

Gitlab CI/CD

- GitLab CI/CD Introduction
- Pipeline Creation & Configuration
- Automated Tests
- Web Application Deployment
- Runner
- GitLab Pages

Complex Tools to Deliver Software



Manage



Plan



Create



Verify



Package



Secure



Release



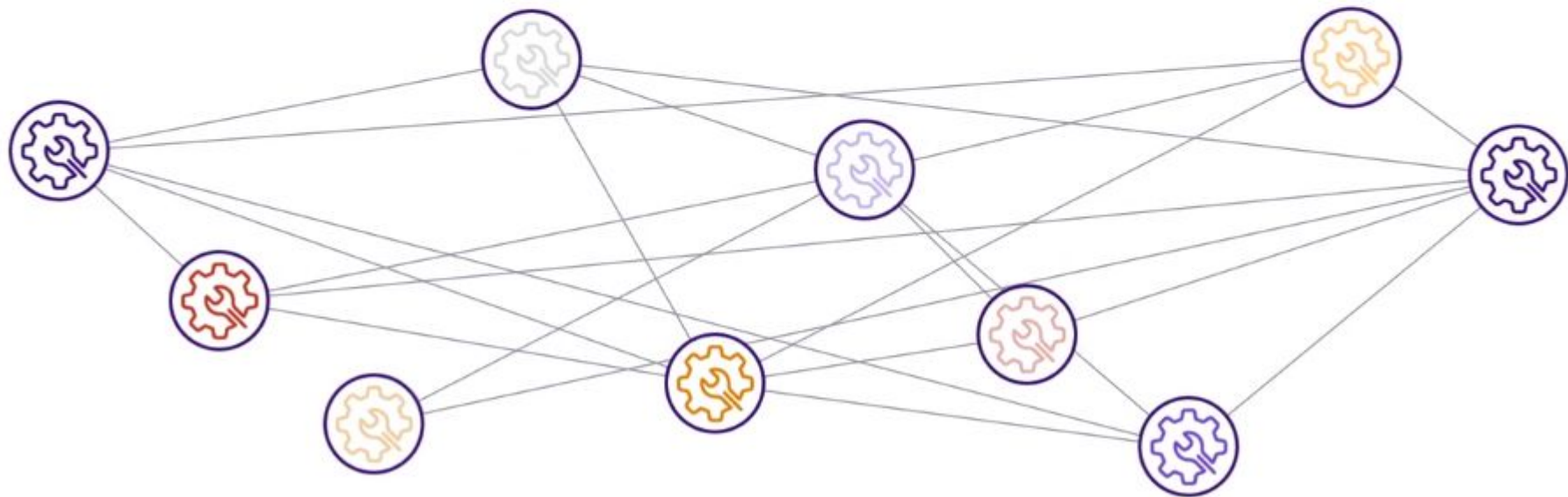
Configure



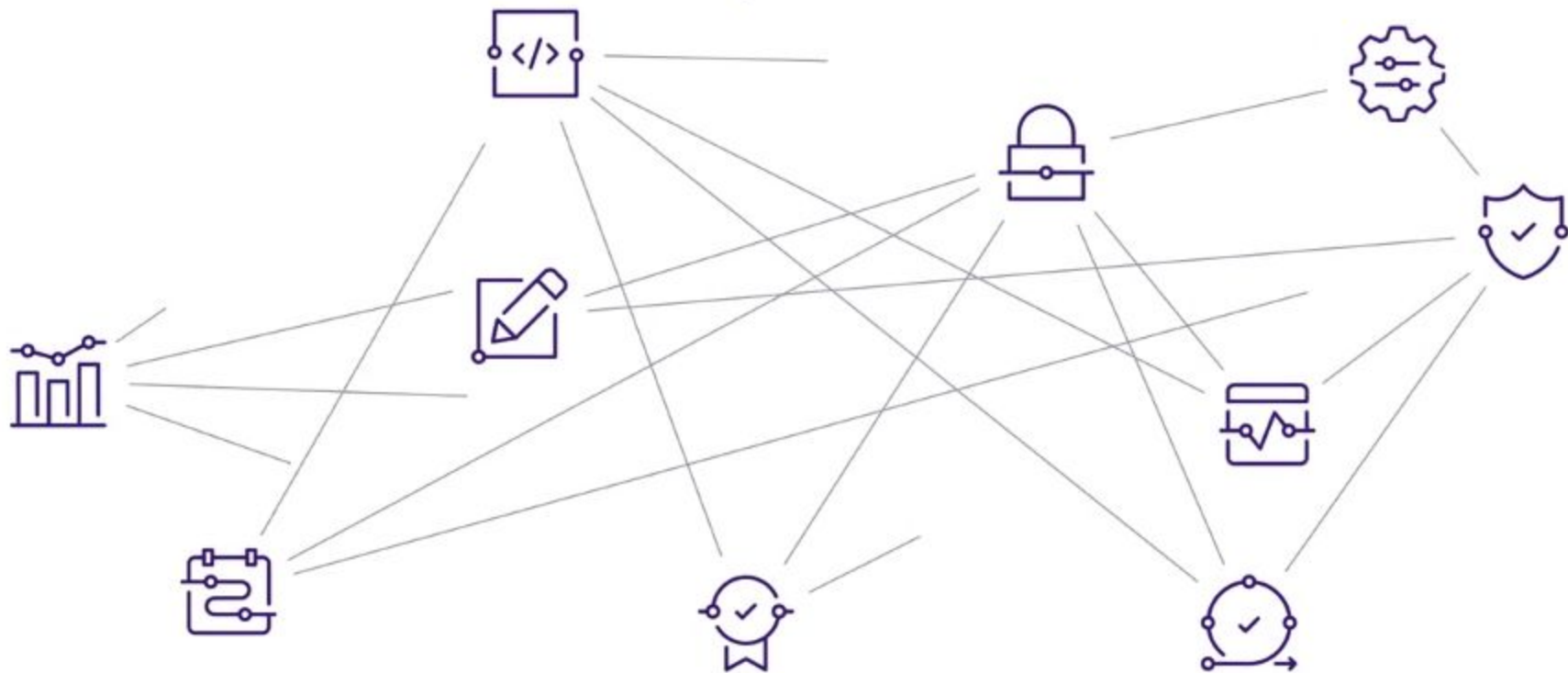
Monitor



Protect



The DevOps Platform



Single application for the DevOps lifecycle



Business



Developers



Security



Operations

One user interface



Manage



Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Protect

Unified data model

Waterfall



The DevOps Platform



● Code ● Test ● Deploy



The DevOps Platform



Manage



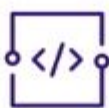
Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Protect

What is GitLab CI/CD ?

CI : Continuous Integration

- Build > Tests (unit, integration, regression...)

CD : Continuous Delivery

- Review > Deployment (staging, production...)

1 Automated Testing

Automates code testing with every modification, enabling rapid identification of errors and bugs.

2 Automated Deployment

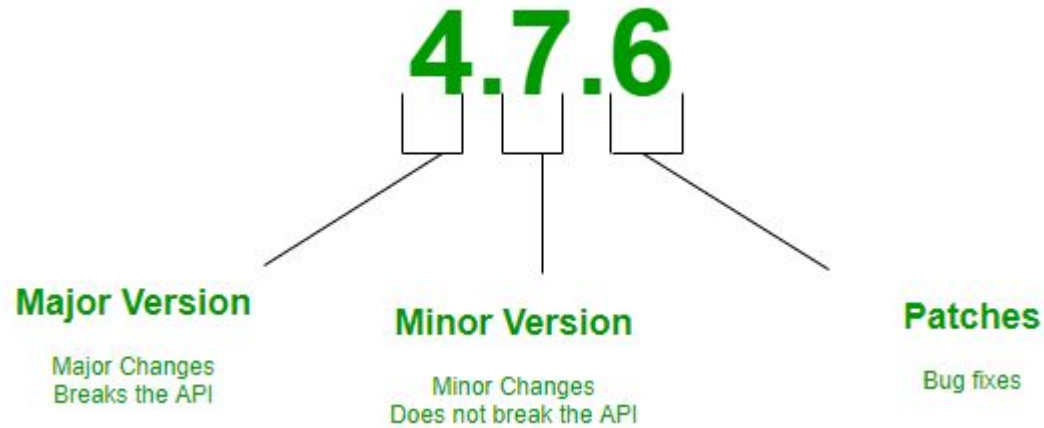
Facilitates automated code deployment once tests are successful, speeding up the production deployment process.

3 Change Tracking

Tracks each code modification and provides detailed reports, ensuring transparency and traceability.

...

SEMVER



1.3.32-alpha

4.3.22-test-93412a

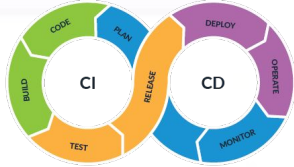
2.1557.0

What is GitLab CI/CD ?

Gitlab as a DevOps platform

- ❖ GitLab strives to become a complete DevOps platform
- ❖ Platform on which you build your complete DevOps workflows
- ❖ Improved code quality
- ❖ Increased development velocity
- ❖ Better collaboration between teams
- ❖ Save time and money

What is GitLab CI/CD ?



Continuously release code changes to the end environment



remote git
repository



Tests



Build &
Package



Artifact
repository



Deploy to
DEV

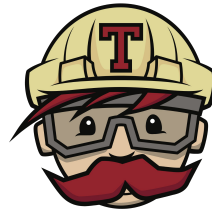


Deploy to
STAGING



Deploy to
PROD

Many CI/CD platforms



Travis CI



Many CI/CD platforms



GitLab CI/CD



Source Code & CI/CD on same platform



Source Code Management on GitLab



Allows keeping CI/CD & code management
in the **same place**



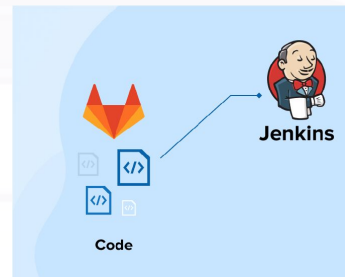
What is GitLab CI/CD ?



- ✓ Seamless integration into code repository
- ✓ Using CI/CD without overhead of setting it up yourself
- ✓ Pipeline configuration as part of your application code
- ✓ Self-Hosted or SaaS (managed)



- ✖ Only CI/CD tool
- ✖ Self-hosting is the only option



Benefits of using GitLab CI/CD

- Improvement of code quality
- Increased development speed
- Enhanced collaboration between teams
- Time and cost savings
- Improved reliability and security
- Adoption of a DevOps culture:

Avantages de l'utilisation de GitLab CI/CD

Single application for the DevOps lifecycle



Business



Developers



Security



Operations

One user interface



Manage



Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Protect

Unified data model

GitLab Architecture

- Serveur GitLab
- GitLab Instance
- GitLab Runners
- Container registry
- Webhooks

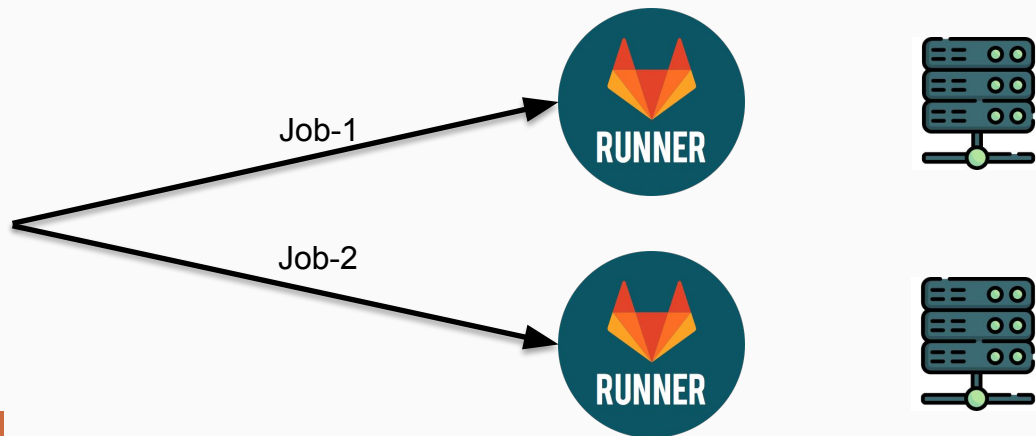
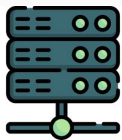
GitLab Runners



GitLab Instance

or

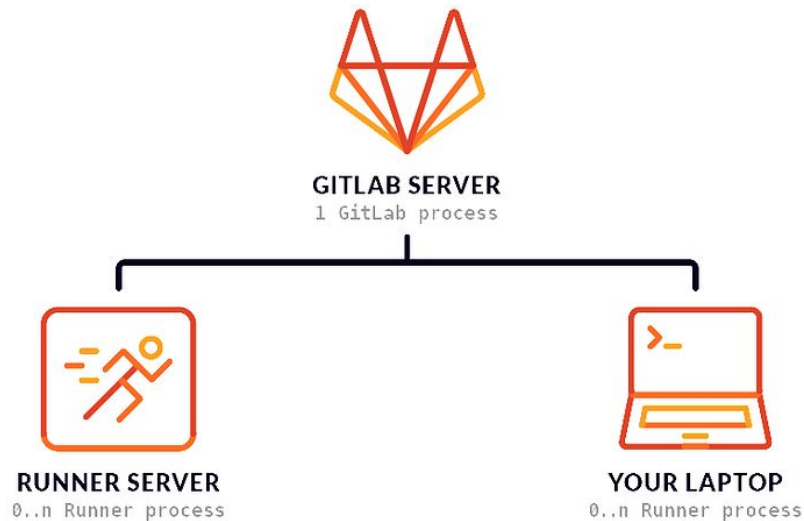
GitLab Server



GitLab Runners

- Agents that run your CI/CD jobs
- GitLab Server assigns pipeline jobs to available Runners

GitLab Runners



GitLab Runners

- An open-source tool to execute the jobs of your CI/CD pipelines
- Available on multiple platforms (Linux, Windows, macOS)
- You can use GitLab CI/CD Runner to execute builds, tests, deployments, and other automated tasks.
- Can be run on local machines, cloud instances, or Kubernetes servers



Creation de Runners

```
# Install a Runner on a Linux machine
curl -L
https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh | sudo bash sudo apt install gitlab-runner

# Install a Runner on a macOS machine
curl -L
https://packages.gitlab.com/install/repositories/runner/gitlab-runner/script.deb.sh | sudo bash sudo brew install gitlab-runner

# Register the Runner
gitlab-runner register
```

Creation de Runners

```
docker run -d --name gitlab-runner --restart always \  
  -v /srv/gitlab-runner/config:/etc/gitlab-runner \  
  -v /var/run/docker.sock:/var/run/docker.sock \  
  gitlab/gitlab-runner:latest
```

Configuring a Runner

Configuration options

- Name: The name of the Runner
- Description: A description of the Runner
- Tags: Tags that can be used to filter Runners
- Executor: The type of executor the Runner will use
- Job limit: The maximum number of jobs the Runner can execute simultaneously

Configuration de Gitlab CI/CD

Features

- Parallel job execution on multiple machines
- Run your jobs in isolated and reproducible environments
- Integration with GitLab: Trigger pipelines automatically from pushes or merges
- Monitoring and reporting: Visualize the status of your pipelines and job history

Started with GitLab CI/CD Runner

- Create a .gitlab-ci.yml file:
- Define the jobs and stages of your pipeline
Specify the runners to use for each job
- Trigger the pipeline: Push your code to GitLab
The pipeline will be automatically triggered

GitLab CI/CD Runner is an essential tool for any team that wants to adopt a CI/CD approach. It can help you improve the quality, speed, and reliability of your software development.

Pipeline Creation & Configuration

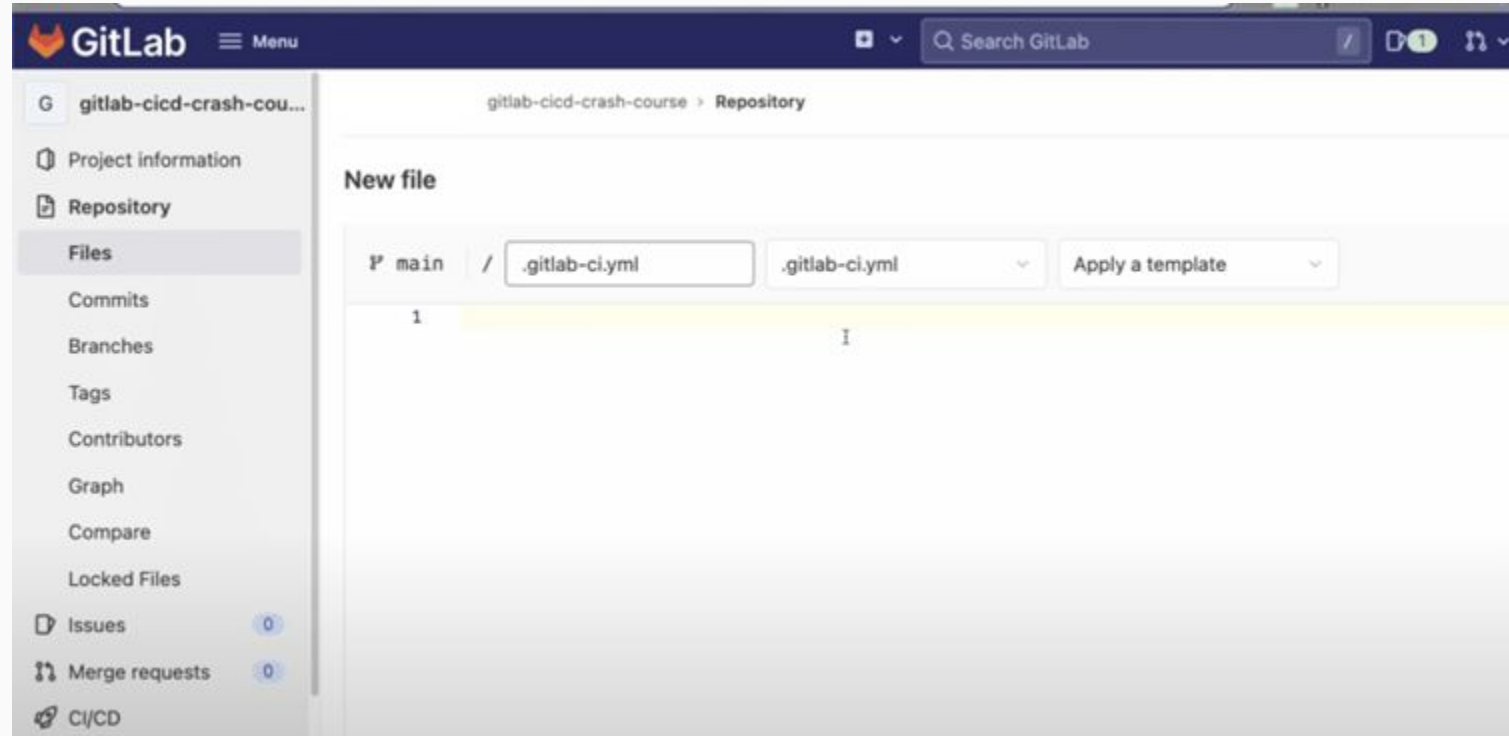


- Pipeline is written in code
- Hosted inside application's git repository

➡ Whole CI/CD configuration is written in **YAML format**

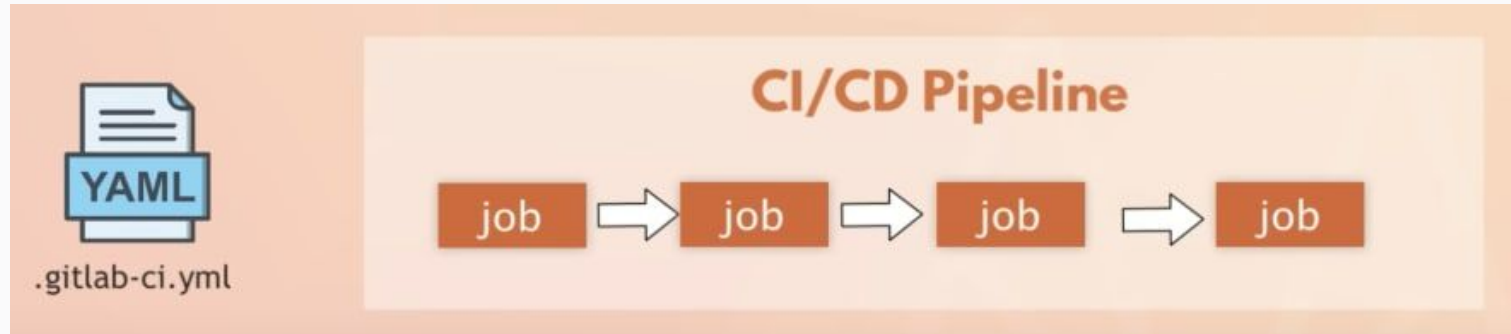
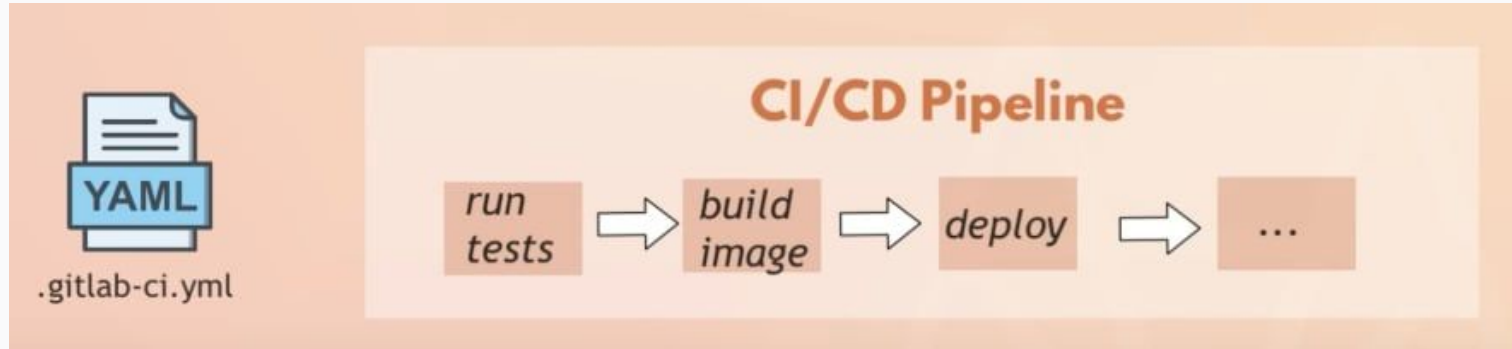


Pipeline Creation & Configuration



Pipeline Creation & Configuration

Jobs



YAML files

receipt: Oz-Ware Purchase Invoice

date: 2012-08-06

customer:

given: Dorothy

family: Gale

items:

- **part_no:** A4786
descrip: Water Bucket (Filled)
price: 1.47
quantity: 4
- **part_no:** E1628
descrip: High Heeled "Ruby" Slippers
size: 8
price: 100.27
quantity: 1

bill-to: &id001

street: |
123 Tornado Alley
Suite 16

city: East Centerville

state: KS

ship-to: *id001

specialDelivery: >

Follow the Yellow Brick
Road to the Emerald City.
Pay no attention to the
man behind the curtain.

...

Pipeline Creation & Configuration

Jobs

```
job1:  
  script:  
    "execute-script-job1"  
  
job2:  
  script:  
    "execute-script-job2"
```

The names of jobs must be unique and must not be part of reserved words

- Image
- Services
- **Stages (obligatoire)**
- Types
- Before_script
- After_script
- variables
- cache

Pipeline Creation & Configuration

Jobs

```
Job test:
  image : python:3.9
  before_script: apt-get
update
  script:
    ./bin/script/my-script.sh
```

- **run_tests:** name of the job.
- **image:** name of the docker image for executing commands.
- **script:** commands which need to be executed.
- **before_script:** any commands which need to be installed before running script commands in this case we need the make command to be installed to run our test.

Pipeline Creation & Configuration

Jobs - Script

```
job:script:  
  script: ./bin/script/my-script.sh ## Calling a script from  
your project.
```

```
job:scripts:  
  script: ## Calling two scripts from your project  
    - ./bin/script/my-script-1.sh  
    - ./bin/script/my-script-2.sh
```

```
job:command:  
  script: printenv # Executing a command
```

```
job:commands:  
  script: # Executing two commands  
    - printenv  
    - echo $USER
```

Steps in the GitLab CI/CD Deployment Process

1

Code Change Detection

The process starts with detecting any changes made to the source code.

2

Test Execution

Automated tests are executed to ensure the proper functioning of the code.

3

Automated Deployment

Once the tests are successful, the code is automatically deployed to the production servers.

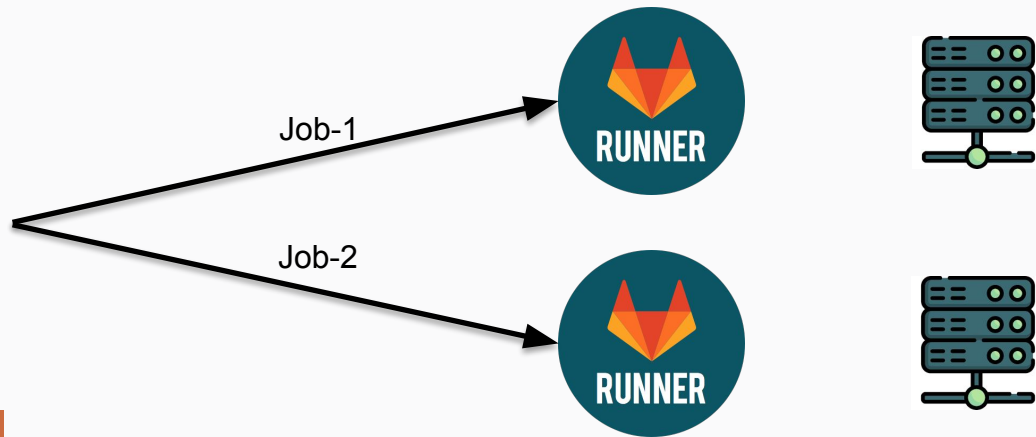
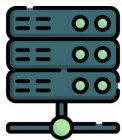
GitLab semver



GitLab Instance

or

GitLab Server



GitLab Runners

- Agents that run your CI/CD jobs
- GitLab Server assigns pipeline jobs to available Runners

GitLab Pages

- Effortless hosting for static websites
- Integrates seamlessly with GitLab repositories
- No infrastructure management required
- Free for public repositories and custom domains

GitLab Pages - Benefits of Using GitLab Pages

- **Simplicity:** Effortless setup and deployment
- **Version control:** Track changes and revert to previous versions
- **Security:** Benefit from GitLab's robust security infrastructure
- **Collaboration:** Work with teams and manage access easily
- **Customization:** Personalize your site with CSS and themes

GitLab Pages with framework

- Build your application
- Specify your build as artifact

TEST

DEPLOY

Example

d9f8e0aa

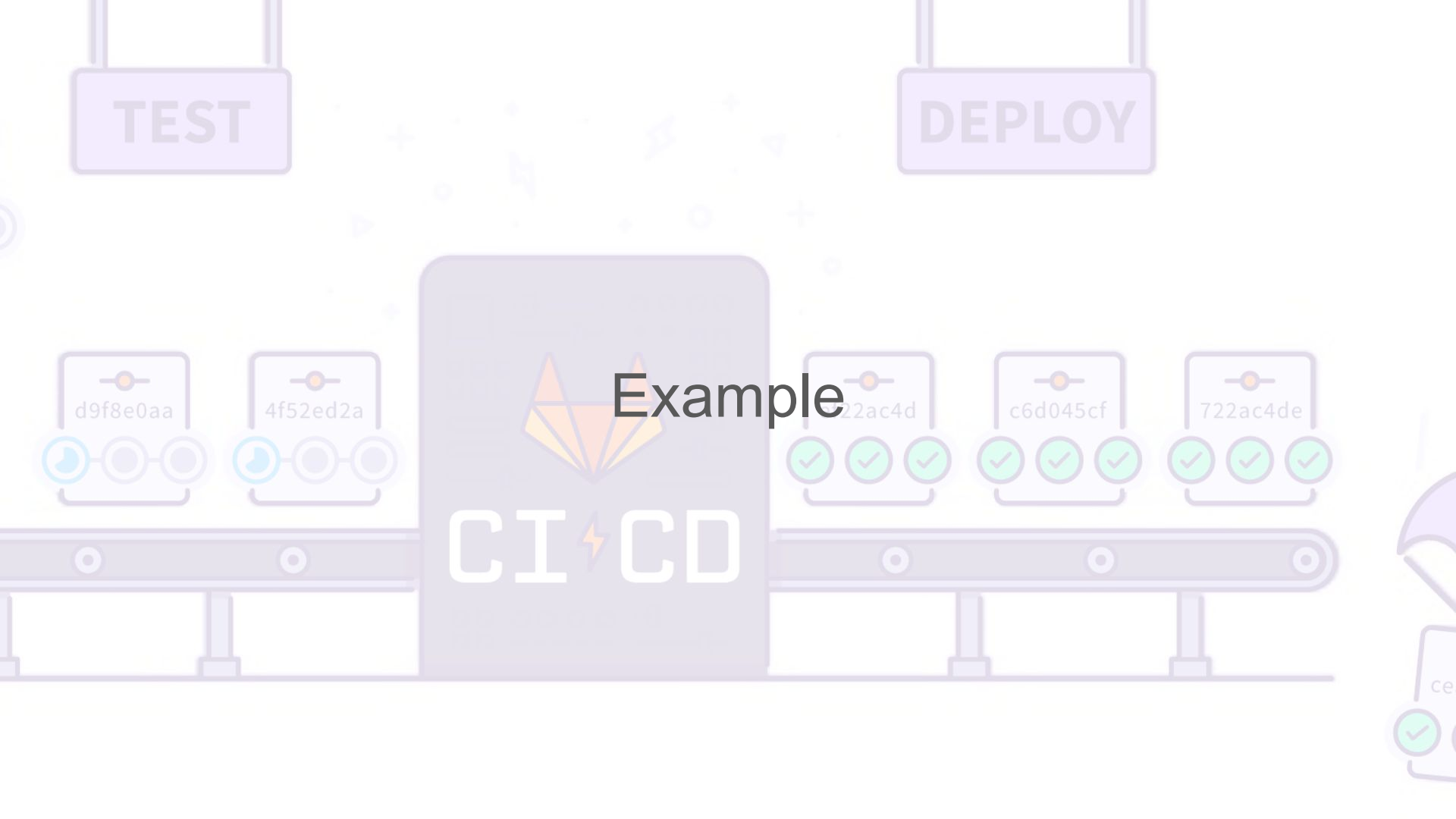
4f52ed2a

22ac4d

c6d045cf

722ac4de

CI⚡CD





Manage



Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Protect

Menu

Awesome Co

Subgroup Information

Epics 14

Issues 17

Merge requests 8

Security & Compliance

Packages & Registries

Analytics

Wiki

tech-marketing

Awesome Co

A

Awesome Co

Group ID: 12152054

Subgroups and projects

Shared projects

Archived projects

Search by name

Name

C Consumer Products

M Mobile App

W Web App

L Logistics

S Services

T Teams

I Ideas

O Ops

W Wiki

0

2

1

0

1 month ago

0

1 month ago

0

2

1

1

0

2 weeks ago

0

1 month ago

0

2 months ago

Collapse sidebar



Manage



Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Protect

GitLab

dev-sec-product-private/tenos/wayne-ent...

routes.py

```
21 delete_form = DeleteForm()
22
23 if add_form.validate_on_submit():
24     add_note(add_form.note_field.data)
25     flash('Note "{}" has been added!'.format(
26         add_form.note_field.data))
27
28     return redirect('/')
29
30 # TODO: Add return and flash error if note does not exist
31 if delete_form.validate_on_submit():
32     delete_note(delete_form.id_field.data)
33     flash('Note "{}" has been Deleted!'.format(
34         delete_form.id_field.data))
35
36     return redirect('/')
37
38 return render_template('index.html', notes=narr, add_form=add_form, delete_form=delete_form)
39
40 @note.route('/sitemap.xml', methods=['GET', 'POST'])
41 def site_map():
42     try:
43         return send_file('static/sitemap.xml', attachment_filename='sitemap.xml')
44     except Exception as e:
45         return str(e)
46
47 @note.route('/faq', methods=['GET'])
48 def faq():
49     try:
50         return render_template('faq.html')
51     except Exception as e:
52         return str(e)
53
54 @note.route('/add', methods=['POST'])
55 def add_note(msg=""):
56
57     if not msg:
58         data = request.get_json(force=True)
59         msg = data.get('message')
60
61     if not msg:
62         return "No Message sent in request."
63
64     conn = db.create_connection()
65     with conn:
```



Manage



Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Protect

Projects ▾ Groups ▾ More ▾

Search or jump to...

Platform **Open** Mobile Website re-design

Subscription metric decline pls fix #16

0 0

Oldest first ▾ Show comments only ▾ Create merge request ▾

Parker - Product Manager @parker.pm · 4 hours ago
@presley.design Some users have pointed out that the website is not entirely responsive and difficult to navigate on mobile devices

Presley - Product Designer @presley.design · 4 hours ago
@parker.pm Will check it out next sprint

Presley - Product Designer @presley.design · 2 hours ago
@parker.pm Uploaded Design to Design Management, let's discuss here and involve development

Parker - Product Manager @parker.pm · just now
Great, I agree, Sasha could you help us here? @sasha-dev

Write Preview

Write a comment or drag your files here...

Markdown and quick actions are supported

Comment ▾ Close issue

Developer

Labels: Design, Mobile UI, req_assigned, in_progress

Weight: None

Health status: None

Confidentiality: Not confidential

Lock issue: Unlocked

4 participants

Issue Tracking • Issue Boards • Design Management • Web IDE • SCM • Continuous Integration • CI: Code Testing, Code Coverage, SAST • Continuous Delivery • GitLab Metrics



Manage



Plan



Create



Verify



Package



Secure



Release



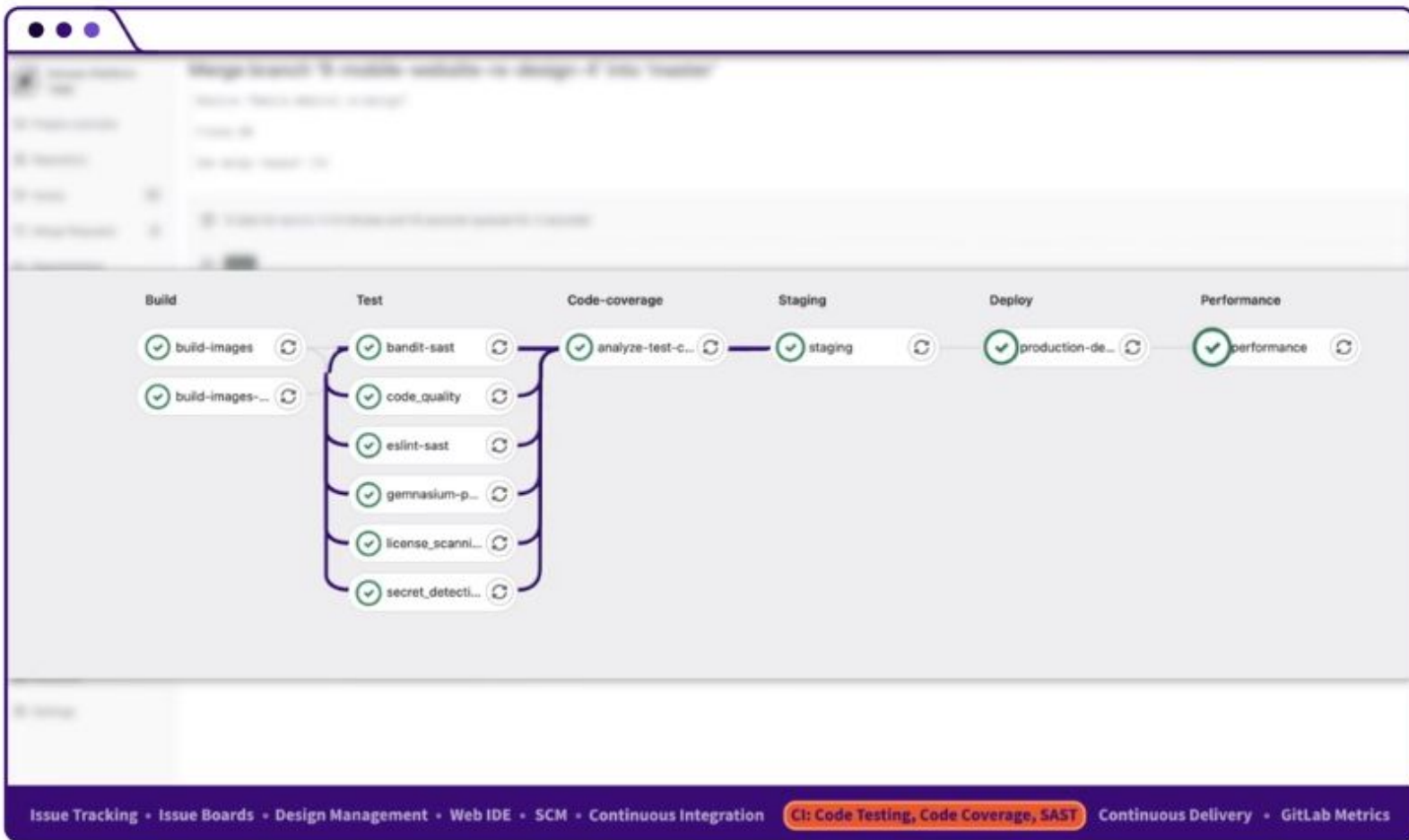
Configure



Monitor



Protect





Manage



Plan



Create



Verify



Package



Secure



Release



Configure



Monitor



Protect

GitLab

Next

Menu

Search GitLab

🔍

🌐 🌙 🏠 🔄 👤

GitLab.org

GitLab

Releases

Released date

17

New release

GitLab 14.3

90% complete

Milestone
14.3

Issues 1470
Open: 153 • Closed: 1317

Assets 8

Source code (zip)

Source code (tar.gz)

Source code (tar.bz2)

Source code (tar)

Other

GitLab 14.3 release post

NEW FEATURES 31

TOTAL FEATURES 2141

Ultimate

new features 2

total features 293

Secure

- License Compliance now supports Java 15: [License Compliance](#)
- Next Generation SAST to reduce Ruby false positives: [SAST](#), [Vulnerability Management](#)

Protect

- Project-level DAST and secret detection scan execution policies: [Security Orchestration](#), [DAST](#), [SAST](#)

Premium

new features 10

total features 351



Manage



Plan



Create



Verify



Package



Secure



Release



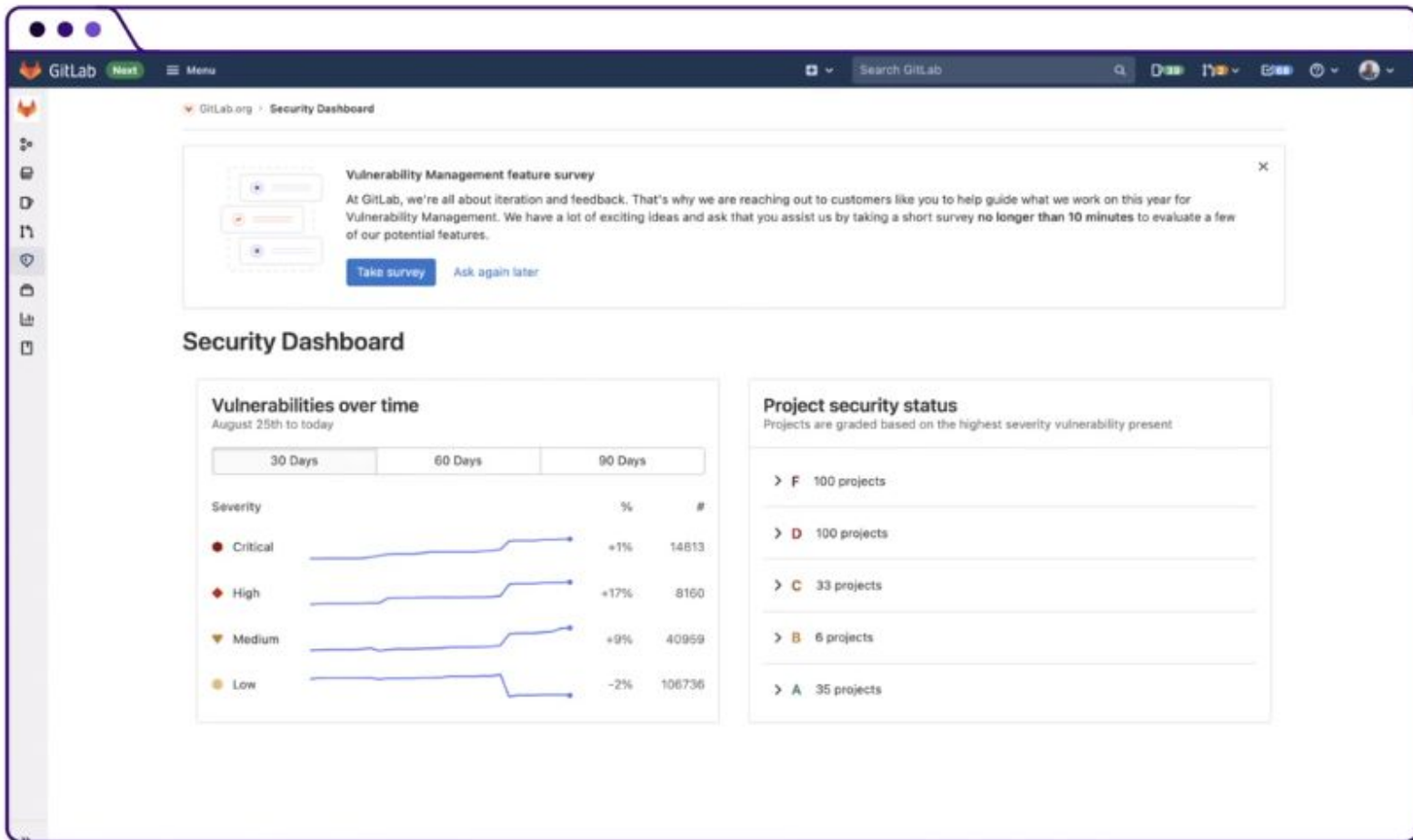
Configure



Monitor



Protect





Manage



Plan



Create



Verify



Package



Secure



Release



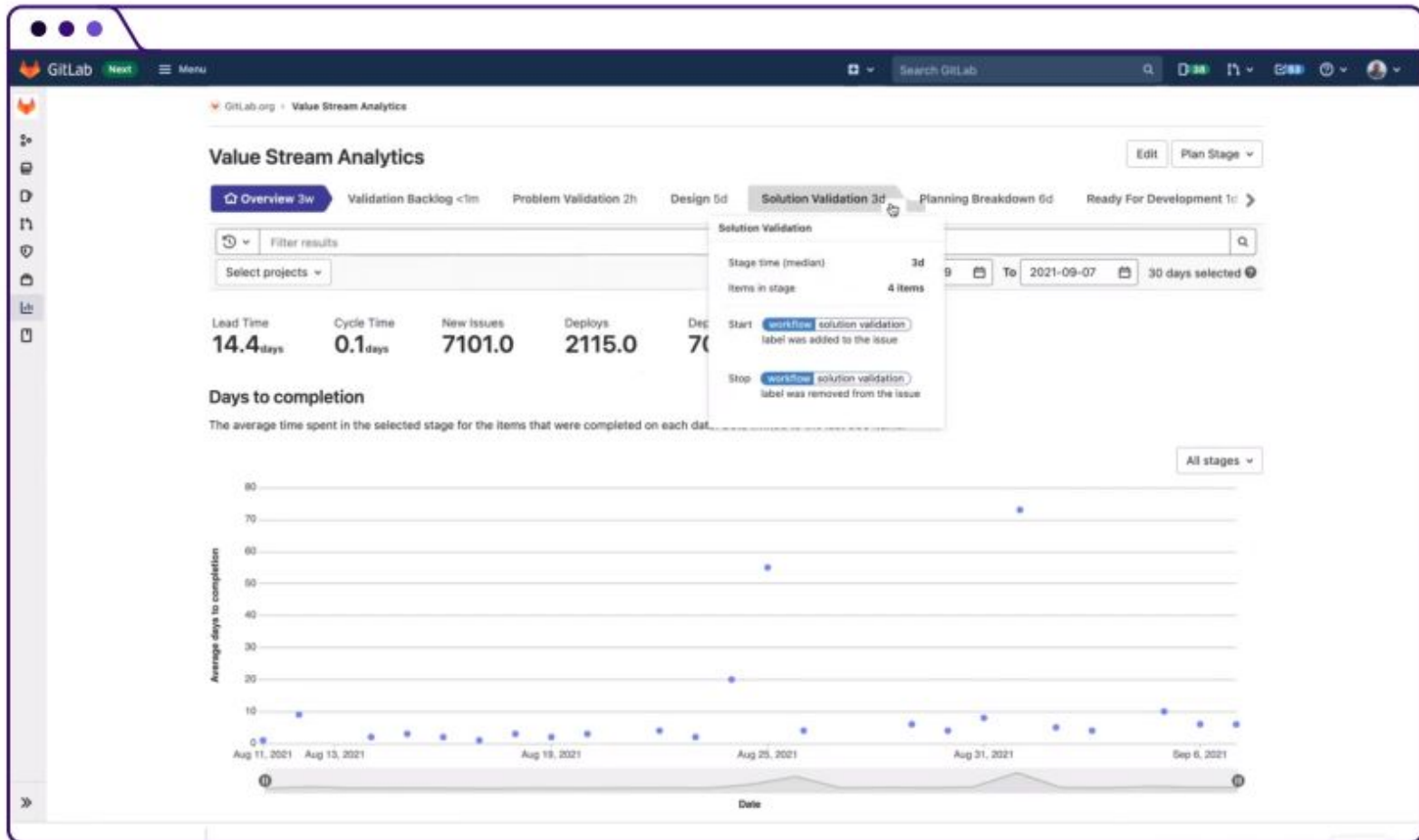
Configure



Monitor



Protect





Manage



Plan



Create



Verify



Package



Secure



Release



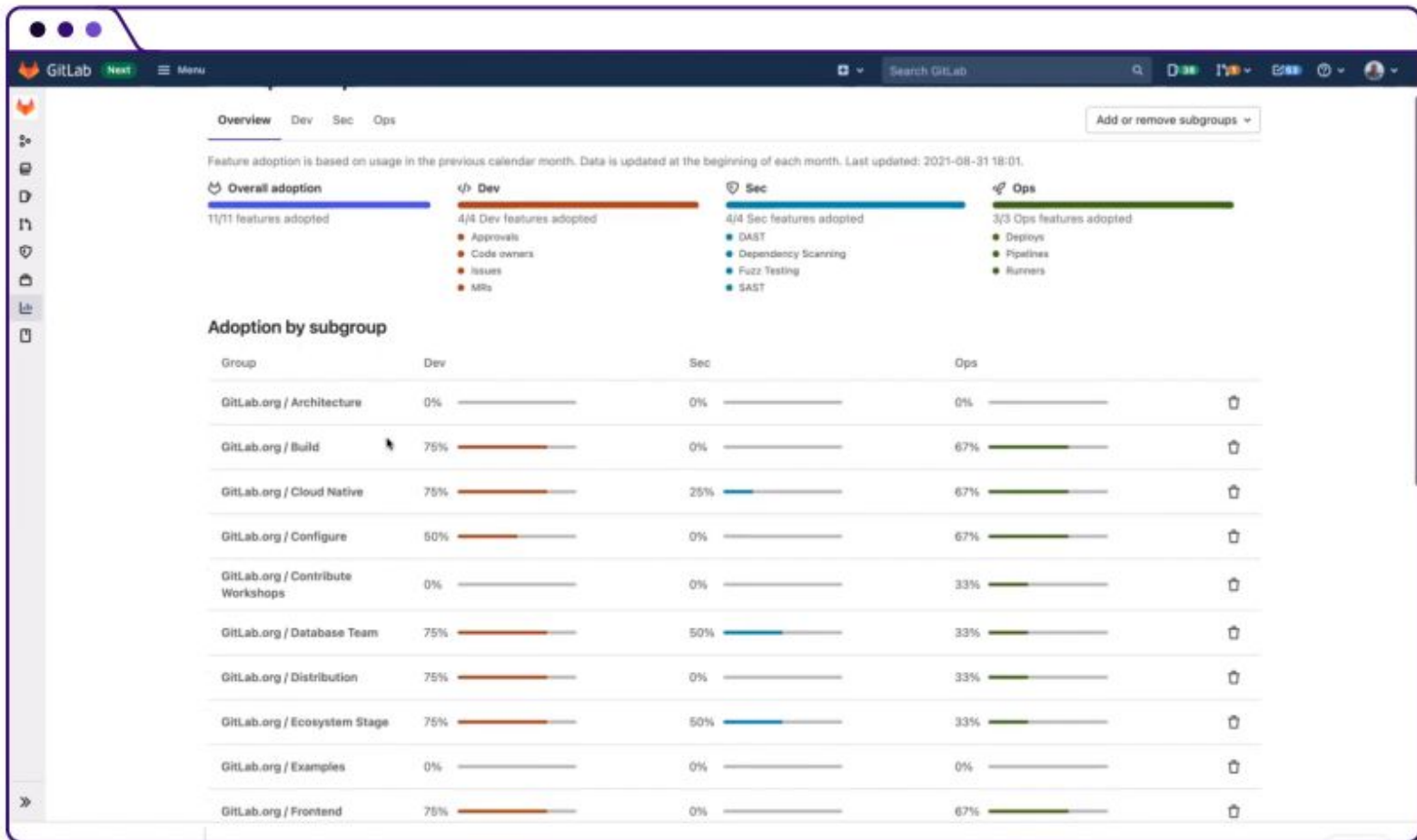
Configure



Monitor



Protect



GitLab

Gestionnaires de repositories git basés sur le web avec des fonctionnalités :

- un wiki
- un suivi d'issue
- un registry docker
- un suivi de code
- une review de code une CI/CD
- . . .