

# Merge Requests

---

# Merge requests

*Merge requests are how you integrate separate changes that you've made in a branch to a project.*

They allow you to **visualize and collaborate** on the **proposed changes** to source code that exist as commits **on a given Git branch**.

Before you start, you should have already **created a branch and pushed** your changes to GitLab.

Christophe Villeger > Symfony Demo > Details

You pushed to `feature/update-readme` at [Christophe Villeger / Symfony Demo](#) 4 minutes ago

Create merge request

S

**Symfony Demo** 🌐

Project ID: 10152705



☆ Star

0

🔗 Fork

1

Clone ▾

# Merge requests

At a minimum, add a title and a description to your merge request. Optionally, select a user to review your merge request. You may also select a milestone and labels.

## New Merge Request

From `feature/update-readme` into `master` [Change branches](#)

Title

Start the title with `WIP:` to prevent a **Work In Progress** merge request from being merged before it's ready.

Add [description templates](#) to help your contributors communicate effectively!

Description

WritePreview

**B** *I* “ ” < />

## About

This add a single docker instruction to use the container shell.

Markdown and quick actions are supported [Attach a file](#)

Assignee

Milestone

Labels

# Merge requests

From there, you have the option to select the source branch and the target branch you'd like to compare to. The default target project is the upstream repository, but you can choose to compare across any of its forks.

Merge request dependencies

Enter merge request URLs or references (e.g. path/to/project!merge\_request\_id)

List the merge requests that must be merged before this one.

Approval rules

Approvers

No. approvals required

All members with Developer role or higher and code owners (if any)

1

Edit

Add approval rule

Tip: add a CODEOWNERS to automatically add approvers based on file paths and file types.

Source branch

feature/update-readme

Target branch

master

Change branches

☐ Delete source branch when merge request is accepted.

☐ Squash commits when merge request is accepted. ?


Submit merge request

Cancel

Commits 1

Changes 1


03 Nov, 2019 1 commit



Add shell in container instruction in README

Christophe Villeger authored 5 minutes ago

55f0f1e3



# Merge requests

Gather feedback from  
your team and inspect  
your GitLab CI.

## [feat] Add shell in container instruction in README

### About

This add a single docker instruction to use the container shell.



Request to merge `feature/update-readme` into `master`

Open in Web IDE

Check out branch



Requires approval.



Merge

You can only merge once this merge request is approved.

You can merge this merge request manually using the [command line](#)



0



0



Discussion 0

Commits 1

Changes 1



Changes between

latest version

and

master

1 File

5

0



README.md

5 0



Show all lines

```
26 26 docker-compose exec --user=application web composer install
27 27 docker-compose exec --user=application web php bin/console doctrine:schema:update --force
28 28 ***
29 +
30 + ### Open shell in container
31 + ```bash
32 + docker-compose exec --user=application web bash
33 + ```
```

# Protected Branches

---

# Protected Branches

To protect a branch, you need to have at least Maintainer permission level. Navigate to your project's Settings→Repository

Scroll to find the Protected branches section. From the Branch dropdown menu, select the branch you want to protect and click Protect.

## Protected Branches

Keep stable branches secure and force developers to use merge requests.

By default, protected branches are designed to:

- prevent their creation, if not already created, from everybody except Maintainers
- prevent pushes from everybody except Maintainers
- prevent **anyone** from force pushing to the branch
- prevent **anyone** from deleting the branch

Protect a branch

Branch:

Select branch or create wildcard

▼

Wildcards such as `*--stable` or `production/*` are supported

Allowed to merge:

Select

▼

Allowed to push:

Select

▼

Only groups that [have this project shared](#) can be added here

Require approval from code owners:

☒

Pushes that change filenames matched by the CODEOWNERS file will be rejected

Protect

# Gitlab Board

---



# Milestone

Permet de regrouper les issues/MR entre elles afin de voir l'objectif global du “jalon” pour une deadline.

Un milestone correspond à:

- Sprint
- Release

The screenshot shows the GitLab interface with the 'Milestones' tab selected. The left sidebar contains navigation links: Project, Repository, Issues (105), List, Boards, Labels, Service Desk, Milestones, Merge Requests (7), CI / CD, Operations, Registry, Snippets, and Members. The main content area displays a list of milestones for the 'charts' project. The milestones are grouped by status: Open (8), Closed (36), and All (44). The 'Open' group is selected, showing a list of milestones with their respective progress bars and completion percentages. The milestones are as follows:

Milestone	Issues	Merge Requests	Completion	Action
0.0.37 Jul 17, 2018–Jul 24, 2018	11	10	61% complete	Close Milestone
0.0.38 Jul 24, 2018–Jul 31, 2018 Upcoming	0	0	0% complete	Close Milestone
Week 39 Jul 31, 2018–Aug 7, 2018 Upcoming	0	0	0% complete	Close Milestone
0.0.39 Jul 31, 2018–Aug 7, 2018 Upcoming	0	0	0% complete	Close Milestone
0.0.40 Aug 7, 2018–Aug 14, 2018 Upcoming	0	0	0% complete	Close Milestone
0.0.41 Aug 14, 2018–Aug 21, 2018 Upcoming	0	0	0% complete	Close Milestone
0.0.42 Aug 21, 2018–Aug 28, 2018 Upcoming	0	0	0% complete	Close Milestone
Backlog	10	0	50% complete	Close Milestone

# Labels

Permet de regrouper les issues/MR en catégories. Facilite les recherches.

<b>devops::configure</b> ?	Issues for the Configure stage of the DevOps lifecycle (e.g. Auto DevOps , Kubernetes Configuration, ChatOps, Serverless, PaaS) <a href="#">Issues</a> · <a href="#">Merge requests</a>	Group label ☆ ✎ ⋮ <input type="button" value="Subscribe"/>
<b>devops::create</b> ?	Issues for the Create stage of the DevOps lifecycle (e.g. Source Code Management, Design Management, Web IDE, Code Review, Gitaly, Gitter) <a href="#">Issues</a> · <a href="#">Merge requests</a>	Group label ☆ ✎ ⋮ <input type="button" value="Subscribe"/>
<b>devops::defend</b> ?	Issues for the Defend stage of the DevOps lifecycle (e.g. Runtime Application Self Protection, Threat Detection, Data Loss Prevention) <a href="#">Issues</a> · <a href="#">Merge requests</a>	Group label ☆ ✎ ⋮ <input type="button" value="Subscribe"/>
<b>devops::enablement</b> ?	Issues for the Enablement stage of the DevOps lifecycle (eg. Fulfillment, Geo, Memory, Ecosystem) <a href="#">Issues</a> · <a href="#">Merge requests</a>	Group label ☆ ✎ ⋮ <input type="button" value="Subscribe"/>
<b>devops::growth</b> ?	Issues for the Growth stage of the DevOps lifecycle (eg. Activation, Adoption, Upsell, Retention) <a href="#">Issues</a> · <a href="#">Merge requests</a>	Group label ☆ ✎ ⋮ <input type="button" value="Subscribe"/>
<b>devops::manage</b> ?	Issues for the Manage stage of the DevOps lifecycle (e.g. Audit Management, Cycle Analytics) <a href="#">Issues</a> · <a href="#">Merge requests</a>	Group label ☆ ✎ ⋮ <input type="button" value="Subscribe"/>

# Create an issue

## New Issue

Title

My Issue Title

Add [description templates](#) to help your contributors communicate effectively!

Description

Write Preview

**B** *I* **”** **</>** **≡** **≡** **✓** **✕**

This is my issue's description, written in markdown (GitLab Flavored Markdown).

**This is an** **<h3>**

*Let's quote someone here*

Add a task list:

☐ Task 1

☐ Task 2

Mention merge requests ([gitlab-org/gitlab-ee!1784](#)) and issues ([gitlab-org/gitlab-ee#2101](#)) and hover over them to see their titles.

☐ This issue is confidential and should only be visible to team members with at least Reporter access.

Assignee

Unassigned

Assign to me

Due date

2017-05-30

Milestone

Milestone

Labels

P1

Weight

7

Submit issue

Cancel

# Board

Development ▾

Search or filter results...

Edit board

Add list ▾

Add issues

⛶

▾ Backlog0 +

Welcome to your Issue Board!

Add the following default lists to your Issue Board with one click:

To Do

Doing

Starting out with the default set of lists will get you right on the way to making the most of your board.

Add default lists

Nevermind, I'll use my own

▾ Closed0

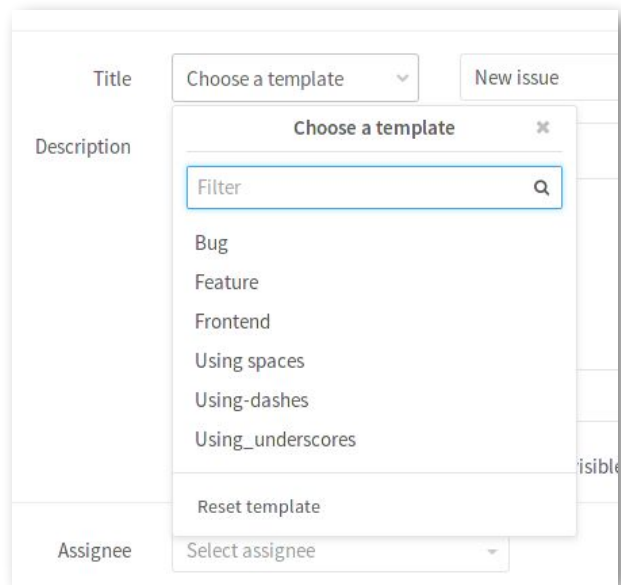
# Templates

---

# Merge request / Issue template

## Fichiers markdown (.md)

- Merge Request
  - Dans le dossier `.gitlab/merge_request_templates/`
- Issues
  - Dans le dossier `.gitlab/issue_templates/`
- Collection
  - [Awesome-github-templates](#)
  - [Symfony](#)
  - [talater.com](#)



The screenshot shows the 'New issue' form in GitLab. The 'Title' field has a dropdown menu open, displaying 'Choose a template'. Below this, the 'Description' field is visible. A search bar with the placeholder 'Filter' and a magnifying glass icon is present. Below the search bar, a list of templates is shown: 'Bug', 'Feature', 'Frontend', 'Using spaces', 'Using dashes', and 'Using underscores'. At the bottom of the dropdown, there is a 'Reset template' link. The 'Assignee' field at the bottom has a dropdown menu with 'Select assignee'.

# CODE\_OF\_CONDUCT.md

- Un code de conduite définit des normes sur la manière de s'engager dans une communauté.
- Il signale un environnement inclusif qui respecte toutes les contributions.
- Il décrit également les procédures à suivre pour résoudre les problèmes entre les membres de la communauté de votre projet.

Template :

- [Contributor-covenant.org](https://contributor-covenant.org)
- [citizencodeofconduct.org](https://citizencodeofconduct.org)

# CONTRIBUTING.md

Des guidelines pour indiquer comment contribuer à votre projet.

Pour aider les contributeurs à faire du bon travail.

Exemples:

- <https://github.com/atom/atom/blob/master/CONTRIBUTING.md>
- <https://github.com/github/opensource.guide/blob/master/CONTRIBUTING.md>
- <https://github.com/rails/rails/blob/master/CONTRIBUTING.md>



# Heroku

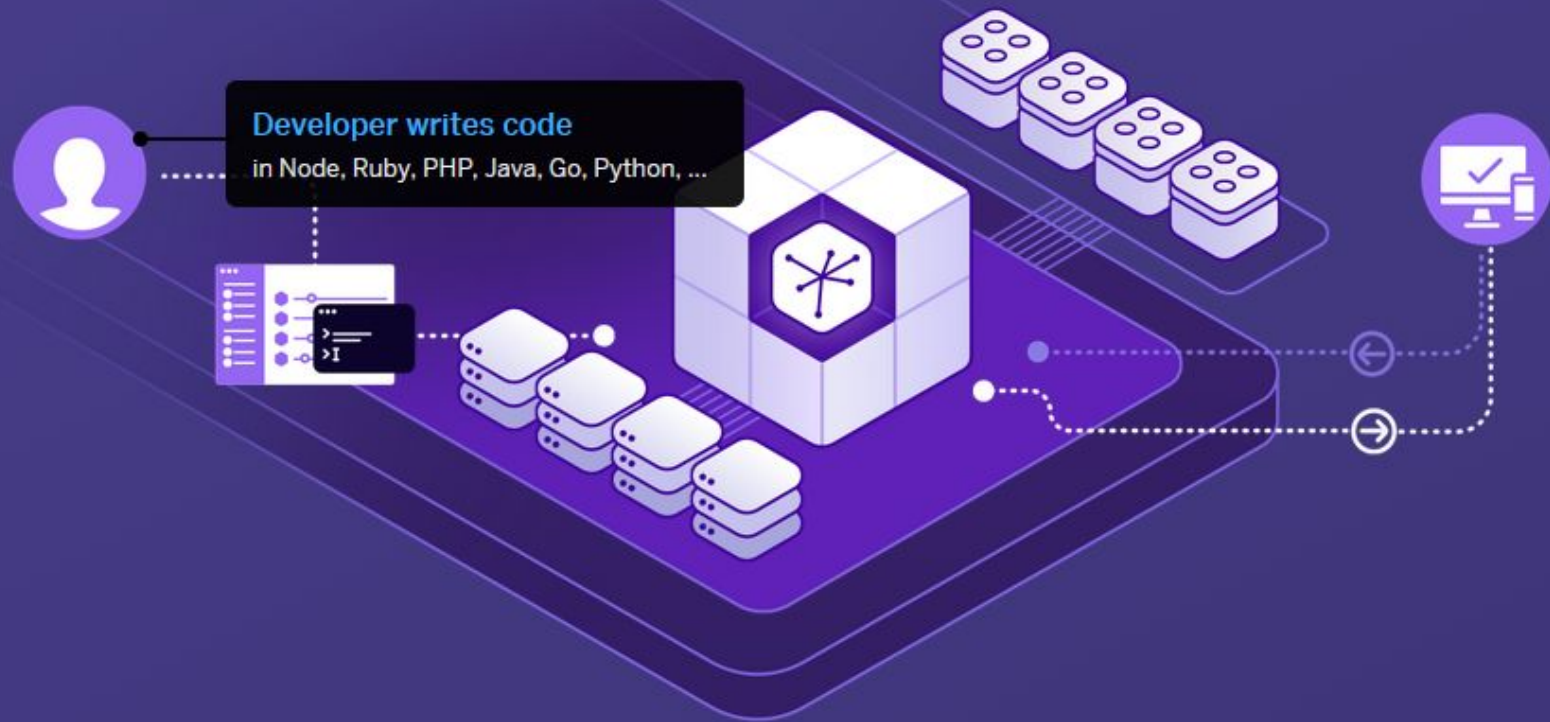
---

# Heroku

*Heroku is a **platform as a service** based on a **managed container system**, with integrated data services and a powerful ecosystem, for **deploying and running modern apps**. The Heroku developer experience is an **app-centric approach** for software delivery, integrated with today's most popular developer tools and **workflows**.*



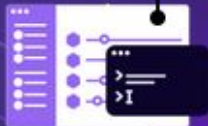
# Heroku



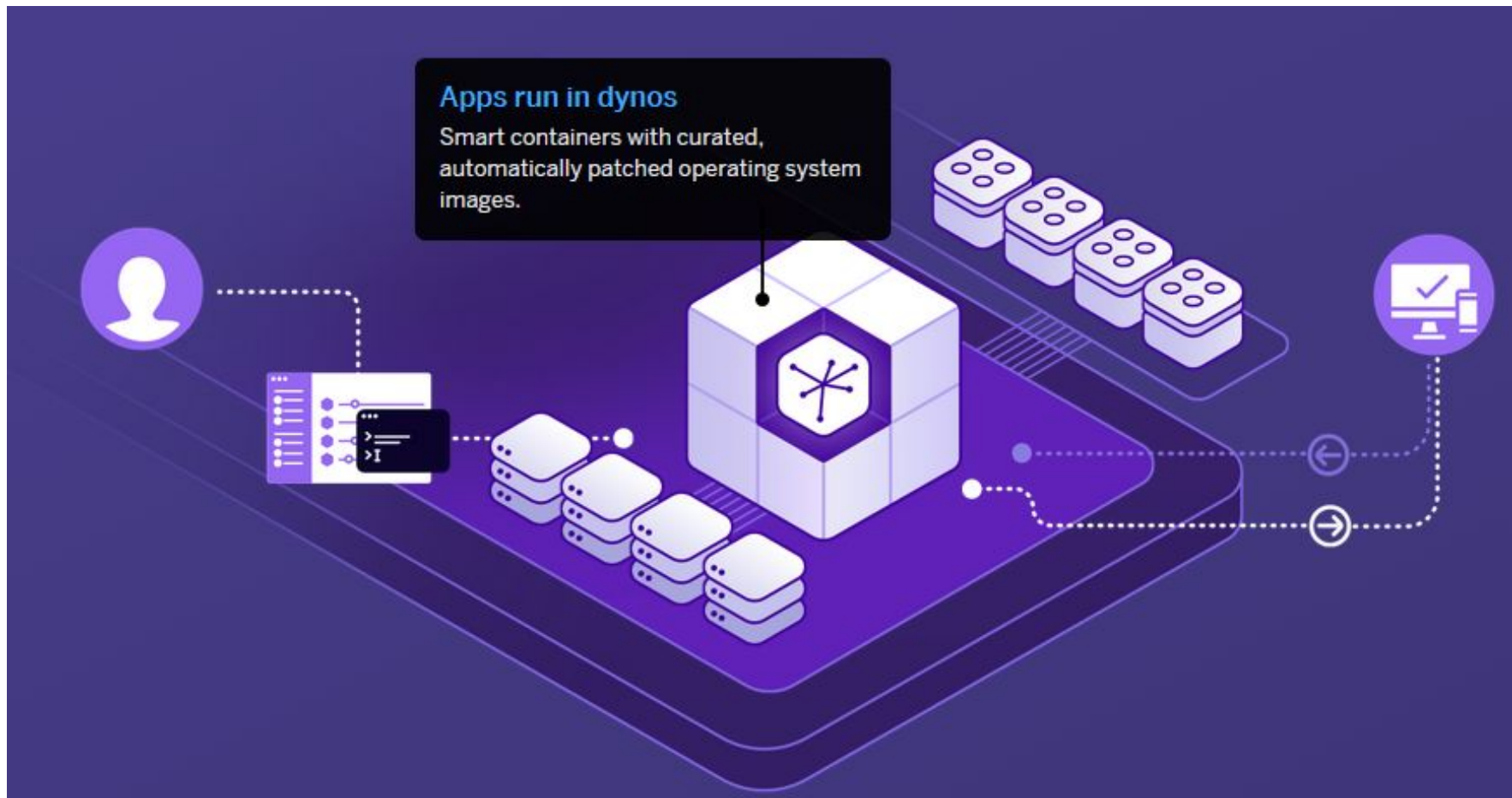
# Heroku

Uses the CLI, GitHub, or  
continuous integration

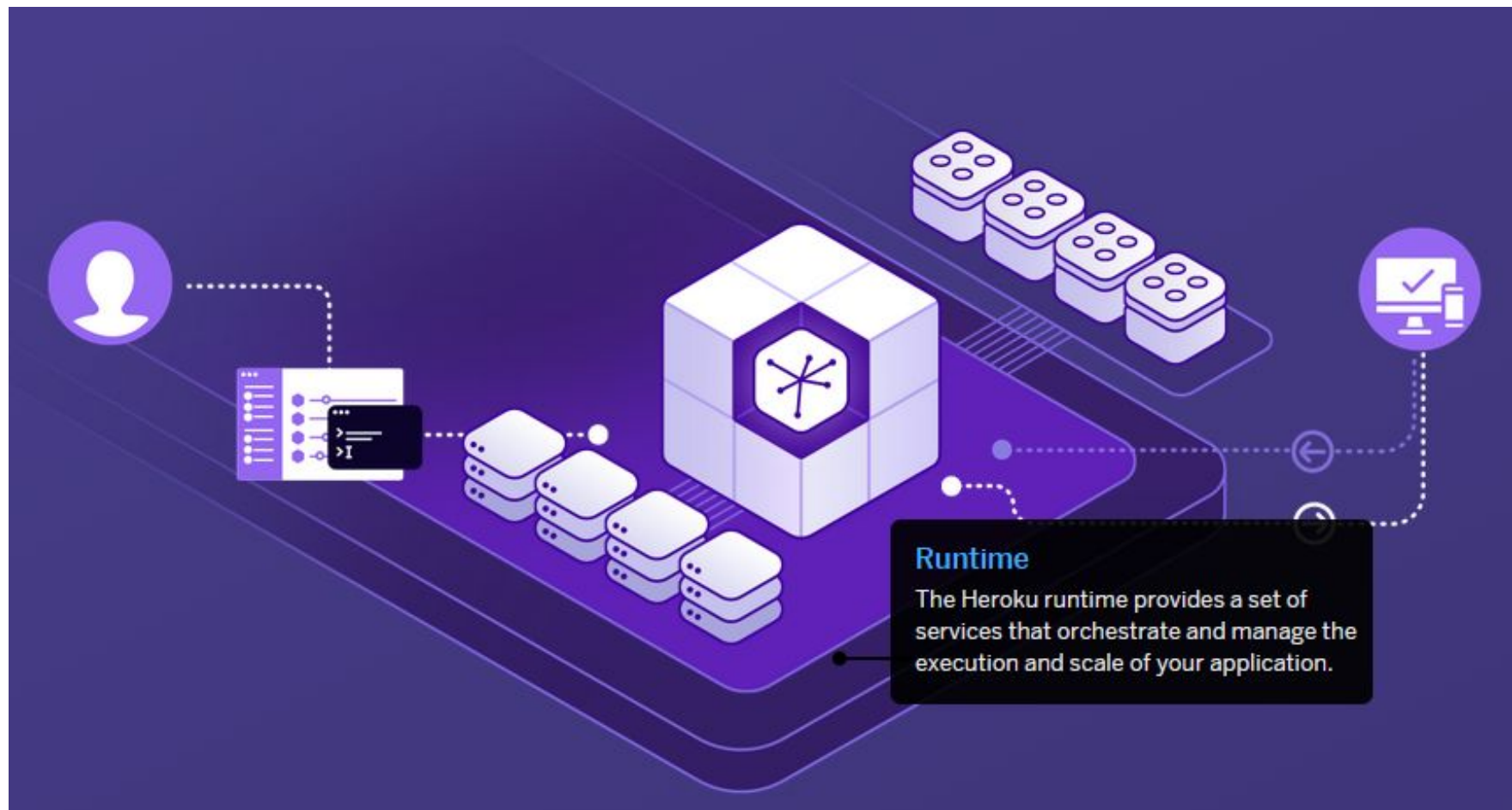
to push code to a build system,  
which produces an app that's  
ready for execution.



# Heroku

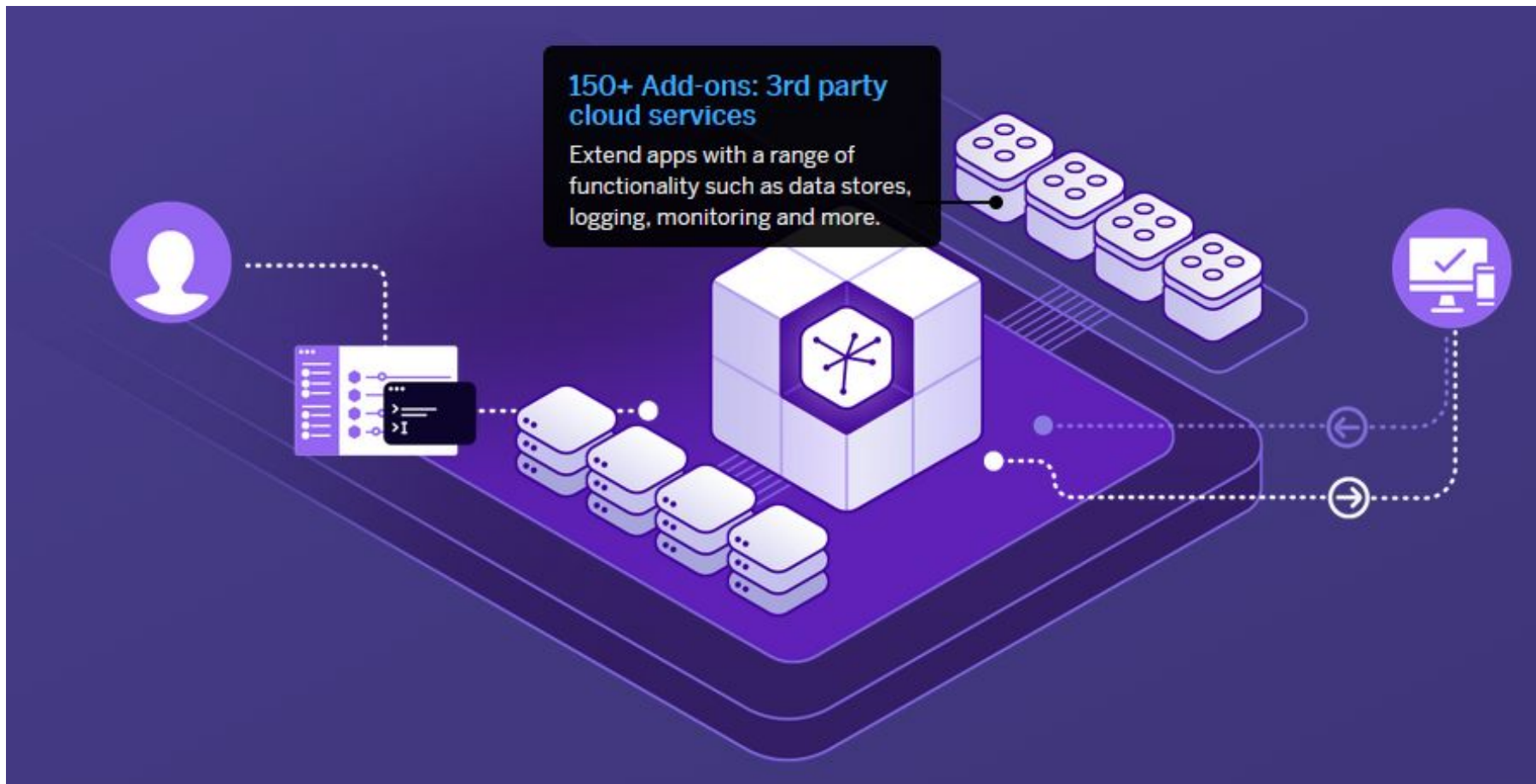


# Heroku

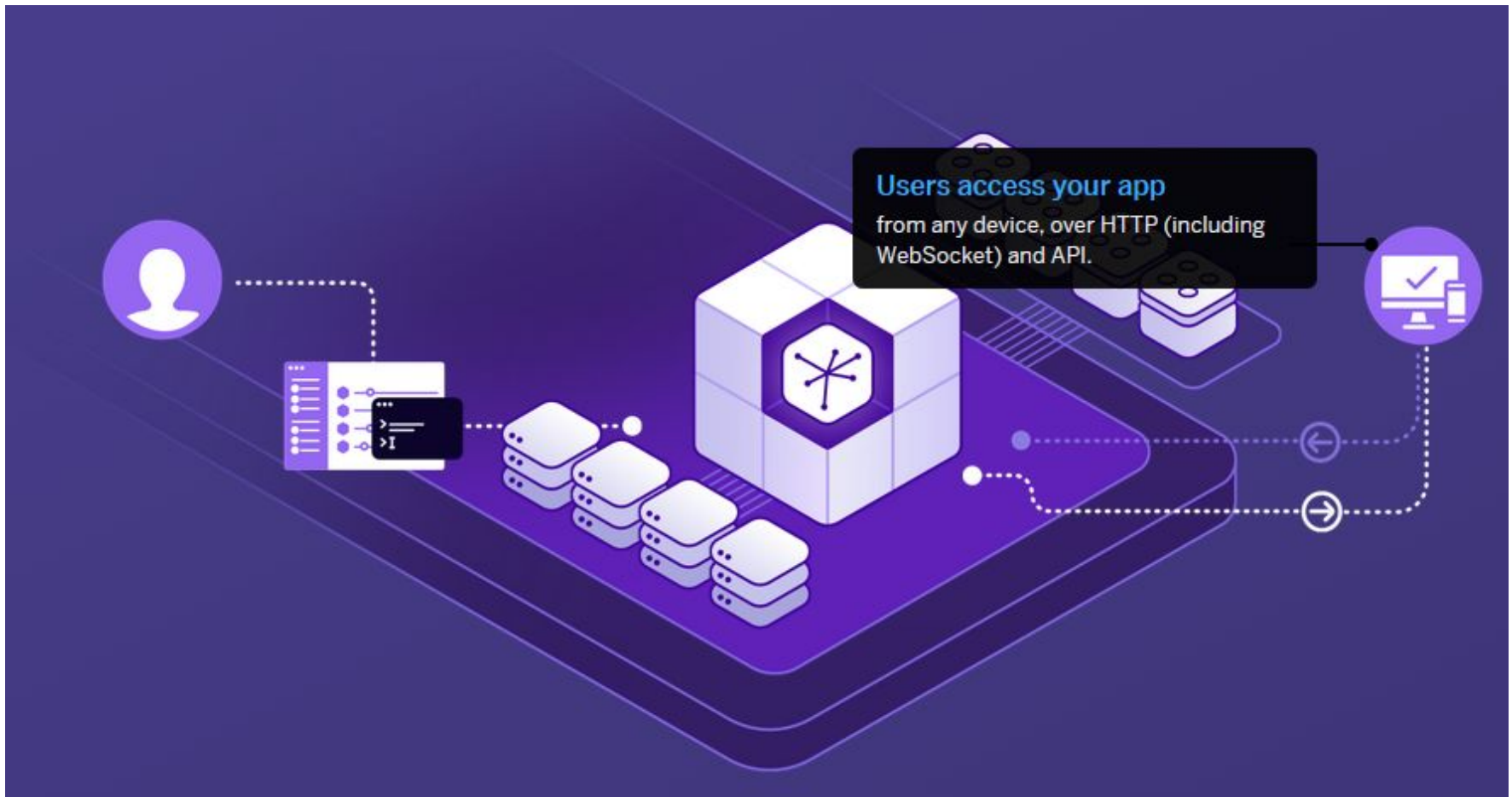




# Heroku



# Heroku





## Create New App

App name

this42app is available

Choose a region

 Europe 

 Add to pipeline...

Create app

# Heroku

Heroku apps include a **Procfile** that specifies **the commands that are executed by the app on startup**. You can use a Procfile to declare a **variety of process types**, including:

- Your app's web server
- Multiple types of worker processes
- A singleton process, such as a clock
- Tasks to run before a new release is deployed (with optional “release” instruction)

If your app includes a web server, you should declare it as your app's web process :

**Procfile** (*Yes, this is your filename*)

```
web: $(composer config bin-dir)/heroku-php-nginx -C nginx.conf public/
```

# Heroku

```
# nginx.conf

location / {
    try_files $uri @rewriteapp;
}

# Rewrite all to index.php. This will trigger next location.
location @rewriteapp {
    rewrite ^(.*)$ /index.php/$1 last;
}

# Redirect everything to Heroku.
location ~ ^/index\.php(/|$) {
    try_files @heroku-fcgi @heroku-fcgi;
    internal;
}

# Return 404 for all other php files not matching the front controller.
# This prevents access to other php files you don't want to be accessible.
location ~ \.php$ {
    return 404;
}
```

# Heroku

Install Heroku CLI : <https://devcenter.heroku.com/articles/heroku-command-line>

Create a new Git repository :

**Initialize a git repository in a new or existing directory**

```
$ cd my-project/
```

```
$ git init
```

```
$ heroku git:remote -a this42app
```

Deploy your application :

**Commit your code to the repository and deploy it to Heroku using Git.**

```
$ git add .
```

```
$ git commit -am "make it better"
```

```
$ git push heroku master
```

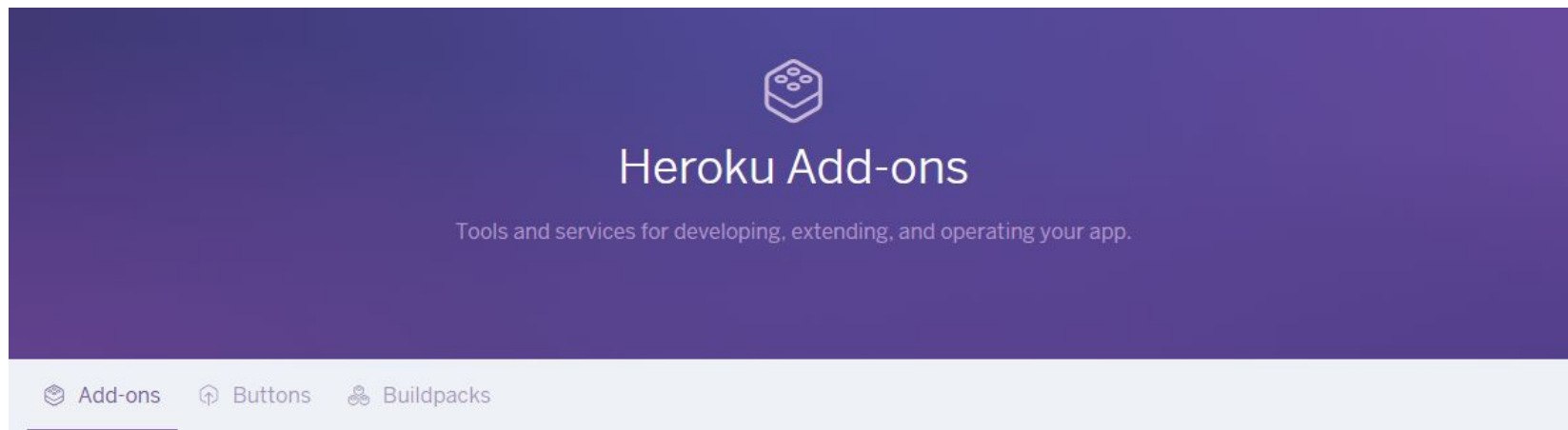
**N'oubliez d'ajouter en variable d'environnement d'Heroku (Settings => Config Vars) :**

**APP\_ENV : prod**

Existing Git repository :

**Simply add the heroku remote**

```
$ heroku git:remote -a this42app
```



## ADD-ON CATEGORIES

- Data Stores
- Data Store Utilities
- Monitoring
- Logging
- Email/SMS
- Caching
- Errors and Exceptions
- Content Management
- Search
- Metrics and Analytics
- Testing

## Data Stores Choose where to store your data.

 <b>Storj</b> <hr/> <p>Distributed object storage</p>	 <b>CloudKafka</b> <hr/> <p>Message streaming as a service powered by Apache Kafka</p>	 <b>AT&amp;T M2X</b> <hr/> <p>AT&amp;T's time-series data storage service for the Internet of Things (IoT)</p>	 <b>RedisGreen</b> <hr/> <p>Redis, Instrumented and Scaled</p>
--	--	---	---