

temBoard



Pierre Giraud

PGSession - Paris - 22 Novembre 2018

temBoard

TITRE : temBoard

SOUS-TITRE :

DATE: Pierre Giraud



ARCHITECTURE

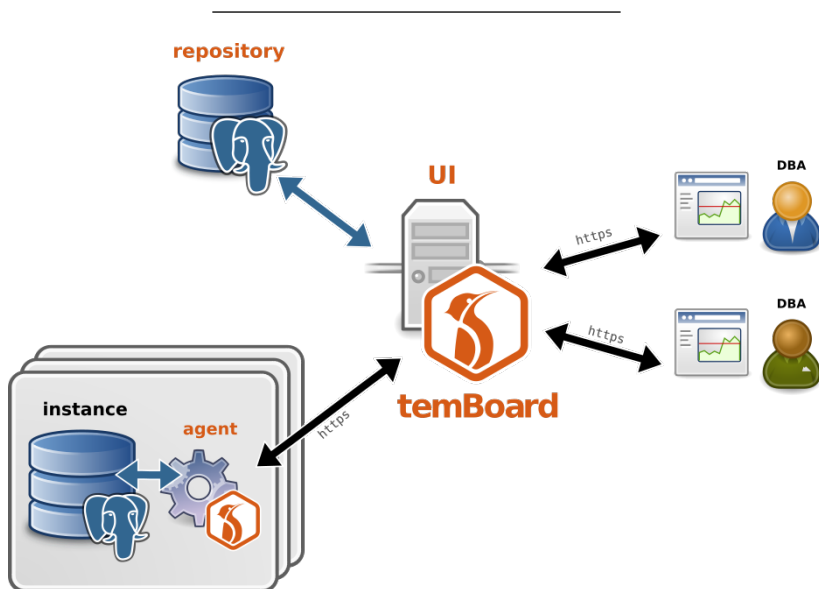


Figure 1: temBoard architecture

AGENT

- Mono-instance
- Pas de dépendances
- API REST
- Authentification
- Packagé pour centos/RHEL 6/7 et Debian

TemBoard agent est installé à côté de l'instance PostgreSQL.

Développé en python compatible 2 et 3.

1 instance -> 1 agent

L'agent récupère des métriques à intervalle régulier pour les envoyer au serveur (UI).

Il peut aussi recevoir des requêtes envoyées par le serveur pour l'affichage en temps réel.

Échange de données entre l'agent et le serveur via authentification dédiée et via protocole *HTTPS*.

SERVEUR

(aussi appelé UI)

- Interface Web
 - Python 2.7 / Tornado / SQLAlchemy
- Base de données historique et metadonnées
 - PostgreSQL 9.4+
- Authentification
- Packagé pour CentOS/RHEL 7 et Debian
- Un serveur pour un parc d'agents / instances

L'interface utilisateur de temboard est développée en python 2.7 et repose sur le framework web Tornado.

Une base de données, appelée *repository* est nécessaire à son fonctionnement, en effet, celle-ci va permettre de stocker :

- la liste des comptes utilisateurs habilités à se connecter à l'interface;
- la liste des instances Postgres à gérer;
- l'historique des données collectées.

L'accès à cette interface est protégée par une authentification utilisateur.

FONCTIONNALITÉS

PLUGINS

- Fonctionnement par plugins
 - Activation / désactivation
-

- Tableau de bord
- Configuration Postgres
- Supervision
- Activité

TABLEAU DE BORD

(*Plugin Dashboard*)

- Affichage en temps réel
 - Métriques système : usage CPU, mémoire, *loadaverage*.
 - Métriques Postgres : Cache Hit Ratio, Sessions, TPS.
 - Statut de chaque métrique calculé selon des seuils (*alerting*).
-

CONFIGURATION

(*Plugin pg_conf*)

- Configuration des paramètres de l'instance (postgresql.conf)

Configuration de `pg_hba.conf` et `pg_ident.conf` supprimée car cela va au-delà du périmètre de temBoard.

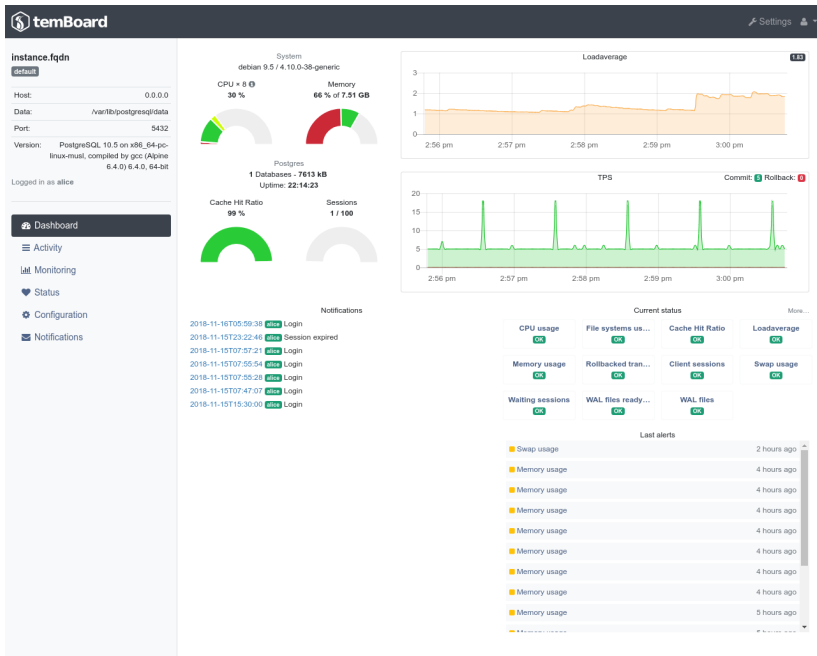


Figure 2: temBoard Dashboard

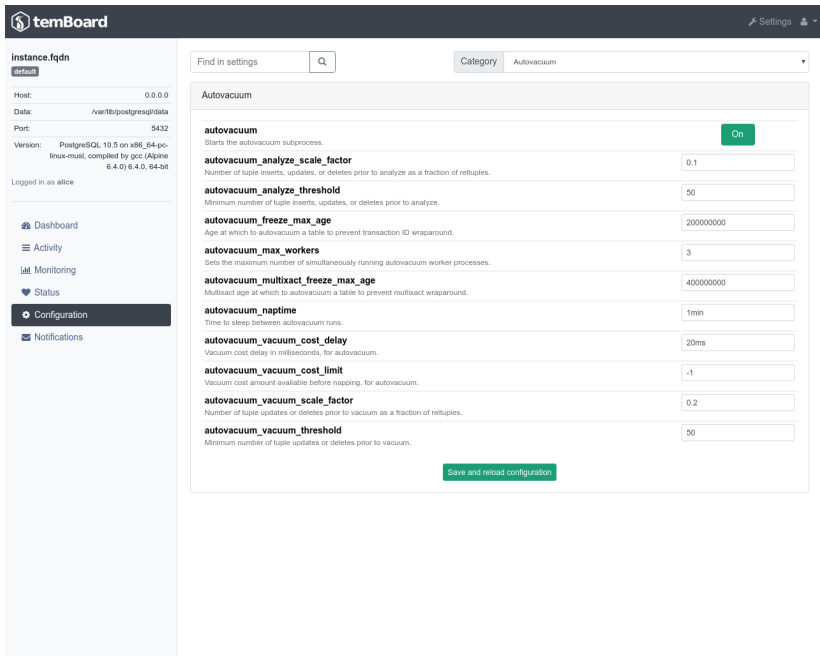


Figure 3: temBoard Configuration

SUPERVISION

(Plugin Monitoring)

- Collecte périodique de métriques par l'agent (système et PostgreSQL)
- Envoi à l'agent (push)
- Agrégation et historisation côté serveur
- Graphiques
- Seuils d'alerte

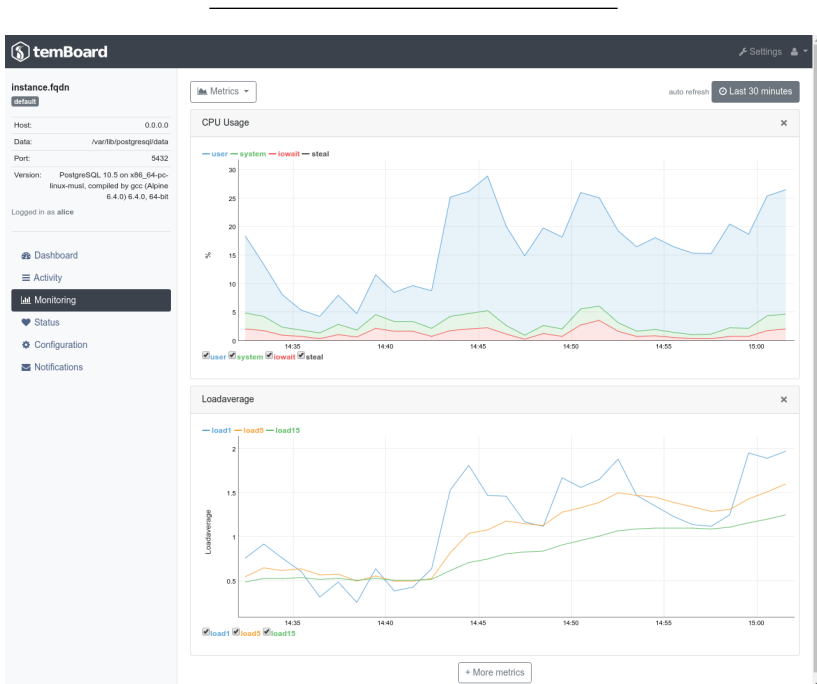


Figure 4: temBoard Monitoring

ACTIVITÉ

(Plugin Monitoring)

- (quasi) Temps réel
- Affichage des requêtes en cours d'exécution
 - base, utilisateur
 - CPU, mémoire, I/O
 - durée
- Requêtes bloquées
- Requêtes bloquantes
- Possibilité de terminer les backends

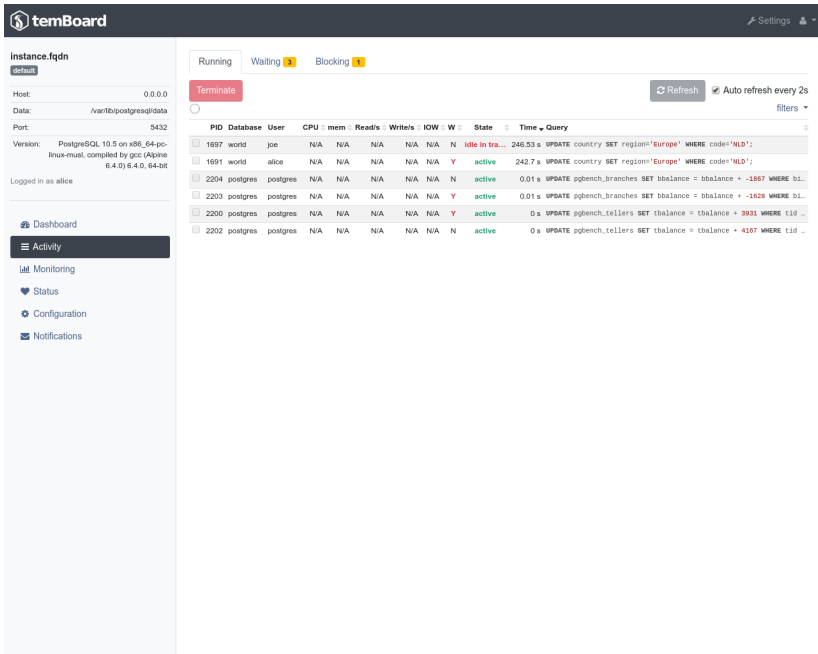


Figure 5: temBoard Activity

NOUVEAUTÉS DE LA V2

AMÉLIORATIONS DE L'INTERFACE

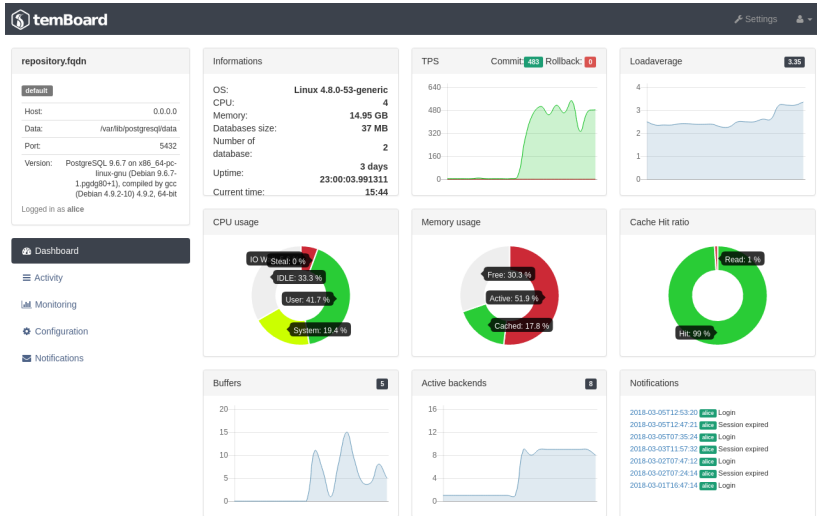


Figure 6: DASHBOARD avant

temBoard

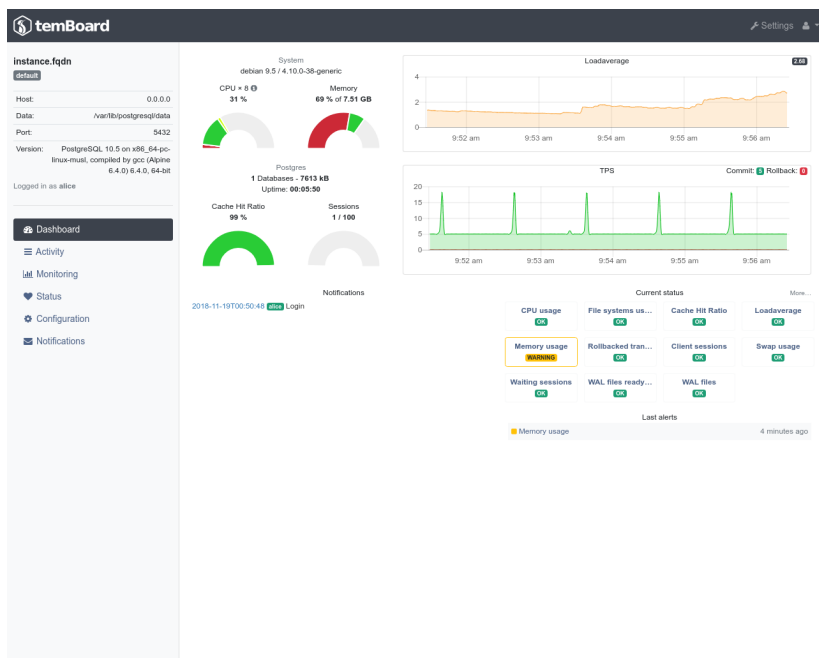


Figure 7: DASHBOARD après

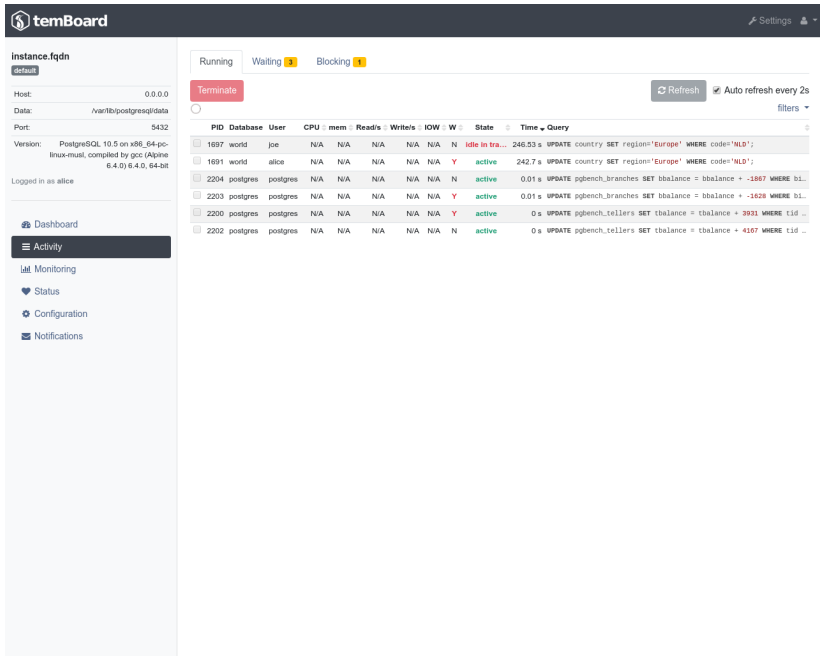


Figure 8: Activity

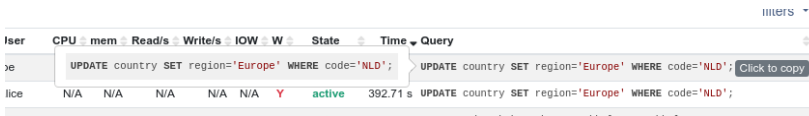


Figure 9: Copie SQL

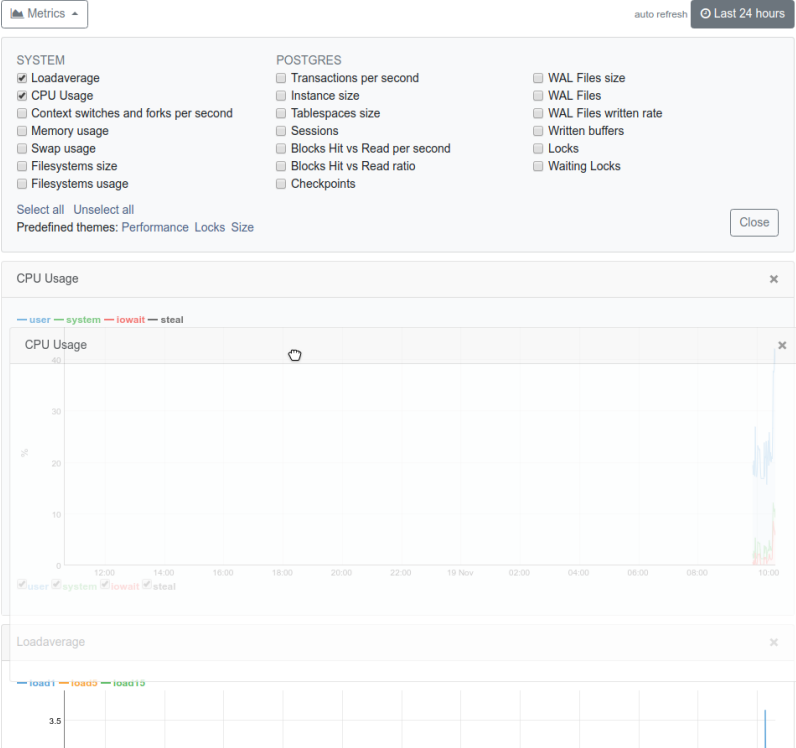


Figure 10: Customisation Monitoring

ALERTES

temBoard Settings

Search instances Sort by: hostname

Instance: fgdn View

Status: **WARNINGS: 1** **CRITICAL: 2**

Host: 0.0.0.0 default

Data: /var/lib/postgresql/data

Port: 5432

Version: PostgreSQL 10.5 on x86_64-pc-linux-musl, compiled by gc...

Graph showing two metrics over time (10:00 to 10:30). The left y-axis ranges from 0 to 200. The right y-axis ranges from 0 to 4. The green line shows a sharp spike around 10:15, reaching approximately 200. The orange line shows a steady increase from 0 to 4.

temBoard
Settings

instance.fqdn

default

Host: 0.0.0.0
 Date: /var/lib/postgresql/data
 Port: 5432
 Version: PostgreSQL 10.5 on x86_64-pc-linux-musl, compiled by gcc (Alpine 6.4.0) 6.4.0, 64-bit

Logged in as alice

Dashboard

Activity

Monitoring

Status

Configuration

Notifications

System
debian 9.5 / 4.10.0-38-generic

CPU = 8 **0** 37 % Memory 84 % of 7.51 GB

Postgres
2 Databases - 79 MB
Uptime: 91:52:27

Cache Hit Ratio 99 % Sessions 98 / 100

2018-11-19T00:50:48 Login

Notifications

Loadaverage

TPS

More...

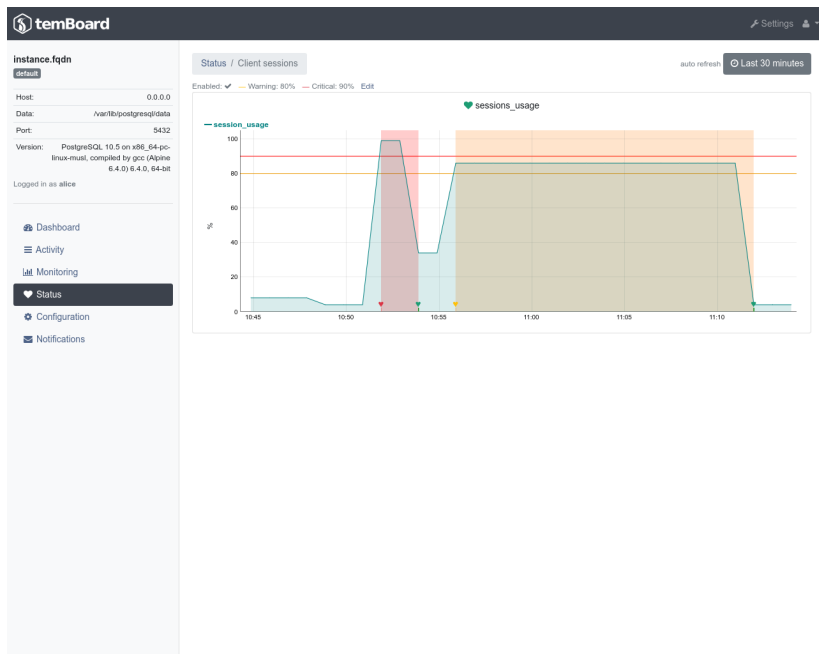
Current status	
CPU usage OK	File systems us... OK
Memory usage WARNING	Rollbacked tran... OK
Cache Hit Ratio OK	Client sessions CRITICAL
Loadaverage OK	Swap usage OK
Waiting sessions CRITICAL	WAL files ready... OK
	WAL files OK

Last alerts

- Waiting sessions - postgres a few seconds ago
- Client sessions a few seconds ago
- CPU usage - global 13 minutes ago
- Loadaverage 13 minutes ago
- Memory usage an hour ago

The screenshot displays the temBoard monitoring interface for an instance named 'fgdn'. The interface is organized into several sections:

- Instance Information:**
 - Host: 0.0.0.0
 - Date: /var/lib/postgresql/data
 - Port: 5432
 - Version: PostgreSQL 10.5 on x86_64-pc-linux-musl, compiled by gcc (Alpine 6.4.0) 6.4.0, 64-bit
 - Logged in as: alice
- Navigation Menu:**
 - Dashboard
 - Activity
 - Monitoring
 - Status** (selected)
 - Configuration
 - Notifications
- Monitoring Widgets:**
 - CPU usage:** Global status: Warning: 50% Critical: 80%
 - File systems usage:** Warning: 80% Critical: 90%
 - Cache Hit Ratio:** Warning: 90% Critical: 80%
 - Loadaverage:** Warning: 4 Critical: 8
 - Memory usage:** Warning: 50% Critical: 80%
 - Rollbacked transactions:** Warning: 10 Critical: 20
 - Client sessions:** Critical: 80% Critical: 90%
 - Swap usage:** Warning: 30% Critical: 50%
 - Waiting sessions:** Critical: 5 Critical: 10
 - WAL files ready to be archived:** Warning: 10 Critical: 20
 - WAL files:** Warning: 50 Critical: 100




FACTORISATION DE CODE

BUG FIXES!

AMÉLIORATIONS DE LA DOC

temBoard

- Home
- Docker Quick Start
- Installation**
- Prerequisites
- Prepare repository
- Installation
- Post Installation
- Install agents
- Configuration
- Upgrade
- Upgrade 1.2 to 2.0 (RHEL/CentOS)
- Upgrade 1.1 to 1.2 (RHEL/CentOS)
- Upgrade 0.0.1 to 1.1 (RHEL/CentOS)
- How to
- temBoard UI howto
- Alerting



Set up, manage and protect your Apple devices with **Jamt Now**. Try for free today.


Sponsored - Ads served ethically

[GitHub](#) [Previous](#) [Next](#)


If this fails, don't worry. You will have to run auto-configuration script with proper parameters, once temBoard package is installed.

Installation


Choose the method matching your target system:



Debian



RHEL / CentOS



PyPI

RHEL / CentOS

temBoard RPM are published on [Dalibo Labs YUM repository](#). temBoard supports RHEL / CentOS 7. Start by enabling Dalibo Labs YUM repository.

```
$ sudo yum install -y https://yum.dalibo.org/labs/dalibo-labs-2-1.noarch.rpm
$ sudo yum makecache fast
```

Warning

Do NOT use temBoard rpm from PGDG. They are known to be broken.

With the YUM repository configured, you can install temBoard UI with:

```
$ sudo yum install temboard
$ temboard --version
```

Post installation

If postinst auto-configuration fails, you can still relaunch it with proper parameters. Call the script `/usr/share/temboard/auto_configure.sh` with libpq-style envvars.

The postinst script creates Postgres role, database and tables, as well as self-signed SSL certificate, UNIX user, configuration file and systemd unit. A few steps are left to the administrator.

Default admin user

By default, temBoard is set up with a dumb `admin` user with password `admin`. This is totally unsecured. It is **strongly recommended to change default password!** See below.

POINTS FAIBLES

- Liste des sondes de supervision pas complète
- Pas de notifications par e-mail pour les alertes
- cloud-ready

FONCTIONNALITÉS À VENIR



MODE PULL (VS. PUSH)

PLUGIN MAINTENANCE

CLOUDIFICATION

The screenshot displays the temBoard interface. On the left is a sidebar with navigation options: Dashboard, Activity, Monitoring, Status, Configuration, Maintenance (highlighted), and Notifications. The main area shows the instance details for 'instance.fqdn' (version 0.0.0.0) and a table titled 'All Databases'.

database		schema		tables			indexes		
name	size	name	size	count	size	bloat	count	size	bloat
world	XXXGB	public	XXXGB	> 20	XXXGB	0%	> 10	XXXGB	0%
		bench	165GB	> 50	149GB	2%	> 32	13GB	0%
MyDB	32.2GB	public	XXXGB	> 200	XXXGB	51%	> 10	XXXGB	82%


Settings 

instance.fgdn

default








Host: 0.0.0.0

Date: /var/lib/postgresql/data





Port: 5432

Version: PostgreSQL 10.5 on x86_64-pc-linux-gnu, compiled by gcc (Alpine 6.4.0) 6.4.0, 64-bit

Logged in as alice

-  Dashboard
-  Activity
-  Monitoring
-  Status
-  Configuration
-  Maintenance
-  Notifications

All Databases / world / public / table1

 Heap Size XXXMB	 Heap Bloat XXXMB	 Indexes Size XXMB	 Indexes Bloat XXMB	Toast Size XXXMB	Fill Factor 20%
---	--	---	--	----------------------------	---------------------------

Last ANALYZE: 14 days ago

Last auto-ANALYZE: 15 days ago

Last VACUUM: 15 days ago

Last auto-VACUUM: 14 days ago

Indexes:

name	size	type	bloat	bloat %	scans	tablespace	definition
idx_12	50MB	BTREE	1MB	0.5%	100	tablespace2	<input type="button" value="Reindex"/> CREATE INDEX idx_12 ON table2 (id)
idx_13	500MB	BTREE	75MB	15%	120	tablespace3	<input type="button" value="Reindex"/> CREATE INDEX idx_12 ON table2 (id)

MERCI

QUESTIONS / REMARQUES ?

Your browser does not support the video tag.

LIENS

- [Code](#)
- [Documentation](#)
- [Quickstart docker](#)