# Contentful CMS
# Best Practices

Managing Environments & New Content

# Audience & Space Roles

**Administrator** - has unlimited access to all content and actions in the space.

**Editor** - can create, edit, publish, archive and delete all content in a space.

**Author** - can create and edit all content in a space, but cannot publish it.

**Translator** - can edit content in a selected locale.

**Freelancer** - can create content and edit and delete content created by them. Cannot publish content.

Anyone with the **Admin** role will be able to create, manage, and alias environments

Anyone with an **Editor** or **Author** role will be able to edit content in multiple environments

○ Admin
Manages everything in the space

○ Other roles

☐ Editor

☐ Author

☐ Freelancer

☐ Translator

# Purpose

These best practices are outlined for the purpose of managing development of new content and releasing it to production via Contentful. Creating a workflow for developers & content editors to safely make changes and go through comprehensive testing before changes are live. Ensuring a rollback environment is available.

# Environment Alias

An environment alias allows you to access and modify the data of an environment, called the target environment, through a different static identifier.

The default alias is `master`. This alias cannot be renamed or deleted & is connected to **production**. We also have a `develop` alias it is connected to the **develop site** for testing.
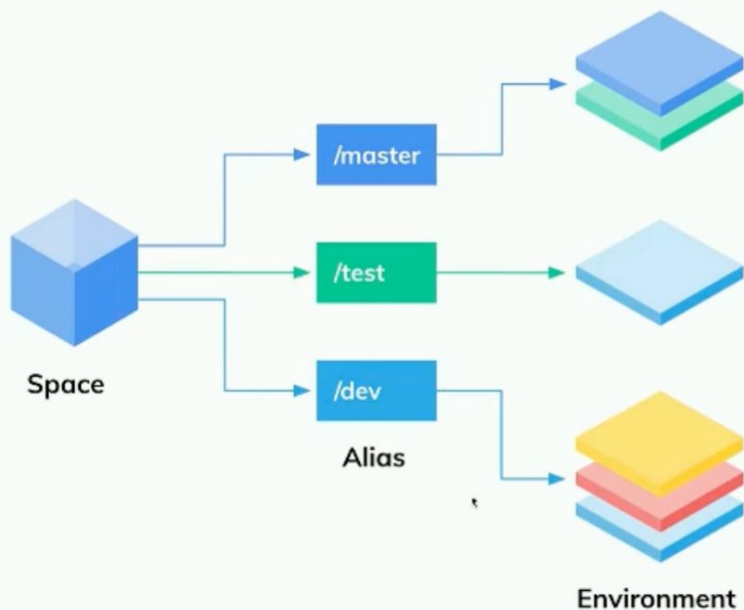
# Target Environments

**Target Environments** are sandbox environments that have been promoted to an environment alias.

*NOTE* →

Aliases and target environments can be used to enhance your development and deployment workflows. For example, you can safely apply content model changes and roll them back when necessary or use them in your CI/CD pipelines.

# Environment Aliases

Iterate and launch with confidence



An environment alias is a **static id** that **controls access to an environment**.

Environment aliases **can point to any environment in Contentful**.

The target environment **can then be updated at will** via the CMA, CLI tools or the Web App.

Using environment aliases enables **risk-free releases** and **easy fallback** as you update and change your content model.

# Master Environment

Master environment is used as a production environment to deliver content to the end users.
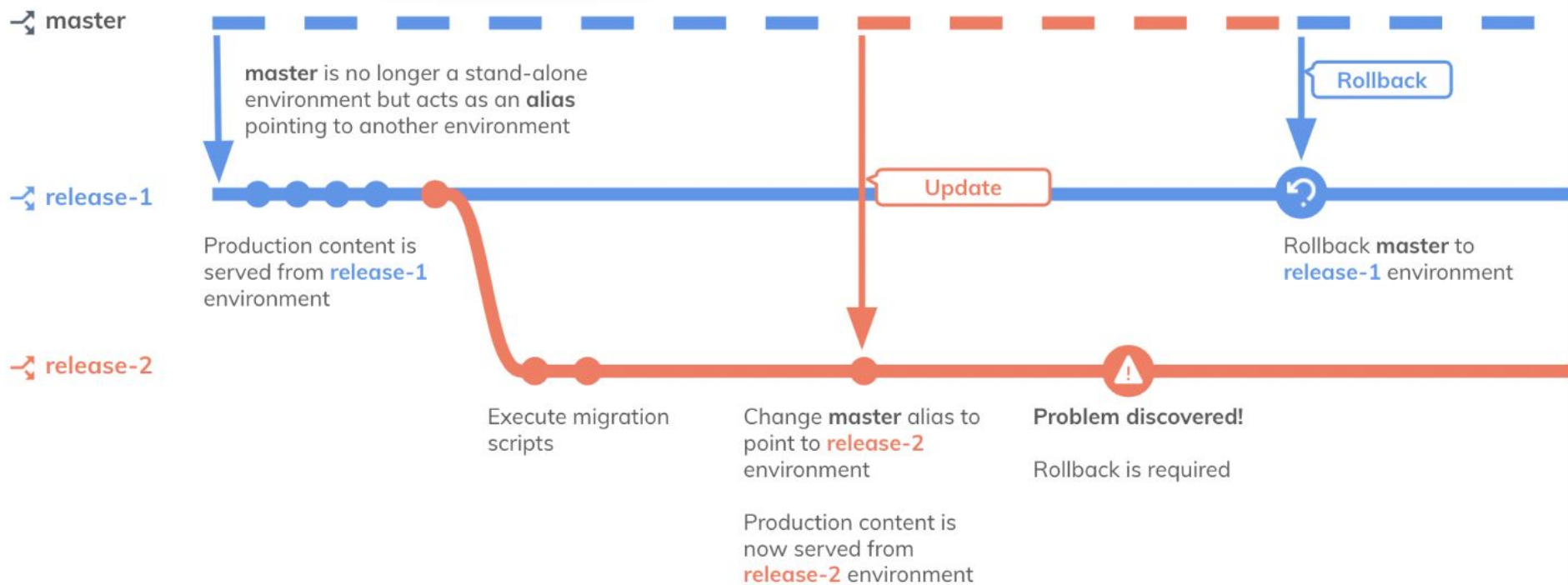
# Sandbox environments

Sandbox environments are used for non-production development and testing.

**NOTE →**

Each environment supports isolated editing. Content edited on the master environment or master's target **WILL NOT** be reflected on a sandbox environment cloned before the updates, and visa versa. Once an environment is created for release **MASTER** should be frozen to all updates.

# Environment aliases

Enabling risk free releases and instant rollbacks

**master**

**master** is no longer a stand-alone environment but acts as an **alias** pointing to another environment

Rollback

**release-1**

Production content is served from **release-1** environment

Update

Rollback **master** to **release-1** environment

**release-2**

Execute migration scripts

Change **master** alias to point to **release-2** environment

Production content is now served from **release-2** environment

**Problem discovered!**

Rollback is required

# Environment Best Practices

Rapidly iterate your approach without creating unintended consequences

A common **misconception** is that Contentful environments should map to development environments.

Though this is within the capabilities of the platform, **it is often an anti-pattern.**

**Environments and Aliases** are meant to help teams manage space iteration. They should be short-lived and disposable.

**Environments give teams an isolated and secure "sandbox"** to edit, update, and delete space configurations before choosing to promote them to production.

Using environments any other way **bypasses the need for a content platform that enables authors** to manage content experiences without depending on technical resources.

## Environments

ENVIRONMENT ALIASES

### master                                                                    Delete

master-2022-08-19 ⧉          Current branch          Change alias target
                              aliased to prod

### develop                                                                   Delete

dev-08-25 ⧉                                           Change alias target

                              Current branch
                              aliased to develop

ENVIRONMENTS

| Environment ID | Created | Status | |
|---|---|---|---|
| master-2022-08-19 ⧉ *Aliased to* master | 2 months ago | READY | Delete |
| dev-08-25 ⧉ *Aliased to* develop | | | Delete |
| dev-08-19 ⧉ | 3 minutes ago | READY | Delete |

All development of new blocks (content models),
pages, or new contentshould be completed & tested here.
On release date this branch will be aliased to master

old branch can be rolled
back to by aliasing it to
either enviorment

# Creating a new environment

1.  Go to settings > environments > add new env

ENVIRONMENT ALIAS USAGE

You are using 2 out of 3 environment aliases
available in this space.

Add environment alias

ENVIRONMENT USAGE

You are using 3 out of 11 environments available
in this space. ⓘ

Add environment

DOCUMENTATION

Environment Aliases

An environment alias allows you to access and
modify the data of an environment through a
different static identifier.
Read our **environment alias documentation** ⧉
for more information.

Environment

Environments allow you to develop and test
changes to data in isolation.
See the **Contentful domain model** ⧉ for
details.
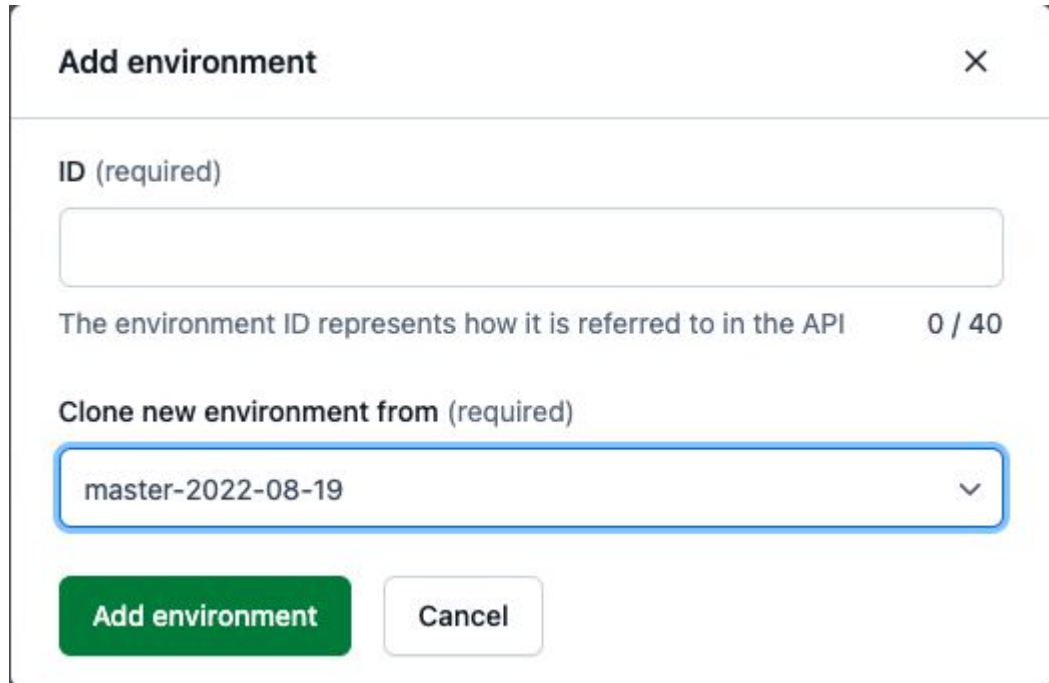
# Creating a new environment

2. Select the env you would like to clone and title it, can be formatted as the branch name or feature name + a release date.

EX: *develop-10-24-22*

*blog-refactor-09-14-22*

**Add environment**                                              ✕

ID (required)

The environment ID represents how it is referred to in the API          0 / 40

**Clone new environment from** (required)

master-2022-08-19                                                    ⌄

**Add environment**          Cancel
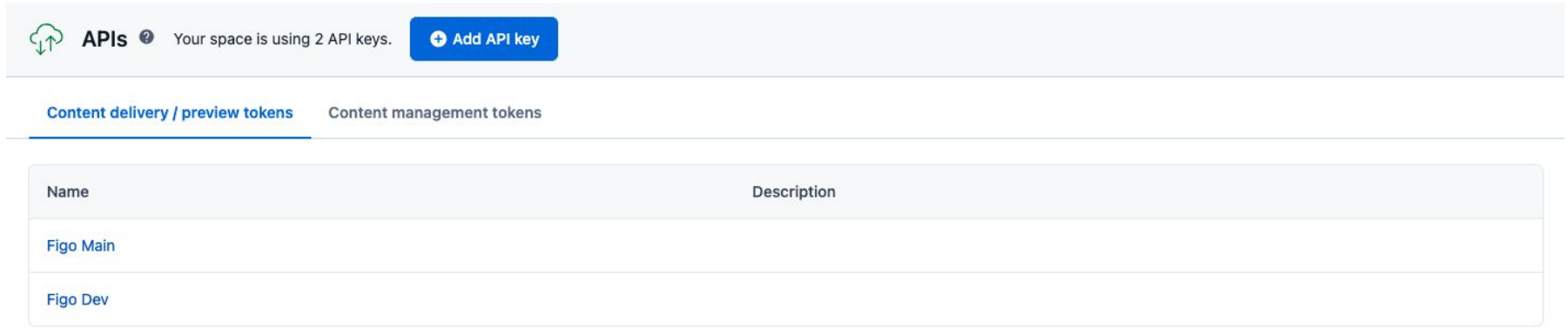
# Target Environments & Release

3. After you have created an env, you are able to set it as the target environment for any alias. This allows that work to be viewed on production or develop, respectively. Select "change alias target" and select your new branch or the branch you would like to release to master.
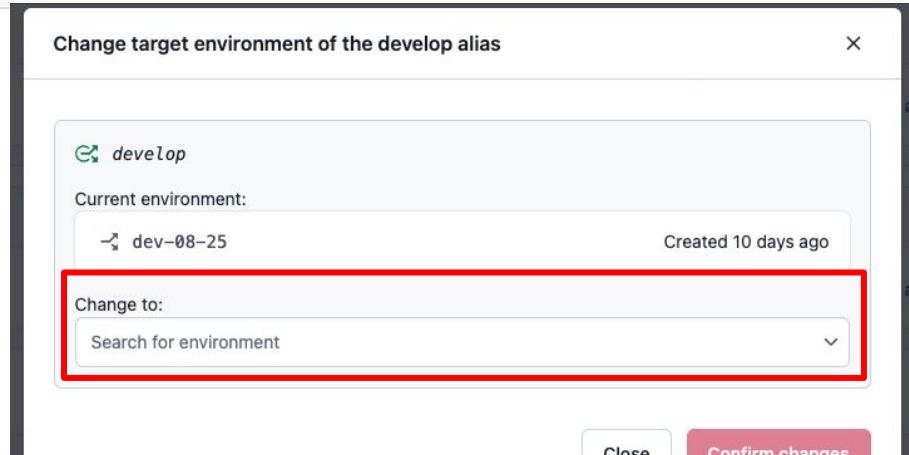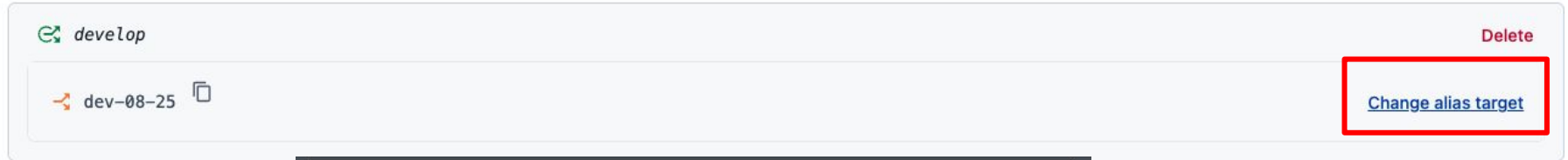
# API KEYS

After you have updated the target environment, on the master environment, go to Settings > API Keys



Select the appropriate environment and verify that the branch you set as a target environment is selected for that API key.
EX: You want to release *dev-08-15* so you set it as the target env for master, then you go to **Figo Main** to verify that branch is selected to use the main API keys.

# Deleting Environments

Contentful environments are meant to be deleted. A Space is limited to having 11 environment branches at a time.

After an env branch has been demoted from target branch, it should remain in the space through the next release in case of any roll back needs. After the env branch is two sprints stale it may safely be deleted.

| Environment ID | | Created | Status | |
|---|---|---|---|---|
| ⤳ master-2022-08-19 ▢ | *Aliased to* ⤢ *master* | 2 months ago | READY | Delete |
| ⤳ dev-08-25 ▢ | *Aliased to* ⤢ *develop* | 13 days ago | READY | Delete |
| ⤳ dev-08-19 ▢ | | 13 days ago | READY | Delete |

# Overview

+ The ***Master environment*** alias should only be updated for content hotfixes and otherwise should be 🥶frozen🥶 to any changes, until the next release where a new target environment is assigned.
+ Any new features, pages, or blocks should be created in a sandbox environment targeted at ***develop***, and tested on the deployed develop site. After all development and testing is complete, it should be assigned as a target environment of master.
+ After a release, the previous target branch should not be deleted in case a rollback needs to occur after final QA. Only **after** it is stale by two or three releases should it be permanently deleted!