

---

# Popcube Documentation

*Version 0.1*

**cl3m3nt**

**sept. 13, 2017**



<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Outils de communication	3
1.1.1	Slack	3
1.2	Contrôle de version	5
1.2.1	Github	6
1.3	Intégration continue	7
1.3.1	Travis	7
1.4	Tests	7
1.4.1	Tests unitaire go	7
1.4.2	Linter	9
1.4.3	Test fonctionnel	9
1.5	Introduction	10
1.6	Contraintes	10
1.6.1	Developpement	10
1.6.2	Home	11
1.6.3	Organisation	11
1.6.4	Channel	11
1.6.5	Messages	11
1.6.6	Settings	11
1.7	Fonctionnalité	11
1.7.1	Organisation	12
1.7.2	Appel	13
1.7.3	Partages	13
1.7.4	Messages	13
1.8	Interface	13
1.8.1	Design	13
1.8.2	Hiéarchie	14
1.8.3	Home	14
1.8.4	Organisations	14
1.8.5	Settings	15
1.9	Introduction	15
1.10	Contraintes	16
1.10.1	Haute Disponibilité	16
1.10.2	Stockage et Sauvegarde	16
1.10.3	Réseaux	16
1.10.4	Sécurité	16

1.11	Hyperviseur . . . . .	16
1.11.1	VMware ESXi . . . . .	16
1.11.2	Vsphere client . . . . .	17
1.11.3	Vcenter . . . . .	17
1.11.4	Vue globale . . . . .	18
1.12	Conteneur . . . . .	18
1.12.1	Cluster swarm . . . . .	18
1.12.2	Storage . . . . .	19
1.12.3	Stack . . . . .	20
1.13	Stockage et réplication . . . . .	20
1.13.1	Minio . . . . .	20
1.14	Monitoring de l'infrastructure . . . . .	21
1.14.1	Interne . . . . .	21
1.14.2	Externe . . . . .	22
1.15	Load Balancing . . . . .	23
1.15.1	Load Balancing externe . . . . .	23
1.15.2	Load Balancing : Swarm mode routing mesh . . . . .	23
1.15.3	Revers proxy / Load Balancing interne . . . . .	23
1.16	Automatisation . . . . .	24
1.16.1	Provisionnement des machines virtuelles . . . . .	25
1.16.2	Ajout aux cluster swarm . . . . .	25
1.16.3	Conteneur . . . . .	27
1.16.4	Monitoring . . . . .	29
1.16.5	Reverse Proxy . . . . .	29
1.17	Vue d'ensemble . . . . .	29
1.18	Introduction . . . . .	30
1.18.1	Qui somme nous ? . . . . .	30
1.18.2	Popcube qu'est ce que c'est ? . . . . .	30
1.18.3	On ne s'arrête pas la ! . . . . .	30
1.19	Comment on gagne du cash . . . . .	30
1.19.1	Free . . . . .	30
1.19.2	Small . . . . .	31
1.19.3	Big . . . . .	31
1.19.4	Assets . . . . .	31
1.20	Prochainement dans PopCube . . . . .	31
1.20.1	Développement de la structure . . . . .	31

The code is open source, and available on [GitHub](#).



Society est une startup dédié aux applications de communication personnelles et professionnelles.  
Leur nouveau produit est un système complet de messagerie instantanée, tel que Skype ou Slack.

## Outils de communication

### Slack

Slack est une plate-forme de communication collaborative propriétaire (SaaS) ainsi qu'un logiciel de gestion de projets

Plugin :

- Monitoring avec grafana

**PopCube Inc.** cl3m3nt

**#grafana** | 3 | Add a topic

**monit** APP 12:38 PM ☆

[Alerting] Test notification

@channel Someone is testing the alert notification within grafana.

High value	Higher Value
100	200

Grafana v4.1.1 May 13th at 12:38 PM (30kB)

**monit** APP 4:06 PM

[Alerting] Cluster Memory Limits alert

@channel

Limited

2805880239.1579

Grafana v4.1.1 May 13th at 4:06 PM

**monit** APP 4:54 PM

+ Message #grafana

## — Travis

**PopCube Inc.** cl3m3nt

**#travis\_notification** | 4 | Add a topic

**Travis CI** APP 9:24 AM

Build #2 (72e19d0) of cl3m3nt666/popcube\_deploy\_api@master by Clément failed in 1 min 48 sec

Build #3 (138c4c9) of cl3m3nt666/popcube\_deploy\_api@master by Clément passed in 2 min 3 sec

**Travis CI** APP 9:35 AM

Build #3 (63ec292) of cl3m3nt666/popcube\_docs@master by Clément passed in 0 min 33 sec

**Travis CI** APP 9:49 AM

Build #4 (1af0ce6) of cl3m3nt666/popcube\_docs@master by Clément passed in 0 min 47 sec

**Travis CI** APP 10:33 AM ☆

Build #313 (1523731) of titouanfreville/popcubeapi@dyn\_mysql\_host by Clément failed in 3 min 53 sec

Build #314 (12b9d07) of titouanfreville/popcubeapi@develop in PR #55 by Clément failed in 3 min 49 sec

**Travis CI** APP 10:53 AM

Build #5 (ba85531) of cl3m3nt666/popcube\_docs@methods by Clément passed in 0 min 32 sec

new messages

+ Message #travis\_notification

## — Github

**PopCube Inc.** cl3m3nt

CHANNELS (13)

- # archi
- # back
- # dev
- # dev\_ops
- # front
- # general
- # github\_notification**
- # grafana
- # ia
- # marketing
- # random
- # travis\_notification
- # unit-testing

DIRECT MESSAGES

- slackbot
- cl3m3nt (you)
- ceylas
- jerem-no-one
- lagaffes
- lagaffes, ceylass

**#github\_notification**

☆ | 👤 4 | 🔒 0 | Add a topic

🔍 Search

📧 ⚙️ 📄

🗓️ April 4th

**github\_bot** APP 12:06 PM  
[PopCube:issue-4-view] 2 new commits by Titouan:

- bce5838 Merge pull request #28 from titouanfreville/development - Titouan
- 97ec6f6 Merge branch 'master' into issue-4-view - Titouan

🗓️ May 3rd

**github\_bot** APP 1:51 PM  
[PopCube:issue-4-view] 1 new commit by ceylass:

- ebccaf6 [WIP] New window for settings - ceylass

🗓️ May 6th

**github\_bot** APP 9:03 PM  
[PopCube:issue-4-view] 1 new commit by ceylass:

- b9aa215 Settings page - ceylass

**github\_bot** APP 9:29 PM  
[PopCube:issue-4-view] 1 new commit by ceylass:

- 8148f47 [WIP] Profil on nav - ceylass

+ Message #github\_notification

— Trello

**PopCube Inc.** cl3m3nt

CHANNELS (13)

- # general
- # github\_notification
- # grafana
- # ia
- # marketing
- # random
- # travis\_notification
- # unit-testing

DIRECT MESSAGES

- slackbot
- cl3m3nt (you)
- ceylas
- jerem-no-one
- lagaffes
- lagaffes, ceylass
- trello**
- + Invite people

**@trello**

☆ | ● active

🔍 Search

🔧 📄

🗓️ January 4th

**trello** APP 10:58 PM ☆

You're all set! Remember to invite me to your channels with `/invite @trello`. Here are some commands to get you started.

- Add a card, optionally assigning teammates**  
`/trello add [teammates] [card name]`  
Example: `/trello add @matt @jessica Finish blog post`
- Link a board from your Trello team to this Slack channel**  
`/trello link [search terms or board url]`  
Example: `/trello link Project Awesome`  
Example: `/trello link https://trello.com/b/nCQJJoZ/project-awesome`
- Change the list cards are added to**  
`/trello set-list`
- Show details of a card or board**  
`/trello [board or card url]`  
Example: `/trello https://trello.com/b/nC8JJoZ/project-awesome`

Note: The commands below will update the last card expanded in the channel

- Assign teammates to a card**  
Example: `/trello assign @sam @katie`
- Add a comment to a card**  
Example: `/trello comment Awesome work!`
- Add or update the due date on a card**

+ Message @trello

## Contrôle de version

Git permet, comme beaucoup de ses congénères (Subversion, CVS, Arch ou encore Darcs), de travailler à plusieurs sur un même projet. Mais contrairement à CVS ou Subversion, par exemple, Git ne fait pas de différence entre un dépôt principal et les copies locales chez les différents contributeurs.

## Github

GitHub est un outil gratuit pour héberger du code open source

**Les branches DEMO et DEVELOP doivent rester stable en toute circonstance. Si un merge casse la branche, il doit être revert puis corriger avant de le merge de nouveau.**

**La branche master sera mise à jour toute les semaines sur la branche develop. Aucun autre merge ne doit être réaliser sur master en aucune circonstance**

**La branche demo doit être mise à jour sur develop avant chaque préparation de demo ou en fin de semaine**

### Préparer son environnement de travail

- Installez *git*, *docker* et *docker-compose*
- Récupérer le code source du projet : `git clone git@github.com :titouanfreville/{REPO}.git` || `git clone https://github.com/titouanfreville/{REPO}.git`
- Aller dans le dossier obtenu
- `docker-compose up`

### Proposer une feature / Séparer les tâches

- Créer une issue sur github contenant un nom (résumant le travail à faire) et l'intégralité de la feature (ex : Nom : HELLO WORLD, Contenu : Créer une page de l'application permettant d'afficher le message : 'Hello World')
- Signaler la création de l'application.
- Voter et ce mettre d'accord ;)

### Travailler sur une issue

#### Quand une feature est acceptée

- Créer une branche depuis la branche 'dévelop' nommer *issue-Numéro\_de\_l'issue-Nom*
- Créer les test unitaires liés à l'issue si non existant
- Faire votre feature
- Tester (lancer les test unitaires, etc.)
- Ouvrir une pull request de votre branche vers master

### Format de commit

- ajouter le template de commit a la configuration git. `git config commit.template {Clone_ROOT}/.git_commit_message.txt`
- Un commit doit contenir : le numéro de l'issue (requis), un résumé du travail réaliser(requis), un détails du travail réalisé, la liste de ce qu'il reste à faire, un ping sur les différentes personne concerné par l'issue.

### Finir une issue

#### Quand une feature est en Pull Request

- Une personne différente de celle ayant fait l'issue vérifie le code
- La feature est testé et une démo doit être faite (merge sur la branche demo)
- La PR est acceptée
- Merge la branche dans develop
- Tester que develop reste stable

## Intégration continue

L'intégration continue est un ensemble de pratiques, vérifier à chaque modification de code source que le résultat des modifications ne produit pas de régression dans l'application développée.

### Travis

Pour faire simple, vous pouvez paramétrer Travis-CI pour qu'à chaque push, il exécute vos tests et vous indique si votre build (plusieurs "jobs") s'est bien déroulé ou non. Cela permet à chaque modification du projet, de s'assurer que tout fonctionne correctement. Cela s'appelle l'intégration continue.

Chaque feature/branch est tester a chaque commit/merge

## Tests

### Tests unitaire go

On utilise goconvey

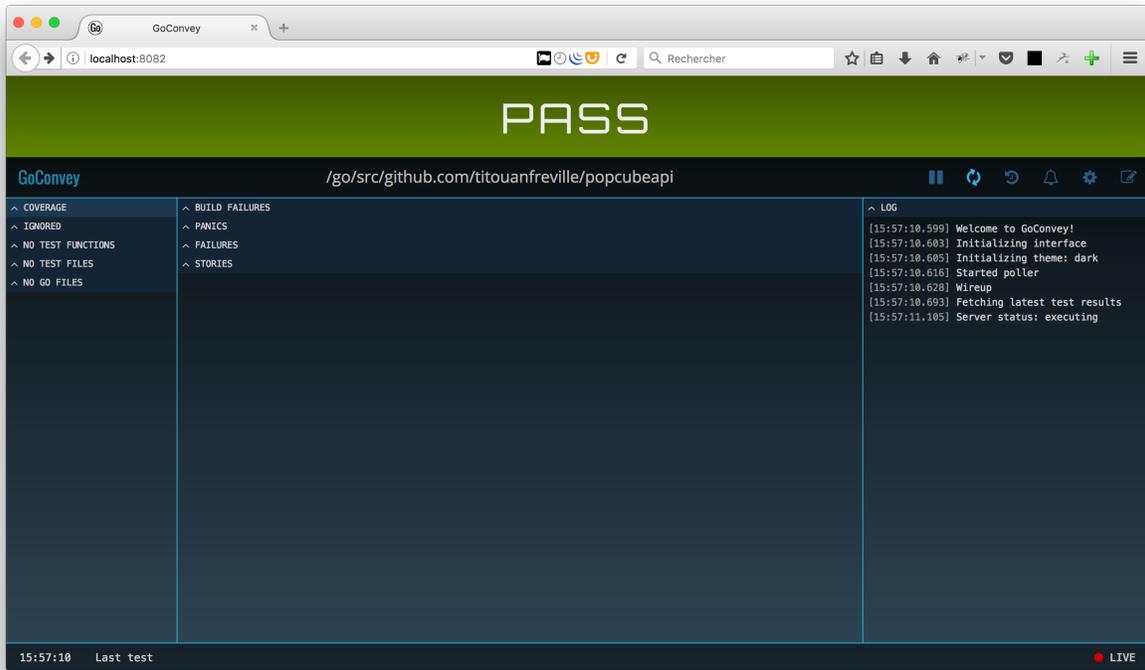
GoConvey, un outil de test Go pour les gophers. Utilisez-le dans le terminal ou le navigateur.

### Exemple web ui

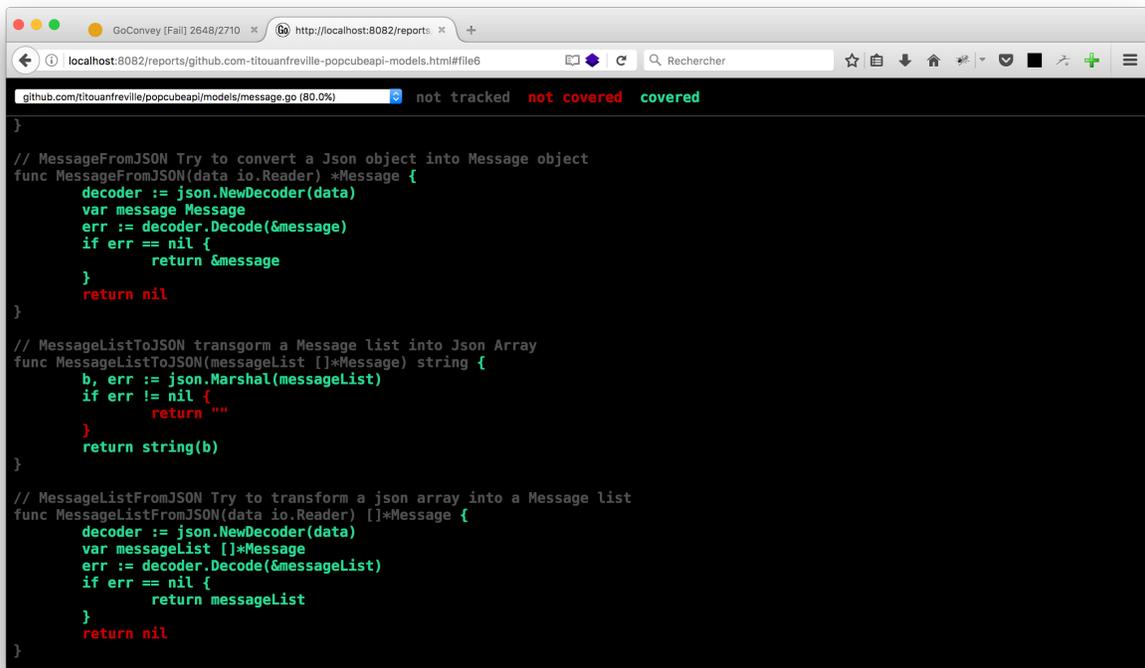
— Erreur

The screenshot shows a web browser window displaying the GoConvey test runner interface. At the top, a large orange banner with the word "FAIL" in white text indicates a test failure. Below this, the GoConvey logo and the path "/go/src/github.com/titouanfreville/popcubeapi" are visible. The main content area is divided into several sections: a left sidebar for navigation, a central pane showing the test details, and a right pane for logs. The test details section shows the test name "datastores/allowedWebMails\_store\_test.go:49" and the failure message: "Error 1146: Table 'popcube\_test.allowed\_web\_mails' doesn't exist----, ". The logs section shows the execution flow, including "Wireup", "Fetching latest test results", "Server status: executing", "Updating watch path", "Compiling package statistics", "Assertions: 2710", "Passed: 2648", "Skipped: 0", "Failures: 62", "Panics: 0", "Build Failures: 0", "Coverage: 36% (+36%)", "Rendering frame (id: 0)", "Rendering finished", "Processing complete", "Server status: idle", "Tests have finished executing", "Fetching latest test results", "Updating watch path", "Compiling package statistics", "Assertions: 2710", "Passed: 2648", "Skipped: 0", "Failures: 62", "Panics: 0", "Build Failures: 0", "Coverage: 36% (+36%)", "Rendering frame (id: 1)", "Rendering finished", "Processing complete". At the bottom, a status bar shows "15:57:39 Last test a few seconds ago 2710 assertions 62 failed 0 panicked 0 skipped 26.029s" and a "LIVE" indicator.

— OK



— Detail du code  
En rouge les zones non coverage



## Lint

L'utilitaire lint a été l'un des premiers outils d'analyse statique de code source.

Il s'agit d'un outil capable de détecter les erreurs et problèmes potentiels dans votre code.

— Linter pour golang

<https://github.com/cl3m3nt666/docker-golinter>

## Test fonctionnel

Avec postman il est possible d'écrire des tests, effectué a chaque jobs travis

```

[130] <git:(develop 830ac95*) > docker run -v $PWD/api_test/popcube.ping.postman_collection.json:/etc/newman/popcube.ping.postman_collection.json postman/newman_alpine33 run /etc/newman/popcube.ping.postman_collection.json
Unable to find image 'postman/newman_alpine33:latest' locally
latest: Pulling from postman/newman_alpine33
6c123565ed5e: Pull complete
a3ed95caeb02: Pull complete
7ca602f472d3: Pull complete
ef952462eb09: Pull complete
Digest: sha256:dcddee52c9f4719d524e2c8d7213f9e4189eb011f8511a6ab5a9db8ccdc998620
Status: Downloaded newer image for postman/newman_alpine33:latest
newman

Popcube inner API

ping
  Pong
    GET https://maxime.popcube.xyz/ping [200 OK, 171B, 384ms]
    1. Response time is less than 200ms
    ✓ Status code is 200
    ✓ Pong ? :D



|                                  | executed | failed |
|----------------------------------|----------|--------|
| iterations                       | 1        | 0      |
| requests                         | 1        | 0      |
| test-scripts                     | 1        | 0      |
| prerequisite-scripts             | 0        | 0      |
| assertions                       | 3        | 1      |
| total run duration: 628ms        |          |        |
| total data received: 4B (approx) |          |        |
| average response time: 384ms     |          |        |



# failure      detail
1. AssertionFailure Response time is less than 200ms

```

Exemple du fichier pour tester l'api

```

{
  "variables": [],
  "info": {
    "name": "Popcube inner API",
    "_postman_id": "93985f54-5ddd-a4ed-2a1a-e74912ffa258",
    "description": "The inner API of Popcube aim to define every methods,
    ↪need to manage communication organisation inside PopCube chat project",
    "schema": "http://schema.getpostman.com/json/collection/v2.0.0/
    ↪collection.json"
  },
  "item": [
    {
      "name": "ping",
      "description": "Folder for ping",
      "item": [
        {
          "name": "Pong",
          "event": [

```

```

    {
      "listen": "test",
      "script": {
        "type": "text/
↪ javascript",
        "exec": [
          "tests[\
↪ \"Response time is less than 200ms\"] = responseTime < 200;",
          "tests[\
↪ \"Status code is 200\"] = responseCode.code === 200;",
          "tests[\
↪ ? :D\"] = responseBody.has(\"pong\");",
          ""
        ]
      }
    },
    "request": {
      "url": "https://maxime.popcube.xyz/
↪ ping",
      "method": "GET",
      "header": [],
      "body": {},
      "description": "Test api ping"
    },
    "response": []
  }
]
}

```

## Introduction

Popcube est un projet permettant de regrouper des équipes venant de divers domaines et de leurs donner les outils de communications dont ils ont besoin. Tchat, appels voix et appels vidéo, partages de fichiers, popcube à tous pour vous charmer alors venez l'essayer.

## Contraintes

### Developpement

- Créer et gérer un compte
- Statut de connexion (En ligne, Absent ...)
- Manager des contact
- Créer des groupes de discussions
- Créer des channel
- Quitter les groupes ou channel
- Gestion avec des modérateurs
- Partages de fichier
- Appel multiples

## Home

- Créer un compte :
  - Email
  - Password
  - Avatar
  - Username
- Register
- reset password

## Organisation

- Accès rapides aux membres, channels.
- Possibilité d'administrer les channels, les membres et les messages pour les modérateurs.
- Quitter une organisation

## Channel

- Groupes de discussion
- Quitter un channel
- Lister les membres connectés dans un channel.

## Messages

- Ecrire des messages
- Historique des conversations
- Ouverture de l'url
- Texte avec la possibilité d'écrire en gras, italique
- Messages avec nom, date et avatar
- Files dragable
- OS Notification
- Emojis
- Message privés

## Settings

- Configurer micro et caméra
- Configurer port
- Changer files download location

## Fonctionnalité

Popcube est avant tout remplis de fonctionnalités utiles aux partages de ressources et communications rapide privée ou groupée.

## Organisation

L'organisation est tout simplement le nom de domaine sur PopCube. Il permet à vos équipes de se retrouver sous un seul nom et à tous vos collaborateurs de vous retrouver facilement.

### Membres

Les membres sont les utilisateurs inscrits sur votre organisation. Ils ont des droits et la possibilité de rejoindre des channels selon leurs habilitations.

### Channel

Un channel est un salon de discussion. C'est la place pour discuter avec votre service et/ou équipe sur un type particulier de besoin. Le nom du channel est important pour aider la hiérarchisation de votre organisation et permettre à vos membres de rejoindre les channels qu'ils désirent.

Exemple :

Il existe 3 types de channel :

- Les channels texte :
  - Dialogue texte
  - Partage de ressources
- Les channels audio :
  - Dialogue texte
  - Partage de ressources
  - Dialogue voix
- Les channels vidéos :
  - Dialogue texte
  - Partage de ressources
  - Dialogue voix avec visio

Les membres peuvent engager des discussions privées entre eux avec la possibilité d'appel audio ou vidéo.

### Droits

Les droits permettent à une organisation de manager leurs équipes en leur donnant accès à des droits sur les channels.

### Priver

Le statut priver s'applique à l'organisation comme aux channels. Il permet de bloquer l'accès de ceux qu'y n'ont pas été invités.

### Public

Une organisation ou un channel public ne bloque aucun accès. Chaque membre d'une organisation peut y accéder sans avoir de droits particuliers.

## Management

Il est possible de créer des groupes auxquels des membres sont associés pour éviter d'avoir à gérer un par un des channels. Les groupes sont associé à des channels et peuvent gérer d'autres groupes permettant ainsi à des équipes d'en gérer d'autres.

## Appel

Les appels permettent aux équipes de communiquer rapidement et directement. Lors de la connexion sur un channel vous êtes directement connecté avec les autres participants. Il est possible de voir qui est actuellement en ligne sur un channel.

## Voix

Les channels voix permettent de créer un salon de discussion dans lequel les membres peuvent discuter à l'aide d'un micro.

## Vidéos

Les channels vidéos permettent de créer un salon dans lesquels les membres peuvent discuter à l'aide d'un micro et d'une caméra.

## Partages

Il est possible d'envoyer des fichiers relié à un channel. Chaque utilisateurs se connectant au channels peut les voir ou recevoir.

## Messages

Les messages sont envoyé avec certaines balises html afin de pouvoir mettre en forme le texte. L'application reconnais si le message contient des liens afin de les faire ressortir et permet de cliquer directement dessus.

## Interface

L'interface intuitive permet la gestion de plusieurs organisations et permet de suivre une multitude de channel.

## Design

Utilisation d'un design épuré pour faciliter la compréhension des utilisateurs.

## Hiérarchie



developpement/../../../../\_static/developpement/interface/hierarchie.PNG

**Home** login register

Organisations Channel

Settings profil organisation channel droit

## Home

La page home apparaît lors de la première connexion et permet de trouver une organisation.



developpement/../../../../\_static/developpement/interface/home.PNG

## Login

Une fois l'organisation trouvée, l'utilisateur se retrouve sur la page de login, qui lui permet de se connecter ou de rejoindre la page register.



developpement/../../../../\_static/developpement/interface/login.PNG

## register

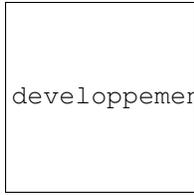
Permet à l'utilisateur de s'enregistrer sur l'organisation.



developpement/../../../../\_static/developpement/interface/register.PNG

## Organisations

Permet de sélectionner l'organisation sur lequel l'utilisateur veut parler.



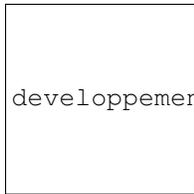
developpement/../../../../\_static/developpement/interface/organisation.PNG

### **Channel**

Permet de se connecter au channel sur lequel l'utilisateur veut parler.

### **Settings**

Permet de gérer tous les paramètres de l'utilisateur et de gérer l'organisation et les channels si l'utilisateur a les droits suffisant.



developpement/../../../../\_static/developpement/interface/settings.PNG

### **profil**

Profil de l'utilisateur

### **organisation**

Gestion de l'organisation par le ou les administrateurs

### **channel**

Gestion des channels par le ou les administrateurs

### **droit**

Gestion des groupes de droits et des droits par utilisateur

## **Introduction**

Popcube est une solution de messagerie instantanée qui se veut rapide, efficace et sécurisée.

## Contraintes

### Haute Disponibilité

L'infrastructure doit être capable de supporter tout type charge.

Pour cela, il faut utiliser 2 datacenters qui fonctionnent en mode actif/actif avec un CDN.

Dans chaque datacenter, il faut un cluster actif/passif pour le serveur frontend.

### Stockage et Sauvegarde

#### 1. Stockage

Historique de conversation Réplication sur tout les noeuds pour que les ressources mis en ligne par les utilisateurs puissent être tout le temps accessible. Changement à chaud des disques

#### 2. Sauvegarde

Etablir un plan de sauvegarde sur un 3ème datacenter pour qu'il puisse assurer une reprise du service (mode dégradé).

La sauvegarde se fera tous les jours à 2H avec un dump des bases de données et des ressources partagées par les utilisateurs.

Une copie des données devra être présente sur le datacenter d'un prestataire pour qu'il puisse effectuer les sauvegardes sur bande tous les jours.

### Réseaux

Choix libre pour les protocoles IPv4/IPv6 ainsi que pour le nombre de lien intersite.

### Sécurité

Implémenter 2 solutions de sécurité (bannissement IP, chiffrement) car les utilisateurs partageront des données sensibles.

#### 1. Bannissement sur les échecs de connexions

Les adresses IP qui auront échoués 10 fois lors des tentatives de connexion à l'application devront être bannies pendant une durée de 5 minutes.

Si elle rééchoue au bout de 10 essais, elle sera bannie durant 1 semaine.

Enfin, elle est bannie pour 2 mois si 10 autres tentatives ont échouées ...

#### 2. Chiffrement

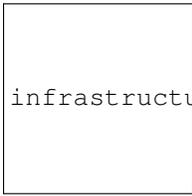
Les connexions entre les 3 datacenters doivent être entièrement chiffrées.

## Hyperviseur

### VMware ESXi

VMware ESXi est un hyperviseur qui s'installe directement sur un serveur physique (type bare-metal) pour gérer des machines virtuelles. Conçus de façon à ce que la surface d'attaque exposée aux logiciels malveillants et aux menaces réseau soit réduite, il est fiable et sécurisé.

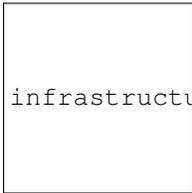
Une fonctionnalité qui le démarque des autres est le vMotion. Elle permet de déplacer des machines virtuelles à chaud d'un serveur physique à un autre sans interruption de service.



infrastructure/../../../../\_static/infrastructure/hyperviseur/esxi.png

## Vsphere client

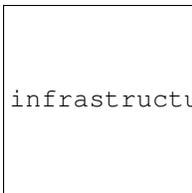
Logiciel permettant d'administrer un ESXi ainsi que ses machines virtuelles.



infrastructure/../../../../\_static/infrastructure/hyperviseur/vsphere.jpg

## Vcenter

C'est une appliance conçus par VMware qui permet de créer et manager un cluster d'ESXi (un hyperviseur d'hyperviseur) car dans notre cas, nous avons besoin d'assurer une haute disponibilité des services.

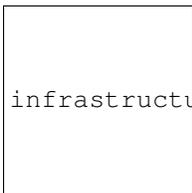


infrastructure/../../../../\_static/infrastructure/hyperviseur/vcenter.PNG

## Clustering

Depuis l'interface Vcenter, nous ajoutons les différents ESXi afin de pouvoir les gérer.

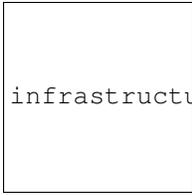
Nous créons ensuite un cluster de tous ces hyperviseurs en appliquant les meilleures pratiques pour faire de la haute disponibilité.



infrastructure/../../../../\_static/infrastructure/hyperviseur/cluster.PNG

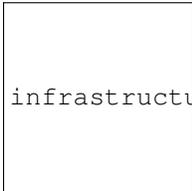
## Stockage

Des baies de stockage seront présentes dans chaque datacenter pour provisionner un stockage en ISCSI afin de stocker les différentes machines virtuelles. Ajout du stockage ISCSI sur un hyperviseur :



infrastructure/../../../../\_static/infrastructure/hyperviseur/stockage.PNG

## Vue globale



infrastructure/../../../../\_static/infrastructure/hyperviseur/hyperviseur.PNG

## Conteneur

Docker utilise simplement des fonctionnalités natives de Linux comme son noyau, ses LXC et cgroups

Docker permet de créer des environnements (appelées containers) de manière à isoler des applications.

Dans le cas de Docker, le kernel va partager les ressources du système hôte et interagir avec le(s) container(s). Techniquement, Docker n'est pas une VM, pas le moins du monde, mais en terme d'utilisation, Docker peut-être apparenté à une VM.

Comme le container n'embarque pas d'OS, à la différence de la machine virtuelle, il est par conséquent beaucoup plus léger que cette dernière. Il n'a pas besoin d'activer un second système pour exécuter ses applications. Cela se traduit par un lancement beaucoup plus rapide, mais aussi par la capacité à migrer plus facilement un container (du fait de son faible poids) d'une machine physique à l'autre. Autre avantage : les containers Docker, du fait de leur légèreté, sont portables de cloud en cloud.

## Cluster swarm

### Conception décentralisée.

Au lieu de gérer la différenciation entre les rôles de nœud au moment du déploiement, le Docker Engine gère toute spécialisation au moment de l'exécution. Vous pouvez déployer les deux types de nœuds, de MANAGER et de WORKER, en utilisant le Docker-engine.

### Modèle de service déclaratif

Docker Engine utilise une approche déclarative pour vous permettre de définir l'état désiré des différents services dans votre stack d'applications. Par exemple, vous pouvez décrire une application composée d'un service Web front-end avec des services de mise en file d'attente de messages et d'un backend de base de données.

## Scaling

Pour chaque service, vous pouvez déclarer le nombre de tâches que vous souhaitez exécuter. Lorsque vous augmentez ou descendez, le gestionnaire de swarm s'adapte automatiquement en ajoutant ou en supprimant des tâches pour maintenir l'état désiré.

### Conciliation de l'état désiré

Le nœud manager de swarm surveille constamment l'état du cluster et réconcilie toutes les différences entre l'état actuel et l'état souhaité exprimé. Par exemple, si vous configurez un service pour exécuter 10 replica d'un conteneur, et une worker qui héberge deux de ces répliques se bloque, le manager créera deux nouvelles replica pour remplacer les replica qui se sont écrasées.

Le manager Swarm attribue les nouvelles répliques aux travailleurs en cours d'exécution et disponibles.

### Réseau multi-hôte (overlay network)

Vous pouvez spécifier un overlay network pour vos services.

Le manager de swarm attribue automatiquement les adresses aux conteneurs sur le overlay network lorsqu'il initialise ou met à jour l'application.

### Découverte du service

Les nœuds du manager Swarm assignent à chaque service dans swarm un nom DNS unique et les load balances exécutant des conteneurs.

Vous pouvez interroger tous les conteneurs qui fonctionnent dans swarm à travers un serveur DNS intégré dans swarm.

### Rolling updates

Au moment du déploiement, vous pouvez appliquer les mises à jour du service aux nœuds de façon incrémentielle.

Le manager Swarm vous permet de contrôler le délai entre le déploiement du service sur différents ensembles de nœuds.

Si quelque chose ne va pas, vous pouvez roll-back a la version précédente.

### Etats du cluster

Exemple :

```
docker@docker-01:~$ docker node ls
ID                HOSTNAME        STATUS    AVAILABILITY    MANAGER STATUS
fuptalg81pxdsnpkwuewfa648    docker-02    Ready    Active          Reachable
rsjyyzupmv81ucaklydkp0rfl    docker-03    Ready    Active          Leader
sjxyentbtr71eal698jv38w7c    docker-05    Ready    Drain
xthdbocmuvdtgpuqrzq9v7 *    docker-01    Ready    Active          Reachable
```

## Storage

Pour le stockage on utilise MINIO : expliqué dans la parti stockage.

## Stack

Toutes les organisations sont créée avec docker-compose v3 en stack.

## Stockage et réplication

Toutes les données sont stockées sur un cluster `minio`

- Core `miniofs`
- CLI `mc`
- Docker `plugin`

## Minio

- S3 amazon

Avoir une compatibilité S3 amazon. Les plugins et les SDKs S3 sont assez nombreux sur le marché du cloud.

- Web ui

Une interface pour gérer les buckets.

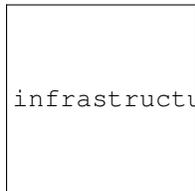
- Erasure Code

Minio protège les données contre les pannes de matériel et la corruption de données.

## Web Ui

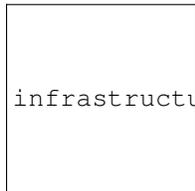
Pour la mises en cluster il est obligatoire d'avoir 4 noeud minio mini

Web interface



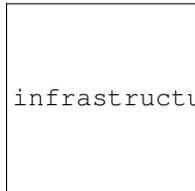
`infrastructure/../../../../_static/infrastructure/minio/web_1.png`

Share object



`infrastructure/../../../../_static/infrastructure/minio/web_2.png`

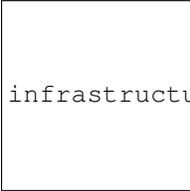
Public link



`infrastructure/../../../../_static/infrastructure/minio/web_3.png`

## Minio CLI

Installation et copie des fichiers dans le bucket *management*

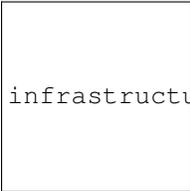


```
infrastructure/../../../../_static/infrastructure/minio/cli_1.png
```

## Integration avec docker

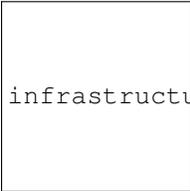
### Installation du plugin

(Très très très très simple ...)



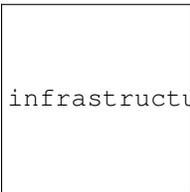
```
infrastructure/../../../../_static/infrastructure/minio/docker_1.png
```

### Déclaration d'un volume minio dans le docker-compose v3



```
infrastructure/../../../../_static/infrastructure/minio/docker_2.png
```

- Term 1 : client
- Term 2 : API\_1
- Term 3 : API\_2



```
infrastructure/../../../../_static/infrastructure/minio/docker_3.png
```

Les fichiers générés par l'API sont bien présent dans les deux conteneurs

## Monitoring de l'infrastructure

### Interne

- ElasticSearch

Puissant moteur de recherche utilisé par exemple par github

- Logstash  
Analyse des logs
- Kibana

Les données brutes sont analysées dans Logstash, stockées dans Elasticsearch, mais ne sont pas encore exploitables. Kibana est une interface Web qui se connecte au cluster Elasticsearch, et permet de faire des requêtes en mode texte pour générer des graphiques

- grafana  
Grafana vous permet d'interroger, visualiser, alerter et comprendre vos paramètres, peu importe où ils sont stockés.
- prometheus

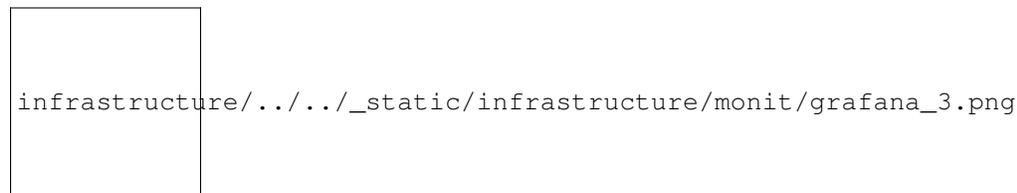
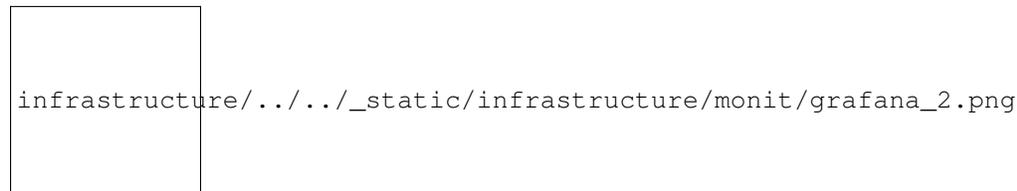
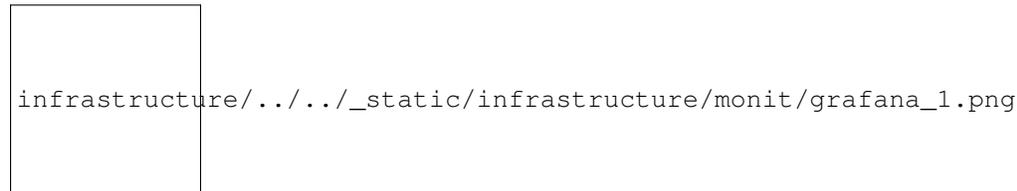
Prometheus est une boîte à outils de monitoring et d'alerte. Il s'agit maintenant d'un projet Open Source autonome et maintenu indépendamment de toute entreprise.

- node-exporter / docker-exporter  
L'exportateur de Prometheus pour les métriques de matériel et d'OS exposées par les nœuds.

## Grafana

grafana

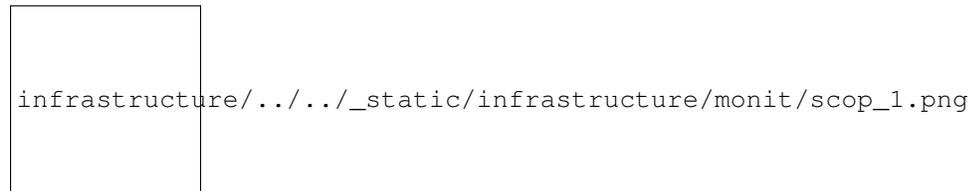
Les noeuds et les conteneur sont ajoutés/supprimés automatiquement a la supervision



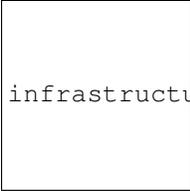
## Externe

cloud weave

Vue des conteneurs

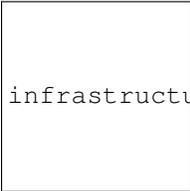


Vue des connexions réseaux In/out



infrastructure/../../../../\_static/infrastructure/monit/scop\_2.png

Vue des docker-hosts

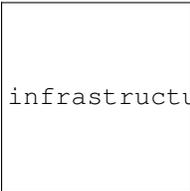


infrastructure/../../../../\_static/infrastructure/monit/scop\_3.png

## Load Balancing

### Load Balancing externe

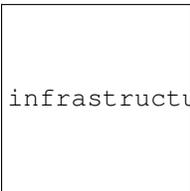
Le Load Balancing répartit la charge entre les adresses IP publique des différents noeud du cluster swarm sur les différents datacenters.



infrastructure/../../../../\_static/infrastructure/lb/external\_lb.png

### Load Balancing : Swarm mode routing mesh

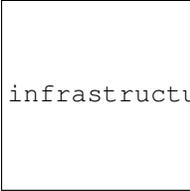
Le maillage de réseau interne de Swarm permet à chaque noeud du cluster d'accepter les connexions à tout port de service publié dans swarm en acheminant toutes les demandes entrantes vers les noeuds disponibles hébergeant un service avec le port publié.



infrastructure/../../../../\_static/infrastructure/lb/swarm\_lb.png

### Revers proxy / Load Balancing interne

Traefik Est un proxy HTTP inverse moderne et un équilibreur de charge conçu pour déployer des microservices avec facilité. Il supporte plusieurs backends (Docker, Swarm, Kubernetes, Marathon, Mesos, Consul, Etcd, Zookeeper, BoltDB, Eureka, Amazon DynamoDB, Rest API, fichier ...) pour gérer sa configuration automatiquement et dynamiquement.



```
infrastructure/../../../../_static/infrastructure/lb/internal_lb.png
```

Traefik écoute sur la sockets docker

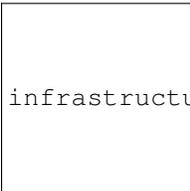
Selon les labels (swarm) du service, Traefik génère automatiquement la configuration de redirections

Exemple :

```
1  deploy:
2    labels:
3      traefik.port: "3000"
4      traefik.frontend.rule: "Host:${ORGANISATION}.popcube.xyz"
5      traefik.backend.loadbalancer.sticky: "true"
6      traefik.backend.loadbalancer.swarm: "true"
7      traefik.docker.network: "traefik-net"
```

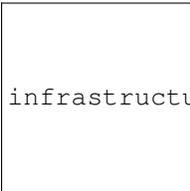
## Traefik

### Configuration



```
infrastructure/../../../../_static/infrastructure/lb/traefik_dashboard.png
```

### Statistiques



```
infrastructure/../../../../_static/infrastructure/lb/traefik_health.png
```

## Automatisation

Ansible est un outil open-source de gestion de configuration écrit en python (aussi dispo en version commerciale avec une interface graphique et un service de déploiement). La configuration se fait via des fichiers appelés “Playbooks”.

Les avantages :

- Un système déclaratif : syntaxe YAML facilement lisible, ce qui rend l’apprentissage très rapide.
- Templating des fichiers de configuration : qui permet d’avoir des fichiers dynamiquement générés en fonction de ce que vous voulez, tel que le rôle du serveur, ou bien dépendant d’un autre serveur. En plus le langage de template par défaut est Jinja2, ça plaira aux amateurs de Django.

Quasiment rien à installer. A part Ansible sur votre machine hôte, tout ce dont vous avez besoin c’est d’un accès root via SSH sur vos serveurs cibles.

## Provisionnement des machines virtuelles

Nous avons besoin d'automatiser la création de machines virtuelles car elles servent d'hôte pour les conteneurs Docker. Ainsi, nous pourrions créer autant de stack docker (dans la limite des ressources physiques disponibles).

Exemple de configuration :

```
- vsphere_guest:
  vcenter_hostname: vcenter.vsphere.local // On se connecte sur le vCenter qui
↳manage les esxi
  username: user
  password: password
  guest: vm_name
  state: powered_on
  vm_extra_config: // On active l'ajout à chaud CPU & RAM
    vcpu.hotadd: yes
    mem.hotadd: yes
    notes: docker-node
    folder: VM
  vm_disk: // On définit le stockage de la VM
    disk1:
      size_gb: 64
      type: thin
      datastore: ISCSI
      folder: VM
  vm_nic: // On définit le réseau auquel sera
↳connecté la VM
    nic1:
      type: vmxnet3
      network: VM Network
      network_type: standard
  vm_hardware: // On spécifie le matériel & l'image
↳iso utilisée
    memory_mb: 8192
    num_cpus: 2
    osid: debian8
    scsi: paravirtual
    vm_cdrom:
      type: "iso"
      iso_path: "Datastore/iso/debian-8.iso"
  esxi: // On indique sur quel ESXi hôte la vm
↳fonctionnera
    datacenter: Datacenter
    hostname: esx1.popcube.xyz
```

## Ajout aux cluster swarm

Pour rejoindre le cluster j'utilise le roles :

[Github du roles swarm](#)

Exemple Playbooks :

```
TASK [atosatto.docker-swarm : Install python-pip.]
↳*****
skipping: [sw01] => (item=(0, u'python-pip'))

TASK [atosatto.docker-swarm : Install the Python SNI support packages.]
↳*****
```

```

skipping: [sw01] => (item=python-dev)
skipping: [sw01] => (item=libssl-dev)
skipping: [sw01] => (item=libffi-dev)

TASK [atosatto.docker-swarm : Install the Python SNI python-pip dependencies.
↳] ***
skipping: [sw01] => (item=pyopenssl)
skipping: [sw01] => (item=ndg-httpsclient)
skipping: [sw01] => (item=pyasn1)

TASK [atosatto.docker-swarm : Import Docker APT public key.]_
↳*****
skipping: [sw01]

TASK [atosatto.docker-swarm : Install yum-utils if necessary.]_
↳*****
skipping: [sw01]

TASK [atosatto.docker-swarm : Install apt-transport-https if necessary.]_
↳*****
skipping: [sw01]

TASK [atosatto.docker-swarm : Add the YUM Docker repository.]_
↳*****
skipping: [sw01]

TASK [atosatto.docker-swarm : Add the APT Docker repository.]_
↳*****
skipping: [sw01]

TASK [atosatto.docker-swarm : Install Docker dependencies.]_
↳*****
skipping: [sw01] => (item=linux-image-extra-virtual)

TASK [atosatto.docker-swarm : Install the Docker Engine.]_
↳*****
skipping: [sw01]

TASK [atosatto.docker-swarm : Enable the Docker daemon as a service and_
↳start it.] ***
skipping: [sw01]

TASK [atosatto.docker-swarm : Install docker-py 1.9.0 to fix Ansible issue_
↳17495] ***
skipping: [sw01]

TASK [atosatto.docker-swarm : Install docker-py]_
↳*****
skipping: [sw01]

TASK [atosatto.docker-swarm : Add the Docker administrators to the Docker_
↳group] ***
skipping: [sw01] => (item=root)

TASK [atosatto.docker-swarm : Kill the ansible_user active SSH connections]_
↳****
skipping: [sw01]

```

```
TASK [atosatto.docker-swarm : Check if "Swarm Mode" is enabled.]_
↳*****
ok: [sw01]

TASK [atosatto.docker-swarm : Init "Swarm Mode" on the first manager.]_
↳*****
```

## Conteneur

### Réseaux et dns

Je ping la database de l'organisation de maxime :

```
docker@docker-02:~$ docker exec -it maxime_api.1.93ms44c6jdsllhazikwbdrki /bin/sh
/go/src/github.com/titouanfreville/popcubeapi # ping maxime_database
PING maxime_database (10.0.3.4): 56 data bytes
64 bytes from 10.0.3.4: seq=0 ttl=64 time=0.080 ms
64 bytes from 10.0.3.4: seq=1 ttl=64 time=0.092 ms
```

Depuis la database je ping le conteneur disponible :

```
docker@docker-02:~$ docker service scale maxime_api=5
docker@docker-03:~$ docker exec -it maxime_database.1.rrppb6d5qaed3cm8n7oa42qes /bin/
↳bash
root@1c32df28a830:/#
root@1c32df28a830:/# ping maxime_api
PING maxime_api (10.0.3.2): 56 data bytes
64 bytes from 10.0.3.2: icmp_seq=0 ttl=64 time=0.053 ms
64 bytes from 10.0.3.2: icmp_seq=1 ttl=64 time=0.100 ms
root@1c32df28a830:/# ping maxime_api.1.m19g9tbv4m6kx4794ilaoi0hh
PING maxime_api.1.m19g9tbv4m6kx4794ilaoi0hh (10.0.3.18): 56 data bytes

root@1c32df28a830:/# dig tasks.maxime_api

; <<>> DiG 9.9.5-9+deb8u11-Debian <<>> tasks.maxime_api
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 42947
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;tasks.maxime_api.          IN      A

;; ANSWER SECTION:
tasks.maxime_api.          600     IN      A       10.0.3.7
tasks.maxime_api.          600     IN      A       10.0.3.9
tasks.maxime_api.          600     IN      A       10.0.3.3
tasks.maxime_api.          600     IN      A       10.0.3.11
tasks.maxime_api.          600     IN      A       10.0.3.18

;; Query time: 0 msec
;; SERVER: 127.0.0.11#53(127.0.0.11)
;; WHEN: Mon May 29 12:23:16 UTC 2017
;; MSG SIZE rcvd: 194
```

Le nom de staks et service :

```
root@1c32df28a830:/# nslookup maxime_api
Server:      127.0.0.11
Address:     127.0.0.11#53

Non-authoritative answer:
Name:       maxime_api
Address:    10.0.3.2

root@1c32df28a830:/# nslookup tasks.maxime_api
Server:      127.0.0.11
Address:     127.0.0.11#53

Non-authoritative answer:
Name:       tasks.maxime_api
Address:    10.0.3.3
Name:       tasks.maxime_api
Address:    10.0.3.11
Name:       tasks.maxime_api
Address:    10.0.3.7
Name:       tasks.maxime_api
Address:    10.0.3.9
Name:       tasks.maxime_api
Address:    10.0.3.18
```

## Mise a jours des conteneurs

Exemple : ajout d'un labels :

```
docker@docker-02:~$ docker service update maxime_database --update-parallelism 2 --
↳with-registry-auth --container-label-add xyz.popcube.org=maxime
maxime_database

"UpdateStatus": {
  "State": "completed",
  "StartedAt": "2017-05-29T12:11:49.342150133Z",
  "CompletedAt": "2017-05-29T12:12:32.897318749Z",
  "Message": "update completed"
}
docker@docker-02:~$ docker service update maxime_api
"UpdateStatus": {
  "State": "updating",
  "StartedAt": "2017-05-29T12:36:21.96182509Z",
  "CompletedAt": "1970-01-01T00:00:00Z",
  "Message": "update in progress"
}

  ID          NAME          IMAGE          ERROR  PORTS
↳ NODE      DESIRED STATE  CURRENT STATE
fg54tw7l62lb maxime_api.1   registry.popcube.xyz:5000/popcubeapi:alpha-1.
↳1.11 docker-01 Running        Running 2 minutes ago
m19g9tbv4m6k  \_ maxime_api.1 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↳1.11 docker-03 Shutdown      Shutdown 3 minutes ago
93ms44c6jds1  \_ maxime_api.1 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↳1.11 docker-02 Shutdown      Shutdown 27 minutes ago
xyq4od3bl214  \_ maxime_api.1 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↳1.11 docker-01 Shutdown      Shutdown 3 days ago
```

```

vitoudmvt0nd maxime_api.2 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↪1.11 docker-03 Running Running 2 minutes ago
tnlhudgg18ve \_ maxime_api.2 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↪1.11 docker-01 Shutdown Shutdown 3 minutes ago
klvmqfyi3vn9 maxime_api.3 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↪1.11 docker-02 Running Running 2 minutes ago
p3opbb8iq86w \_ maxime_api.3 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↪1.11 docker-02 Shutdown Shutdown 3 minutes ago
n9wg7howeasg maxime_api.4 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↪1.11 docker-01 Running Running 2 minutes ago
f75oupbzedf1 \_ maxime_api.4 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↪1.11 docker-01 Shutdown Shutdown 3 minutes ago
uhha8n4lajcr maxime_api.5 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↪1.11 docker-02 Running Running 2 minutes ago
34sja7n3xcv5 \_ maxime_api.5 registry.popcube.xyz:5000/popcubeapi:alpha-1.
↪1.11 docker-02 Shutdown Shutdown 3 minutes ago

```

## Portabilité

Les images se déplace facilement entre les noeuds

Exemple de nos images docker :

```

registry.popcube.xyz:5000/popcubeapi : 332 MB
registry.popcube.xyz:5000/popcube_website : 18 MB

```

## Monitoring

Grace à l'écoute sur la socket les nodes exploreurs récoltent directement les nouveaux conteneurs qui sont créé.

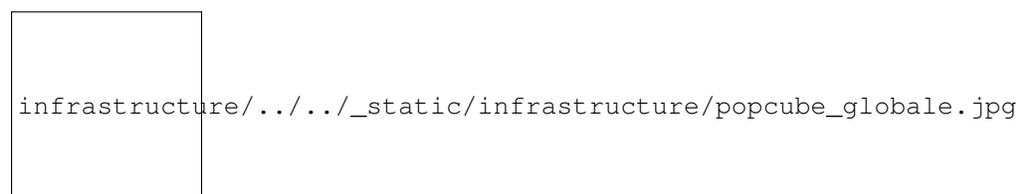
## Reverse Proxy

Selon les labels (swarm) du service, Traefik génère automatiquement la configuration de redirections.

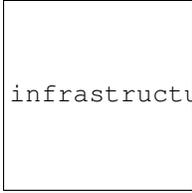
## Vue d'ensemble

Voici l'infrastructure globale

On retrouve les datacenters, les machines virtuels ainsi que les conteneurs



Demo d'un utilisateur qui demande la page web popcube.xyz



## Introduction

PopCube est l'application à utiliser si vous avez une petite ou grande équipes. Sa facilité de mise en place

### Qui somme nous ?

L'équipe du projet est constitué de 5 membres :

- Clément LE CORRE (DevOps / Docker designer & marketing)
- Maxime CORDIER (Front-end & marketing)
- Titouan FREVILLE (Backend API)
- Julien PETRIGNET (Infrastructure Design)
- Jeremy PERSON (Infrastructure Design)

### Popcube qu'est ce que c'est ?

PopCube est une application permettant la communication en sein de votre équipes. Ses points forts se basent sur la facilité de création d'une organisation (la création d'un serveur accueillant vos membres) et la multitude de channel possible afin de créer des groupes de travaux. Messageries instantanée, communication voix et vidéos, partages de fichiers sont les fonctionnalités première de PopCube. Vient ensuite la partie modération et gestion de l'organisation afin de vous permettre de gérer facilement et rapidement vos paramètres.

### On ne s'arrête pas la !

PopCube est développer de telle sorte que si vous avez plusieurs groupes de travail, il est possible de les avoirs directement sous une seule application ! Il n'est pas nécessaire d'avoir plusieurs fenêtre.

De plus nous avons des idées d'innovation dans un future proche qui vont vous ravir.

## Comment on gagne du cash

### Free

PopCube est disponible gratuitement pour toutes les équipes allant jusqu'à 30 membres. Cette option permet aux petites équipes d'avoir un logiciel leurs permettant de dialoguer et partager gratuitement. Elle permet aussi à tous ceux qui veulent s'y essayer de comprendre son fonctionnement sans avoir à payer.

Elle comprend :

- Un accès pour 30 membres (Comptes créés sur l'organisation)
- Possibilité d'ajouter 4bot

## Small

Cette option permet aux plus grosses équipes d'utiliser PopCube.

Elle comprend :

- Un accès jusqu'à 99 membres
- Possibilité d'ajouter jusqu'à 10 Bot
- Un accès aux assets des versions N-1

## Big

Aucune limite à mon Pouvoir !

Elle comprend :

- Aucune limite de membres
- Aucune limite de bot
- Accès à tous les nouveaux assets (voir *Assets*)
- Nom de domain personnalisé (nomdomain.popcube.xyz -> nomdomain.xyz)

## Assets

En plus des modèles économiques présenté ci-dessus il sera possible d'acheter des assets développer par nos soins.

- Emoticon
- Design différents pour l'organisation
- Avatar
- ...

## Prochainement dans PopCube

### Développement de la structure

#### Equipe

Nous n'avons pas besoin de ressources supplémentaire pour PopCube. Grâce à notre système de déploiement automatique nous n'avons besoin que d'un support pour nos serveur. En terme de développement, nous voulons garder la cohésion d'équipe que nous avons et permettre la déploiement soutenu de feature.

#### Réseaux

Notre objectif de cette année est de développer notre réseau de communication au sujet de PopCube. Twitter, facebook et notre forum vont venir vous accompagner pour répondre à vos questions et nous permettre d'être toujours proche de notre communauté afin de tirer le meilleurs de vos commentaires. Vos retours nous importe pour que PopCube reste au top de la communication !