14

Saving Cache 00:00

Preparing environment for testing 00:06

run tests 00:04

WORKFLOWS INSIGHTS ADD PROJECTS TEAM SETTINGS

This screenshot shows a build and test interface. At the top, there's a header with a user profile, navigation links for 'Updates' and 'Support', and a gear icon. The main area displays a build summary for project 'migration-env-demo' under branch 'author\_feature', build number 74. The summary includes tabs for 'Test Summary', 'Queue (00:01)', 'Artifacts', 'Configuration', 'Build Timing', and 'Build Parameters'. A prominent blue button labeled 'Set Up Test Summary' is visible. Below the summary, it says 'Show containers: All (1) Successful (1) Failed (0)'. A timeline bar at the top shows a green segment for '0 (00:45)' followed by a cyan segment for 'TEST'. The detailed timeline below lists several steps with their execution times: 'Spin up Environment' (00:18), 'Checkout code' (00:00), 'Restoring Cache' (00:01), 'install dependencies' (00:14), 'Saving Cache' (00:00), 'Preparing environment for testing' (00:06), and 'run tests' (00:04). On the left side, there's a vertical sidebar with icons for 'WORKFLOWS', 'INSIGHTS', 'ADD PROJECTS', 'TEAM', and 'SETTINGS'.

# Deploy

Shy

Builds » Shy » migration-env-demo » master » 65 (deploy) 2.0

Rerun job with SSH

Test Summary Queue (00:02) Artifacts Configuration Build Timing Build Parameters

**Set Up Test Summary**

Show containers: All (1) Successful (1) Failed (0)

0 (00:56) TEST

Spin up Environment	00:26
Checkout code	00:00
Restoring Cache	00:02
install dependencies	00:06
Saving Cache	00:00
Updating Master Environment	00:02
Deploy Master to Heroku	00:17

# CircleCI config.yml - Ready Migrations

```
- run:  
  name: Preparing environment for testing  
  command: |  
    . venv/bin/activate  
    python migration_prep.py "circletesting_${CIRCLE_BRANCH}"  
    node_modules/contentful-migration/bin/contentful-migration  
      --access-token $MANAGEMENT_API_KEY --space-id ${SPACE_ID}  
      --environment-id "circletesting_${CIRCLE_BRANCH}"  
      --yes  
    migration-tests/migration_helper.js
```

# migration\_prep.py

```
import contentful_management
import os, sys
from dotenv import load_dotenv

SPACE_ID = os.getenv("SPACE_ID")
MANGEMENT_API_KEY = os.getenv("MANGEMENT_API_KEY")
TESTING_ENV = sys.argv[1]

client = contentful_management.Client(MANGEMENT_API_KEY)
environment = client.environments(SPACE_ID)
    .create(TESTING_ENV, {"name": TESTING_ENV})
```

# Version Tracking Content Model

The screenshot shows a dark-themed Content Management System interface. At the top, there's a header bar with the following elements from left to right:

- A logo icon.
- The workspace name: "Shy's DevRel Playground".
- The current space name: "migration\_blog".
- A branch indicator: "master".
- A three-dot menu icon.
- A "Space home" icon.
- A "Content model" icon.
- A "Content" icon.
- A "Media" icon.
- A "Settings" icon.

The main content area has a title bar with a back arrow, a "version\_tracking" title with a green building icon, and an "Edit" link. Below this, there are two tabs: "Fields (1)" which is selected and highlighted with a blue underline, and "JSON preview".

The "Fields (1)" section displays a single field entry:

		<b>version</b>	Short text	Entry title	<a href="#">Settings</a>	...
--	--	----------------	------------	-------------	--------------------------	-----

# Version Tracking Content

Content type	version_tracking	Type to search for entries	Filter	Save as view	Add entry
	Name	Updated	Author	Status	
	<input type="checkbox"/> 0.0.8	2 hours ago	Me	PUBLISHED	

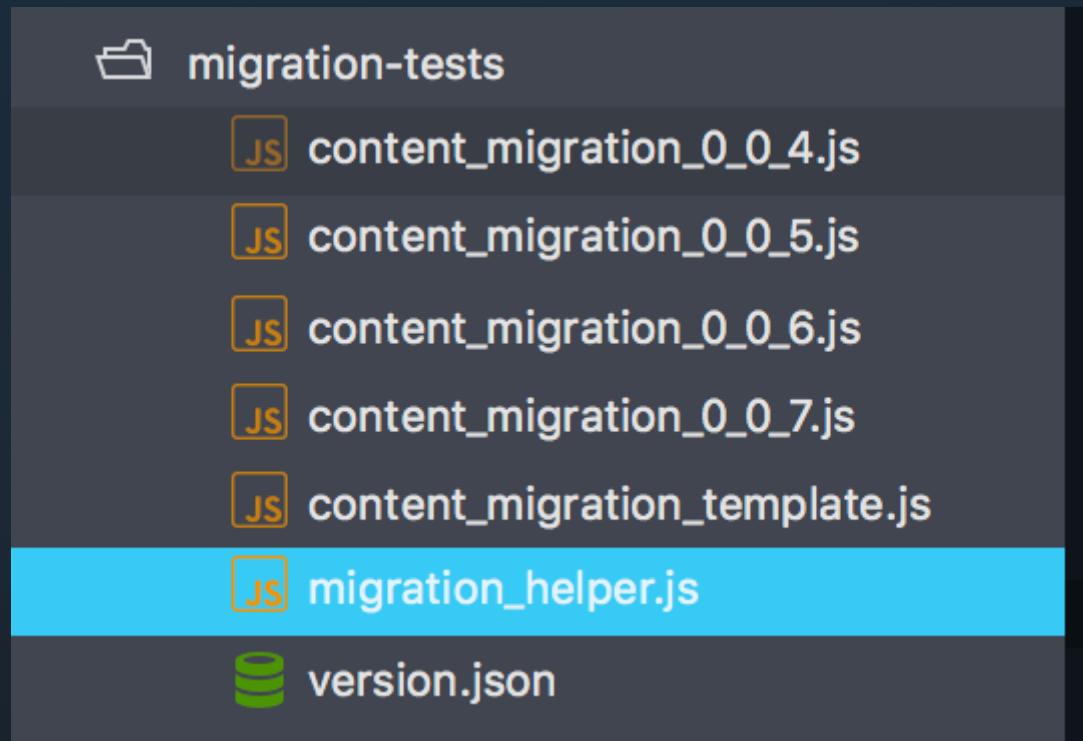
# migration\_helper.js

```
module.exports = function(migration) {
    var parsedJSON = require("./version.json");
const upgradeVersion = parsedJSON.upgradeVersion;

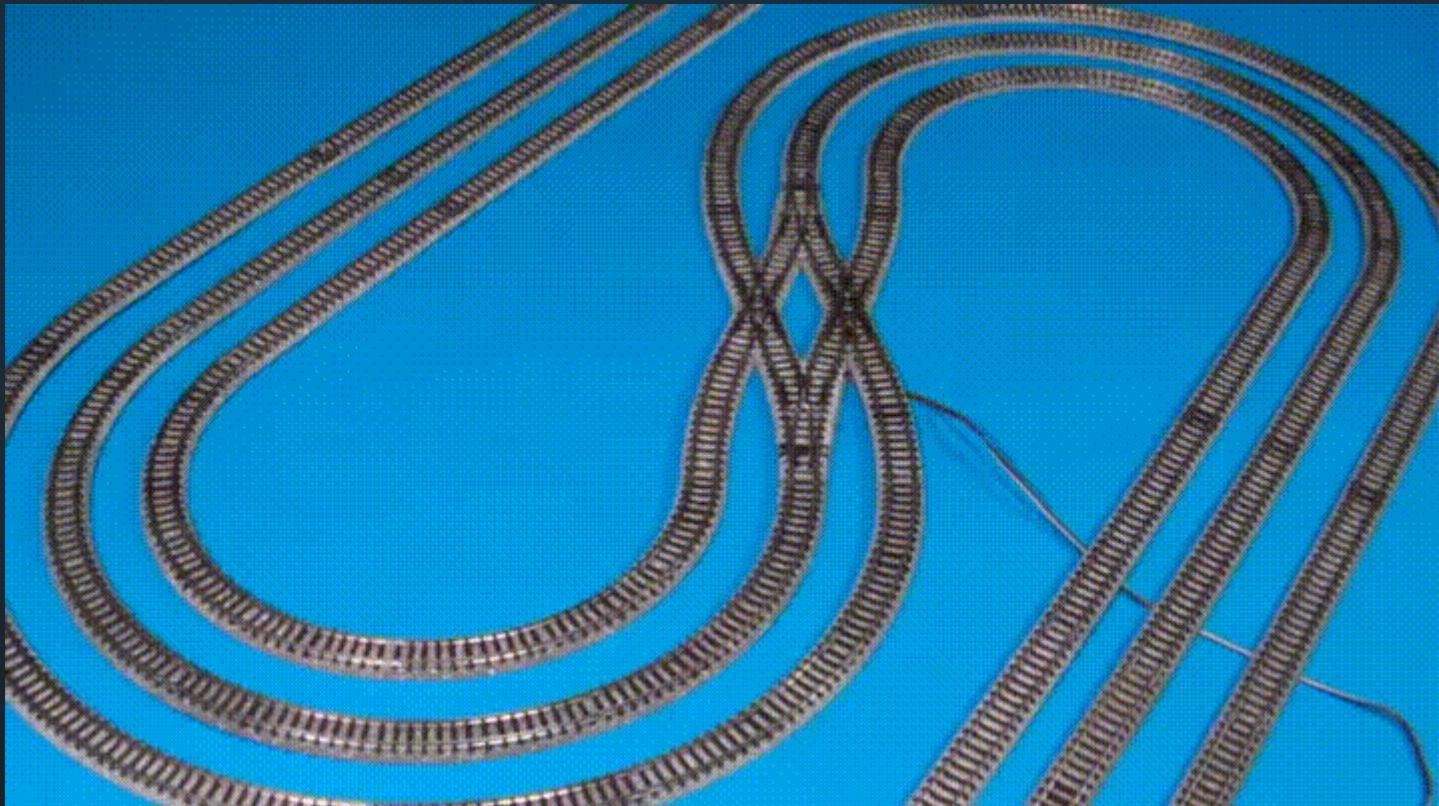
migrationString =
    "./content_migration_" +
    upgradeVersion.split(".").join("_") +
    ".js";
var migrationCode = require(migrationString);

migrationCode.runMigration(migration);
}
```

# Migration Files



# Keeping code and content model in sync



# content\_migration\_0\_0\_7.js

```
function runMigration(migration) {
  const post = migration.editContentType("post");
  post
    .createField("author")
    .name("author")
    .type("Symbol")
    .required(false);
  return;
}

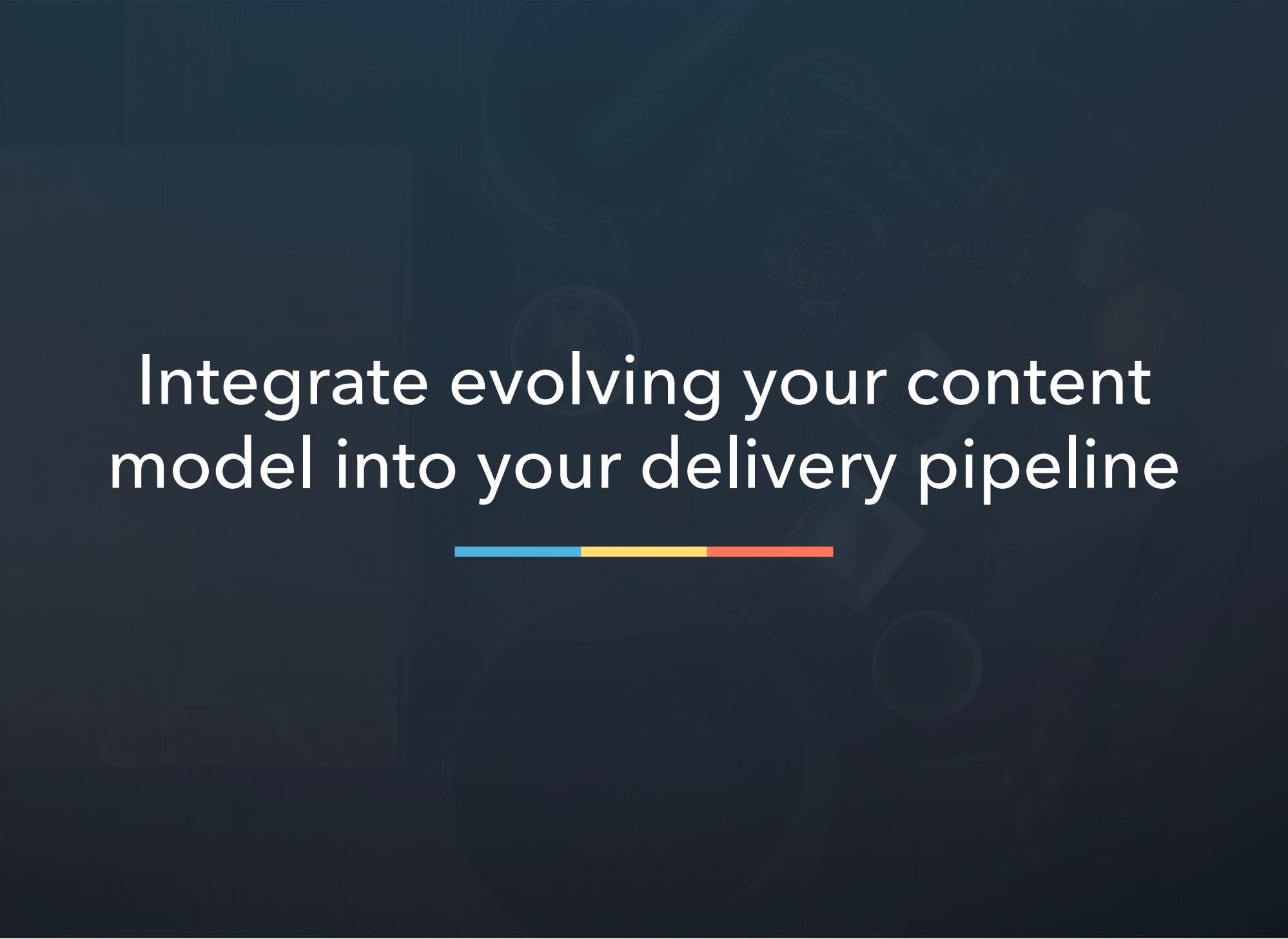
module.exports = {
  runMigration,
  validateVersion
};
```

# CircleCI config.yml - Run Tests

```
- run:  
    name: run tests  
    command: |  
        . venv/bin/activate  
        pytest --environment-id="circletesting_{$CIRCLE_BRANCH}"
```

# Demo

---

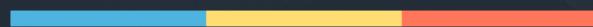


# Integrate evolving your content model into your delivery pipeline

---

Check out the example:

[github.com/contentful-labs/continuous-delivery-environments-example](https://github.com/contentful-labs/continuous-delivery-environments-example)





# Shy Ruparel

---

Twitter: [@ShyRuparel](https://twitter.com/ShyRuparel)

Email: [shy@contentful.com](mailto:shy@contentful.com)

Talks: [GitHub/Shy/Talks](#)