



# Content testing and deployment done the right way

---

By Shy Ruparel



# Shy Ruparel

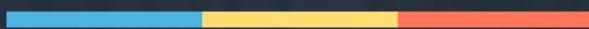
---

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?

**AGAINST MY SUGGESTION, BOSS ASKED  
ME TO MAKE PRODUCTION CHANGE**



**SHIT JUST HIT THE FAN**

# Let's talk 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

```
module.exports = function runMigration(migration) {  
  const post = migration.editContentType("post");  
  post.deleteField("last_appearance");  
  return;  
};
```

# Callable via the Contentful CLI

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

# Programmatic Content model changes

# Advantages

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

# Demo

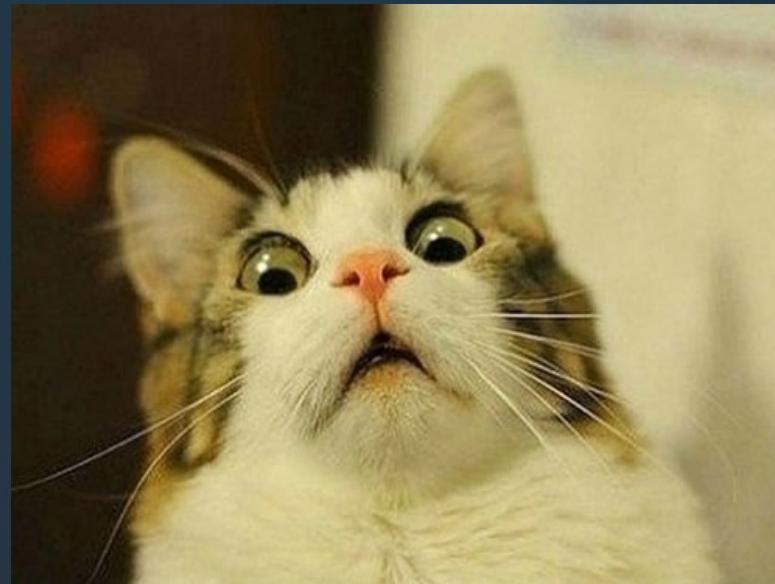
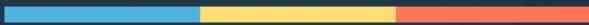
A screenshot of a macOS terminal window. The window title bar shows three colored dots (red, yellow, green) and the path ..ments-example. The main pane displays a command-line session:

```
03:37:47 PM ➜ continuous-delivery-environments-example ➜ master ✘ $ cat migrations/|
```

The bottom status bar indicates the current directory is ~/Documents/Contentful/continuous-delivery-environments-example and the branch is master.

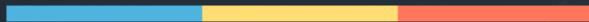
# That's nice

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 the following items: Shy's DevRel Playground, migration\_blog (with master branch), Space home, Content model, Content, Media, Settings (which is currently selected), and a user profile icon.

The main area is titled "Environments". It contains a table with columns: ID, Status, and Actions. There are four environments listed:

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 is a sidebar with sections for "ADD ENVIRONMENT" (containing a blue "Add environment" button) and "DOCUMENTATION" (containing text about environments and a list item). The documentation text reads:

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.

# Initialize the client

```
const contentful = require('contentful');

const client = contentful.createClient({
  space: 'space_id',
  accessToken: 'content_delivery_api_key'
});

client.getEntries();
```

# Initialize the client with environments

```
const contentful = require('contentful');

const client = contentful.createClient({
  space: 'space_id',
  environment: 'environment_id',
  accessToken: 'content_delivery_api_key'
});

client.getEntries();
```

# Combine migrations with environment

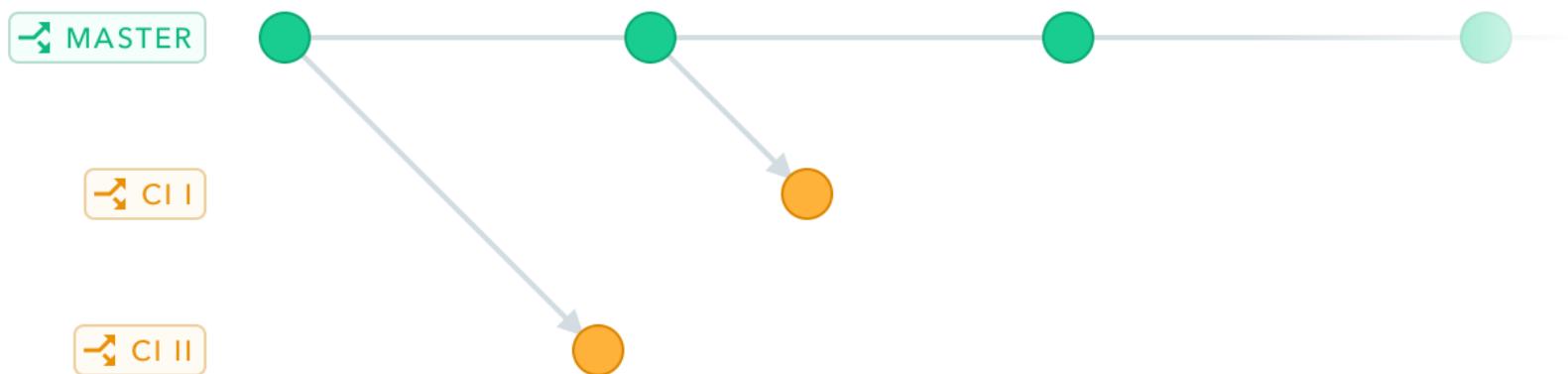
```
contentful space migration
--space-id $YOUR_SPACE_ID
--environment-id $YOUR_ENV_ID
--access-token $CONTENTFUL_MANAGEMENT_ACCESS_TOKEN
migration-demo.js
```

# 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
  - *Create a new environment*
  - *Migrate new environment*
- Test
- Deploy
  - *Migrate master environment*



circleci

# Build & Test

The screenshot shows a build interface for a project named "Shy". The top navigation bar includes "Updates", "Support", and user profile icons. The main area displays a "Test Summary" tab, a "Queue (00:01)" tab, "Artifacts", "Configuration", "Build Timing", and "Build Parameters". A prominent green bar at the top indicates a successful build. Below it, a timeline shows the execution of various build steps:

Step	Description	Duration
0 (00:45)	Initial Setup	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

# 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

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

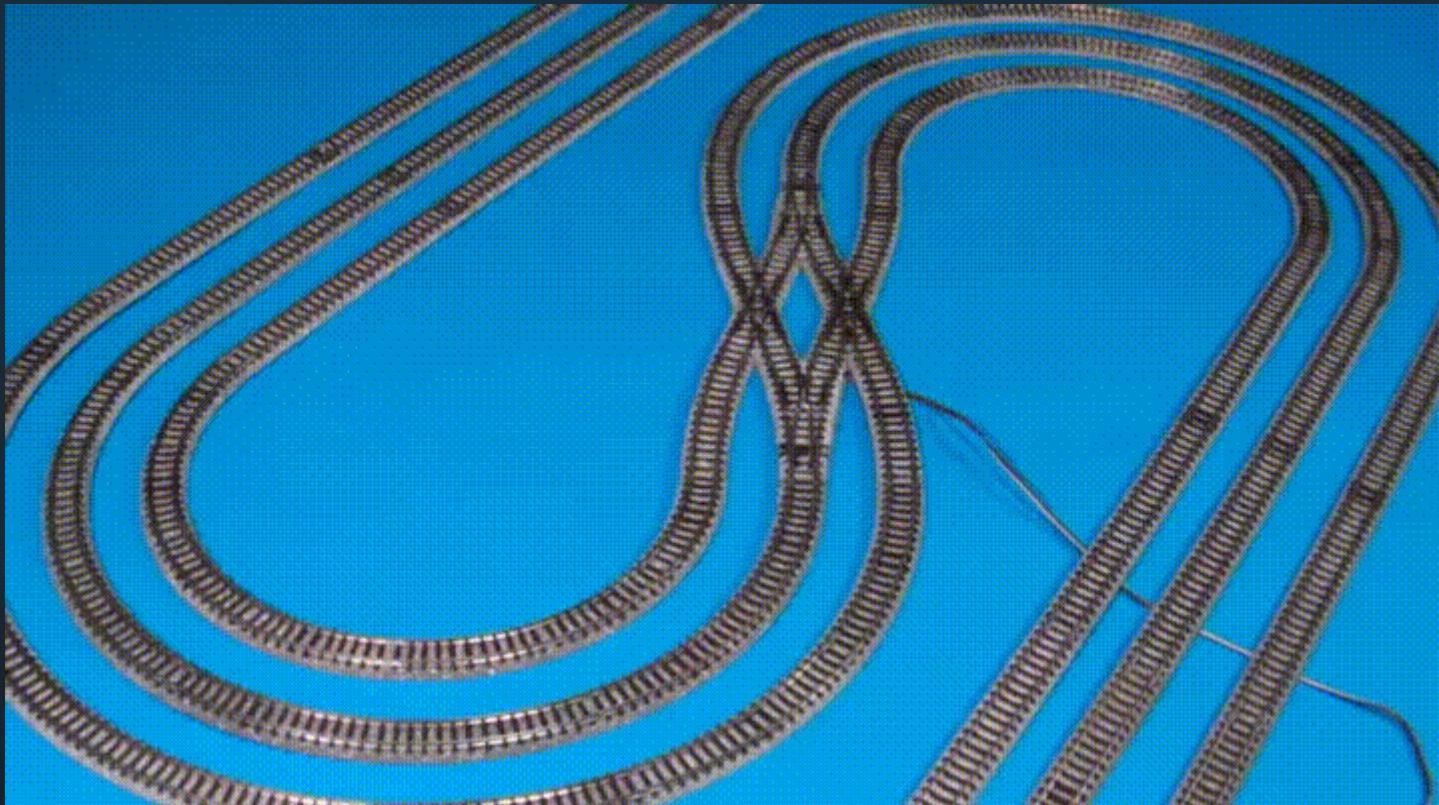
Deploy Master to Heroku 00:17

Set Up Test Summary

Builds WORKFLOWS INSIGHTS ADD PROJECTS TEAM SETTINGS

This screenshot shows a CI/CD pipeline interface for a project named 'migration-env-demo' in the 'Shy' organization. The pipeline has completed successfully, indicated by a green bar at the top. The build summary shows a queue time of 2 seconds and a total duration of 56 seconds. The pipeline steps listed are: Spin up Environment, Checkout code, Restoring Cache, install dependencies, Saving Cache, Updating Master Environment, and Deploy Master to Heroku. The 'Deploy Master to Heroku' step took 17 seconds. On the left, a sidebar provides navigation links for Builds, Workflows, Insights, Add Projects, Team, and Settings.

# Keeping code and content model in sync



# Version Tracking Content Model

The screenshot shows a user interface for managing content models. At the top, there's a dark header bar with the following elements from left to right:

- A logo icon.
- The text "Shy's DevRel Playground".
- A dropdown menu icon (three horizontal lines).
- A "migration\_blog" entry with a "master" branch indicator.
- A "Space home" link with a house icon.
- A "Content model" link with a cube icon.
- A "Content" link with a leaf/cursor icon.
- A "Media" link with a photo album icon.
- A "Settings" link with a gear icon.
- A dropdown arrow indicating more options.

The main content area has a light gray background. On the left, there's a sidebar with a back arrow and a "version\_tracking" title next to a green cube icon. Below it, there are two tabs: "Fields (1)" which is highlighted with a blue underline, and "JSON preview".

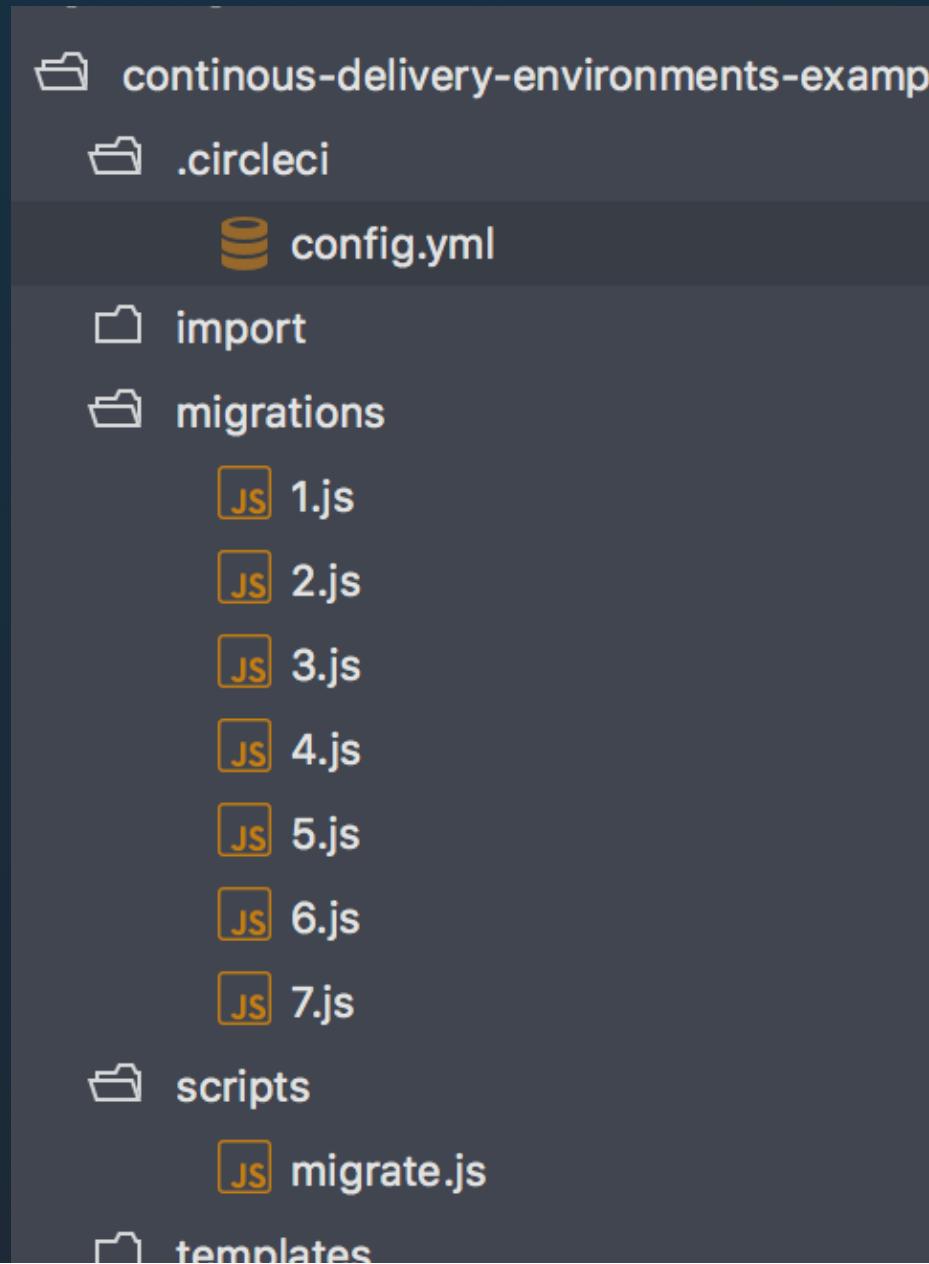
The main panel displays a single field entry:

Ab	<b>version</b>	Short text	Entry title	Settings	...
----	----------------	------------	-------------	----------	-----

# 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 Files



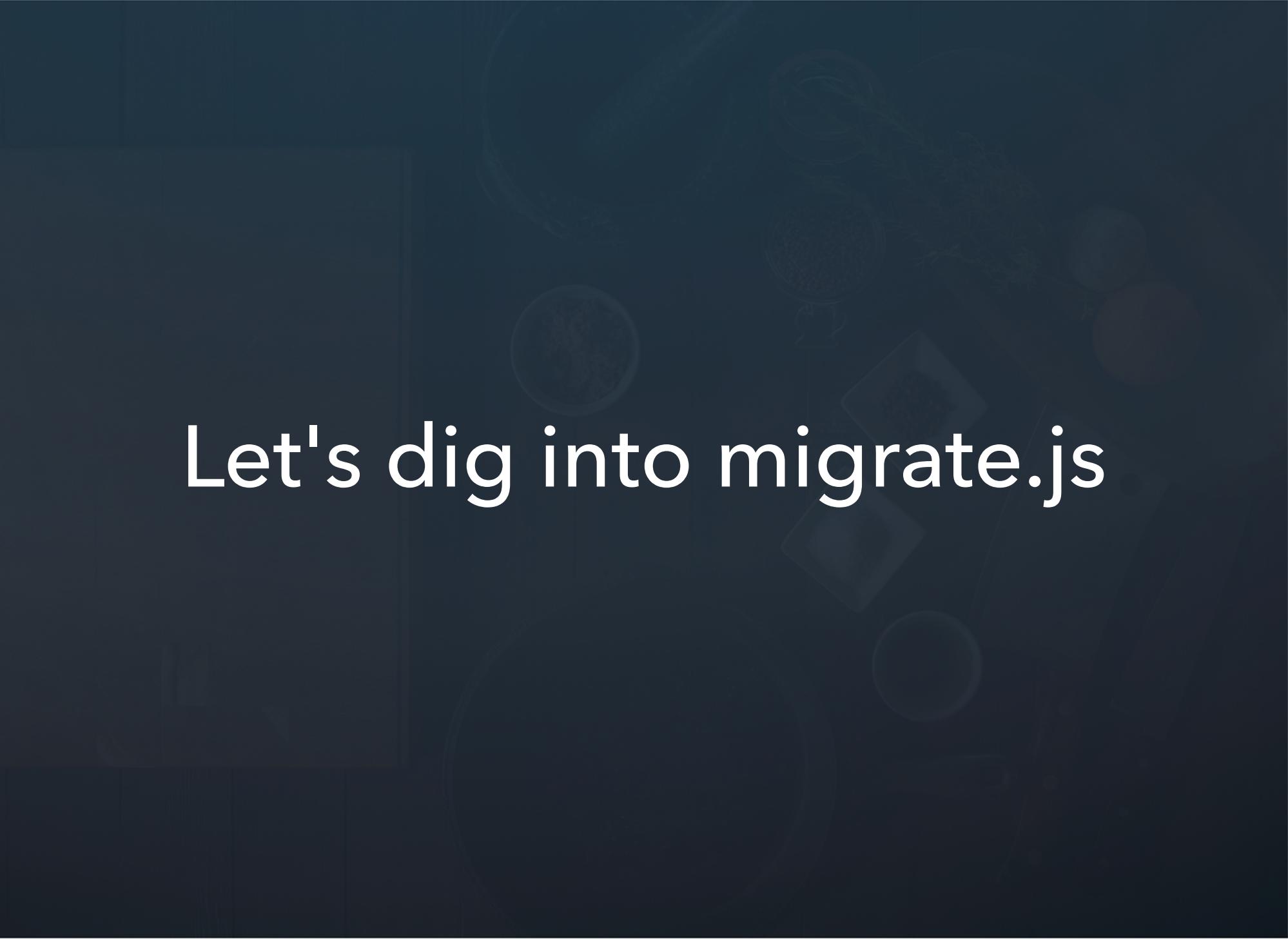


# CircleCI config.yml - Configure Contentful

```
- run:  
  name: Preparing environment for testing  
  command: |  
    . venv/bin/activate  
    scripts/migrate.js $SPACE_ID  
    "CI_${CIRCLE_BRANCH}" $MANAGEMENT_API_KEY
```

# CircleCI config.yml - Update Testing

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



# Let's dig into migrate.js

# Utilize the Contentful CMA

```
const { createClient } = require('contentful-management');

//...//

const client = createClient({
  accessToken: CMA_ACCESS_TOKEN
});
const space = await client.getSpace(SPACE_ID);
```

# Does the environment exist?

```
console.log(`Checking for existing versions of environment: ${ENVIRON  
try {  
  environment = await space.getEnvironment(ENVIRONMENT_ID);  
  if (ENVIRONMENT_ID != 'master') {  
    await environment.delete();  
    console.log('Environment deleted');  
  }  
} catch(e) {  
  console.log('Environment not found');  
}
```

# Create the environment

```
if (ENVIRONMENT_ID != 'master') {
  console.log(`Creating environment ${ENVIRONMENT_ID}`);
  environment = await space.createEnvironmentWithId(ENVIRONMENT_ID,
    { name: ENVIRONMENT_ID });
}
```

# Allow API Key access

```
console.log('Update API Keys to allow access to new environment');
const newEnv = {
  sys: {
    type: 'Link',
    linkType: 'Environment',
    id: ENVIRONMENT_ID
  }
}

const {items: keys} = await space.getApiKeys();
await Promise.all(keys.map(key => {
  console.log(`Updating - ${key.sys.id}`);
  key.environments.push(newEnv);
  return key.update();
}));
```

# Run needed Migrations

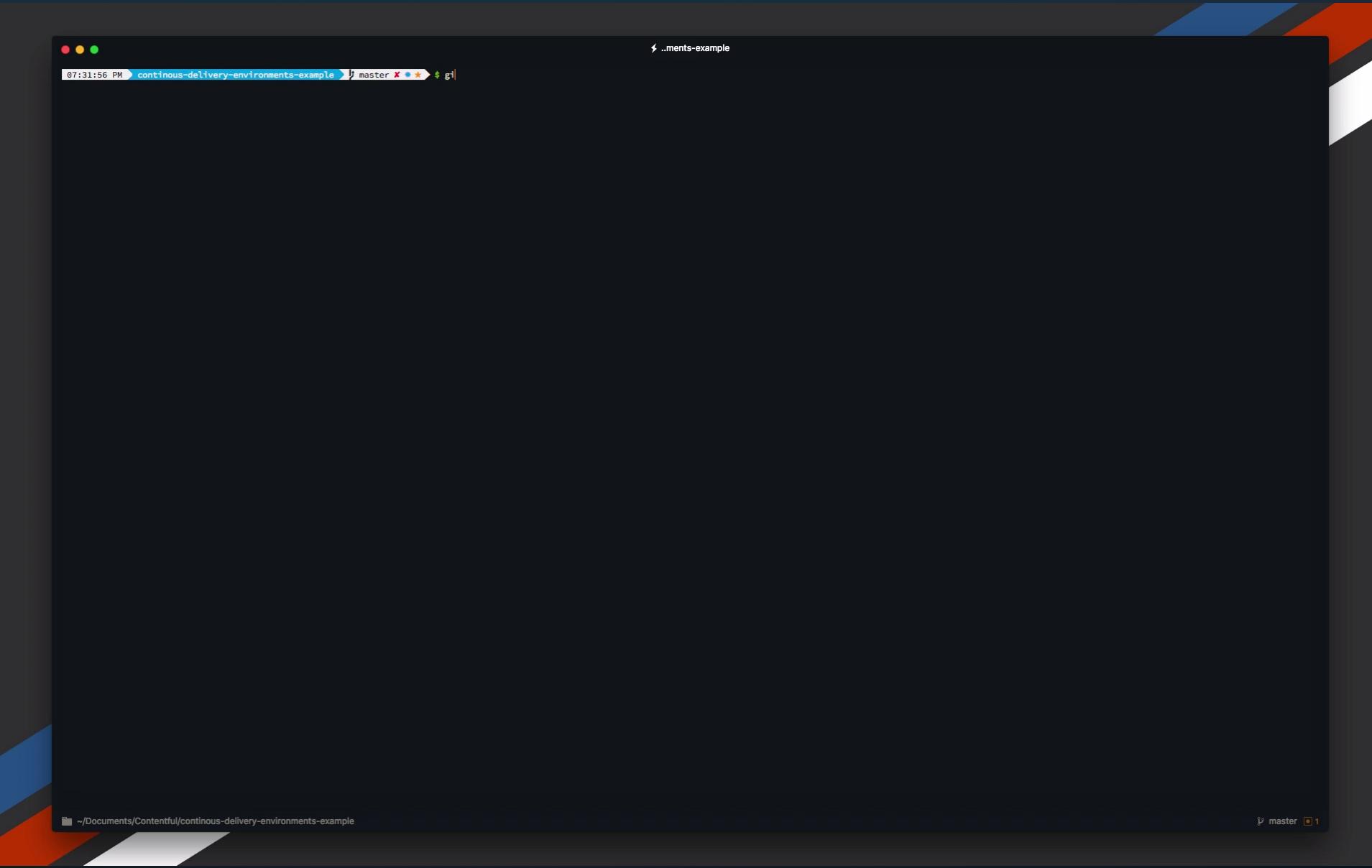
```
console.log('Run migrations and update version entry');
while(migrationToRun = migrationsToRun.shift()) {
  const filePath = path.join(__dirname, '..', 'migrations', getFileOf
  console.log(`Running ${filePath}`);
  await runMigration(Object.assign(migrationOptions, {
    filePath
  }));
  console.log(`${migrationToRun} succeeded`);

  storedVersionEntry.fields.version[defaultLocale] = migrationToRun;
  storedVersionEntry = await storedVersionEntry.update();
  storedVersionEntry = await storedVersionEntry.publish();

  console.log(`Updated version entry to ${migrationToRun}`);
}

console.log('All done!');
```

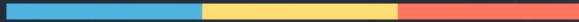
# Demo



```
07:31:56 PM ~continuous-delivery-environments-example master ✘ git
```

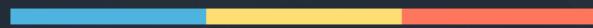
⚡ ..ments-example

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