

# Projet Cloud

Vous travaillez pour une entreprise qui souhaite proposer un système de Storage as a service (STaaS), permettant de mettre à disposition de ses clients un stockage de fichiers à distance. Vous êtes chargé d'élaborer une application web permettant aux clients de consulter et de gérer l'utilisation des disques qu'ils louent, et aux administrateurs de l'entreprise de visualiser les disques.

## Résumé

|                                    |   |
|------------------------------------|---|
| <b>Projet initial à utiliser</b>   | <ul style="list-style-type: none"><li>• <a href="#">Projet RT-Cloud v1.0 sur gitHub</a></li><li>• La base de données est dans le dossier app/database du projet</li></ul> |
| <b>Outils</b>                      | <ul style="list-style-type: none"><li>• <a href="#">Bootstrap</a></li><li>• <a href="#">Micro-framework</a></li></ul>   |
| <b>Principales fonctionnalités</b> | <ul style="list-style-type: none"><li>• Module client</li><li>• Module Administration</li></ul>   |
| <b>Livraison</b>                   | • Jusqu'au jour de la soutenance (Semaine du 9 mai), par gitHub, ou par Moodle (en fonction de votre chargé de TP)  |

## Ressources

- [Module M2105 - RT web dyna - TD/P 4](#)
  - [Prise en main Micro-framework](#)
  - [Documentation API Micro-framework et cloud](#)
  - [Twig documentation](#)
- 
- **Lire impérativement les Modalités de remise de votre travail**
  - Pour vous préparer à l'oral, consulter la [grille d'évaluation](#)

## Règles de gestion

Les clients peuvent disposer d'espaces de stockage (nommés disques) permettant de stocker leurs données. Chaque disque est loué à un certain tarif par le client ; le tarif comprend :

- Un prix (mensuel)
- Un quota (capacité de stockage exprimée dans une unité (o, Ko, Mo, Go ou To))
- Une marge de dépassement, à ne pas dépasser sous peine d'être facturé d'un surcoût.

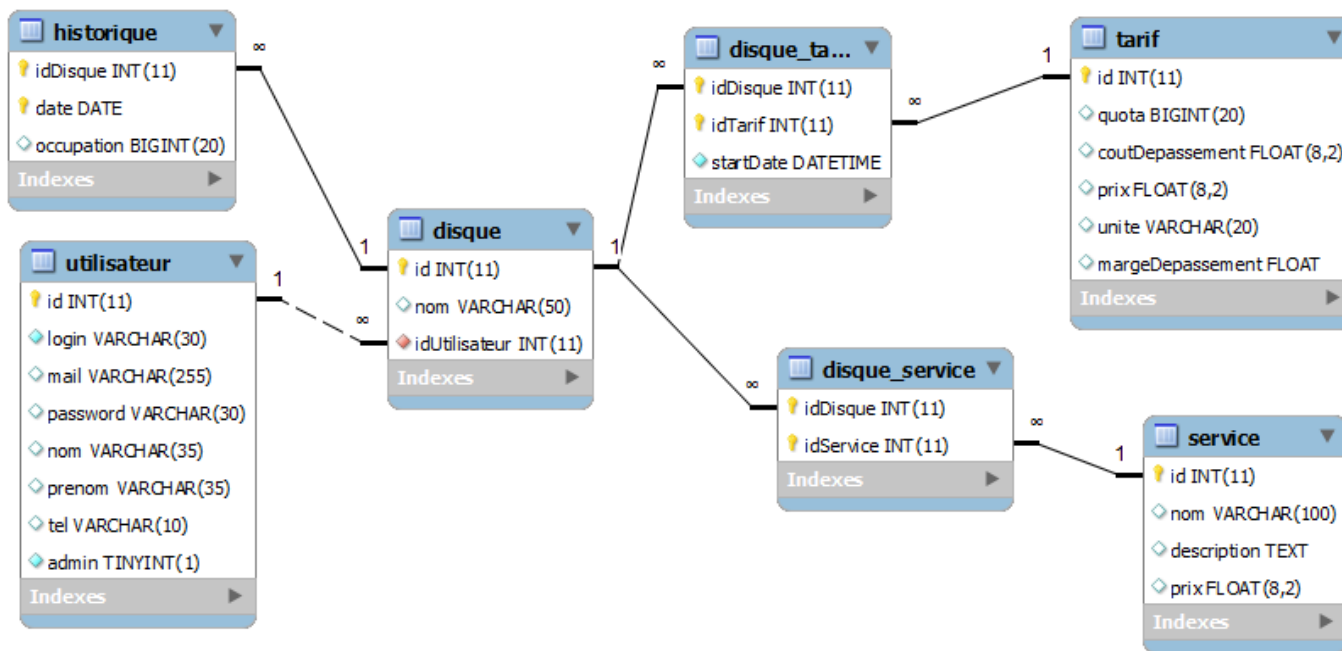
Le client a la possibilité de changer de tarif à tout moment pour chaque disque, de façon à diminuer ou augmenter son quota, en fonction de ses besoins. L'historique des tarifs associés à un disque est conservé.

Le tarif courant appliqué à un disque est celui correspondant à la date la plus récente.

Chaque location de disque peut être associée à des services (backup, loadBalancer...).

L'historique d'occupation des disques est obtenu de manière automatisée par un script PHP lancé de manière quotidienne par une tâche CRON.

## Schéma de la base



## Détail des fonctionnalités à mettre en place

Les différentes pages devront gérer les droits, et n'être autorisées que pour les acteurs mentionnés. Tenir compte du fait que l'administrateur a le droit d'accéder à tout ce qui est autorisé à l'utilisateur.

### -- Module Client

#### -- Affichage des disques d'un client

#### //TODO 1.1

A l'adresse **MyDisques/index**, on souhaite afficher les disques de l'utilisateur actuellement connecté.

L'utilisateur connecté est obtenu par l'appel de la méthode **Auth::getUser()**.



**Indications**

| Elément                      | Indications   |
|------------------------------|---|
| Contrôleur                   | MyDisques   |
| Action                       | index   |
| Utilisateur connecté         | L'utilisateur connecté est obtenu par l'appel de la méthode <b>Auth::getUser()</b> .  |
| Occupation, Quota            | le quota est obtenu sur le tarif actuel du disque, la classe <b>ModelUtils</b> permet de connaître l'occupation en cours du disque  |
| Composants visuels Bootstrap | on pourra utiliser les composants bootstrap <b>Progressbar, GlyphButton, Listgroup</b>  |
| Accès à la config du cloud   | Les disques clients sont localisés dans le dossier <b>/files</b> de l'application, et les disques y sont localisés sous le nom : <b>\$cloud-&gt;prefix-[disque.nom]</b> . Cette configuration est définie dans le fichier de <b>config.php</b> dans la variable <b>cloud</b> , accessible depuis les contrôleurs par <b>\$GLOBALS["config"]-&gt;cloud</b> |
| Tailles en octets            | Vous pourrez utiliser la méthode <b>sizeConverter</b> de <b>ModelUtils</b>  |

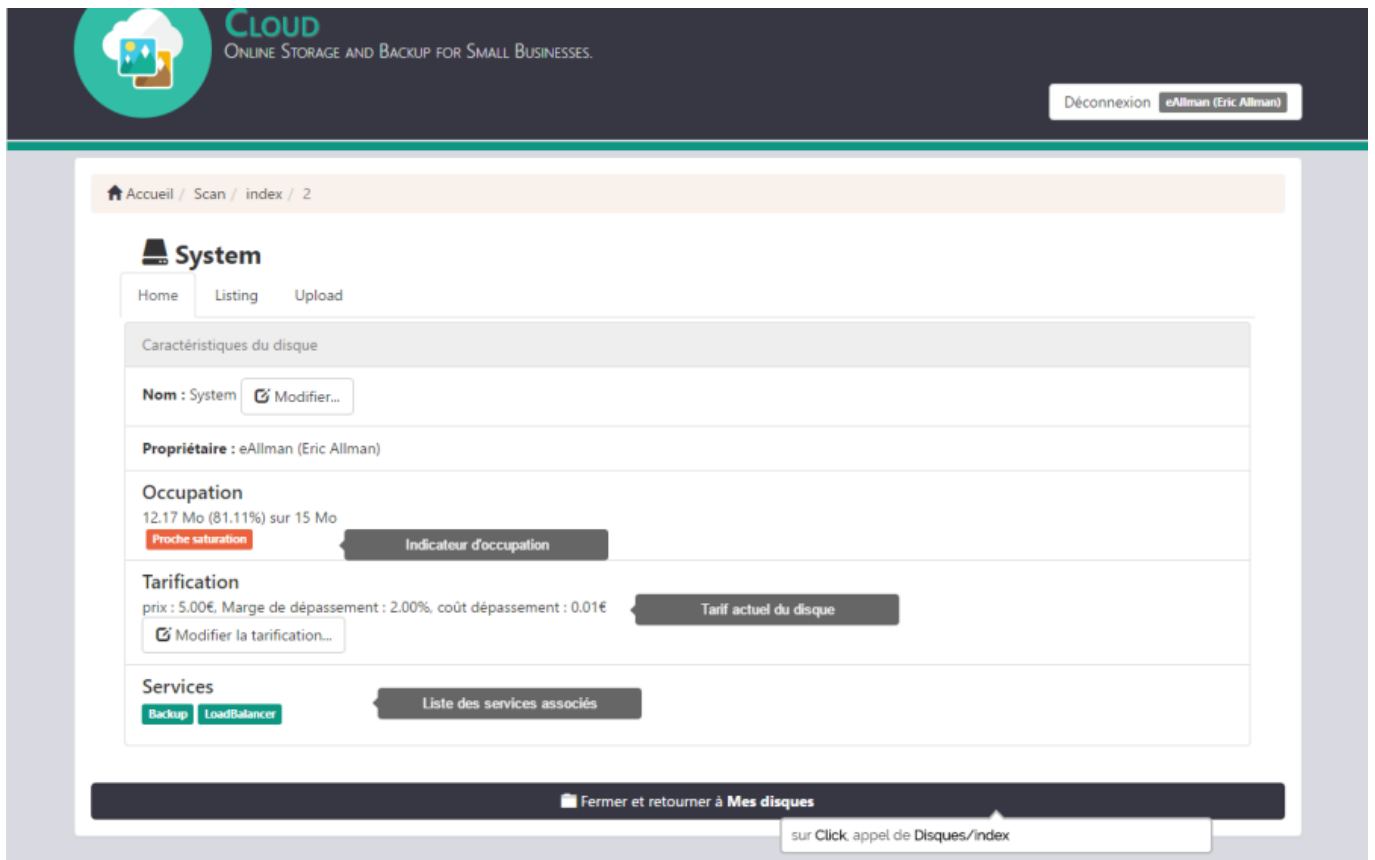
Le style des progressbars doit donner une indication sur le taux d'occupation en % :

| Style   | Valeurs     |
|---------|-------------|
| info    | de 0 à 10%  |
| success | de 10 à 50% |
| warning | de 50 à 80% |
| danger  | plus de 80% |

**-- Affichage d'un disque**

**//TODO 1.2**

A l'adresse **Scan/show/:idDisque**, afficher l'interface suivante :



| Élément                      | Indications  |
|------------------------------|--|
| Contrôleur                   | Scan   |
| Action                       | show   |
| Paramètre                    | idDisque   |
| Utilisateur connecté         | L'utilisateur connecté est obtenu par l'appel de la méthode <b>Auth::getUser(\$controller)</b> où <b>\$controller</b> est le contrôleur actif. |
| Occupation, Quota            | le quota est obtenu sur le tarif actuel du disque, la classe <b>ModelUtils</b> permet de connaître l'occupation en cours du disque             |
| Composants visuels Bootstrap | on pourra utiliser les composants bootstrap <b>Label, GlyphButton, Listgroup</b>   |

L'indicateur d'occupation (**Label**) doit donner une indication sur le taux d'occupation en % :

| Texte             | Style   | Valeurs     |
|-------------------|---------|-------------|
| Peu occupé        | info    | de 0 à 10%  |
| RAS               | success | de 10 à 50% |
| Forte occupation  | warning | de 50 à 80% |
| Proche saturation | danger  | plus de 80% |

### -- Création d'un disque

#### //TODO 1.3

A partir de l'adresse **Disques/frm**

Afficher le formulaire de création d'un disque.

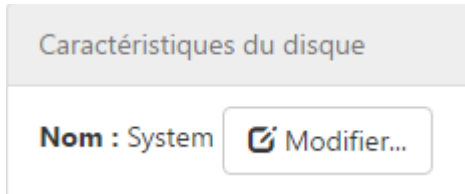
Soumettre le formulaire à l'adresse **Disques/update** pour créer le disque puis rediriger vers l'adresse **Scan/show/:idDisque**

Penser à le créer physiquement.

### -- Renommage disque

#### //TODO 1.4

A partir de l'adresse **Scan/show/:idDisque**



Implémenter la modification du nom d'un Disque (avec renommage éventuel du dossier associé + vérification du nom)

### -- Changement du tarif

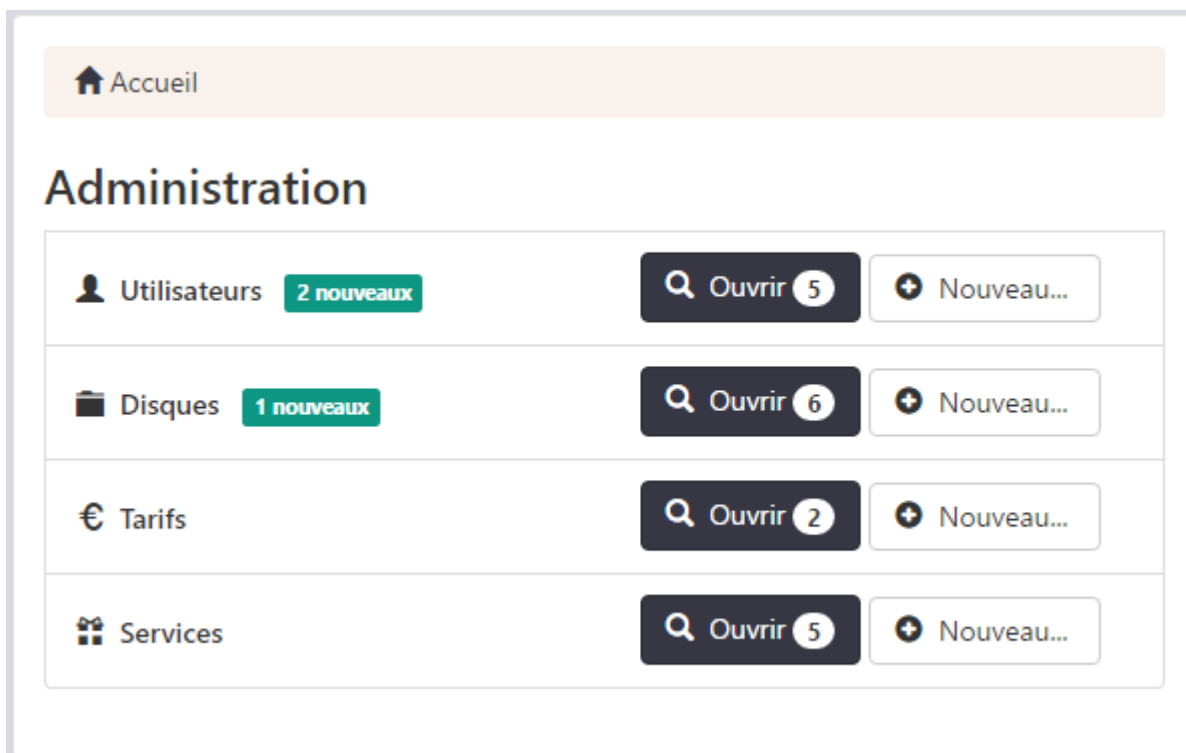
#### //TODO 1.5

Implémenter la modification du tarif par le client (avec vérification de non dépassement du quota)

### -- Module administration

#### -- Page d'accueil

#### //TODO 2.1



| Elément                | Indications   |
|------------------------|---|
| Contrôleur             | Admin   |
| Action                 | index   |
| Indication "nouveau"   | Affiche le nombre d'utilisateurs ou de disques créés dans la journée. |
| Badge du Bouton ouvrir | indique le nombre d'instances dans les tables                         |

-- Accès aux Utilisateurs

//TODO 2.2

The screenshot shows a web interface for 'Utilisateurs'. At the top, there is a breadcrumb 'Accueil / Utilisateurs'. Below the title 'Utilisateurs', there is a table with three columns: 'Utilisateur', 'Disques', and 'Montant'. The table contains five rows of user data. Each row has an 'Ouvrir' button on the right. At the bottom of the table, there is a 'Retour' button with a left arrow.

| Utilisateur   | Disques   | Montant |        |
|---------------|-----------|---------|--------|
| eAllman       | 3 disques | 10 €    | Ouvrir |
| jAppelbaum    | 3 disques |         | Ouvrir |
| kdMitnick     |           |         | Ouvrir |
| djberstein    |           |         | Ouvrir |
| linus.torvals |           |         | Ouvrir |

| Elément    | Indications |
|------------|-------------|
| Contrôleur | Admin       |
| Action     | users       |

-- Accès aux disques

//TODO 2.3

Accueil / Disques

## Disques

eric.allman@gmail.com-eAllman (Administrateur)

|          |  |  |        |
|----------|--|--|--------|
| Datas    | prix : 0.00€, Marge de dépassement : 2.00%, coût : 0.00€ | <div style="width: 50.25%; background-color: #28a745; height: 10px;"></div> 50.25% | Ouvrir |
| System   | prix : 5.00€, Marge de dépassement : 2.00%, coût : 0.01€ | <div style="width: 0%; background-color: #6c757d; height: 10px;"></div> 0%         | Ouvrir |
| Archives | prix : 5.00€, Marge de dépassement : 2.00%, coût : 0.01€ | <div style="width: 0%; background-color: #6c757d; height: 10px;"></div> 0%         | Ouvrir |

jacob.appelbaum@gmail.com-jAppelbaum (Utilisateur)

|             |  |  |        |
|-------------|--|--|--------|
| Jacob datas |  | <div style="width: 0%; background-color: #6c757d; height: 10px;"></div> 0% | Ouvrir |
| System      |  | <div style="width: 0%; background-color: #6c757d; height: 10px;"></div> 0% | Ouvrir |
| Server web  |  | <div style="width: 0%; background-color: #6c757d; height: 10px;"></div> 0% | Ouvrir |

[← Retour](#)

| Elément    | Indications   |
|------------|---|
| Contrôleur | Admin   |
| Action     | disques   |
| paramètre  | idUtilisateur(Facultatif : si précisé affiche les disques de l'utilisateur correspondant) |

### -- Module commun

### -- Contrôle d'accès

#### //TODO 3.1

| URL                  | Accès   | Résultat/message   |
|----------------------|---|--|
| Admin/*              | Accessible uniquement pour un Administrateur connecté                     | Afficher un message d'erreur "Accès à une ressource non autorisée"   |
| Disques/index        | Accessible uniquement pour un utilisateur connecté                        | Afficher message sur l'absence de connexion et proposer la connexion |
| Scan/index/:idDisque | Accessible uniquement pour un disque appartenant à l'utilisateur connecté | Afficher un message d'erreur "Accès à une ressource non autorisée"   |

**Remarque :** L'administrateur hérite des droits de l'utilisateur.

### -- Fil d'ariane

#### //TODO 3.2

Le fil d'ariane (Breadcrumbs bootstrap) doit indiquer à tout moment la localisation de l'utilisateur à partir de sa page d'accueil :

#### Exemple :

## Contraintes techniques

- L'application sera développée en PHP objet, et utilisera un [micro-framework](#) facilitant les échanges avec la base de données.
- Elle respectera au mieux la séparation des couches (objets Métiers), classes techniques et vues (interfaces web de saisie et d'affichage).
- Elle utilisera la base de données Mysql fournie en annexe. Cette base pourra évoluer en fonction des besoins du développement.
- L'utilisation de scripts côté client (javascript et ajax) pourra compléter les validations côté serveur. L'utilisation du micro-framework est obligatoire, d'autres librairies ou frameworks peuvent le compléter..
- [Bootstrap](#) sera utilisé pour la partie présentation.

## Fichiers

- [Projet Github initial à cloner](#)
- [Grille d'évaluation projet](#)

## Modalités de remise du travail

- Date remise : rendre projet PHP + readme.md voir [Modalités de remise de votre travail](#)
- Date passage : soutenances

## Déroulement de l'oral

### Durée

- 5 minutes max de présentation par membre de l'équipe
- 5 minutes max de questions

### Contenu

Il s'agit de montrer, le travail effectué, ainsi que les concepts maîtrisés :

- En présentant les fonctionnalités implémentées (démonstration du fonctionnement)
- En donnant des explications techniques sur le fonctionnement (contrôleurs, vues, classes, sécurisation...)

## Compléments

### Bonnes pratiques

- Alimenter correctement la base de données en ajoutant des enregistrements valides et en nombre suffisant, mettant en valeur votre travail
- respecter la Normalisation HTML 5/Css 3

- Structurer les fichiers et dossiers de manière cohérente
- Nommer en respectant les normes et de manière significative (Contrôleurs, vues, méthodes, variables...)

From:

<http://slamwiki2.kobject.net/> - **SlamWiki 2.1**

Permanent link:

<http://slamwiki2.kobject.net/php-rt/projets/projet-2016>

Last update: **2019/08/31 14:21**

