**.** 

EMSI

# École marocaine des sciences de l'ingénieur

INGÉNIERIE INFORMATIQUE ET RÉSEAUX

Mémoire de projet de fin d'étude

Présenté et soutenue par :

OUDANKAI MOUSSA

**Conception et implémentation d'un environnement de production pour une architecture micro services sur azure cloud et on-premise**

Sous l’encadrement de :

EMSI : M. FILALI MOUHANED

CBI: M. KERTAOUI Khalil

Jury :

Soutenue le 18 juillet 2023

|  |  |
| --- | --- |
| Président | Pr. RIAD IDRISSI |
| Encadrante | Pr. Filali Mouhanad |

Année universitaire 2022/2023

## Remerciements

En préambule à ce mémoire, je souhaite adresser mes remerciements les plus sincères aux personnes qui m’ont apporté leur aide et qui ont contribué à l'élaboration de ce rapport ainsi qu’à la réussite de mon formidable cursus universitaire.

Je veux donc en premier lieu remercier mon professeur et encadrante pédagogique à École marocaine des sciences de l'ingénieur (EMSI), Monsieur FILALI Mouhaned pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter mon réflexion.

Je tiens à remercier vivement notre maitre de stage, Monsieur Khalil KERTAOUI, pour son accueil, le temps passé ensemble et le partage de son expertise au quotidien. Grâce aussi à sa confiance j’ai pu m’accomplir totalement dans mes missions. Elle fut d'une aide précieuse dans les moments les plus délicats. Ainsi, au personnel de CBI, qui m’a aidé durant ce stage afin de me procurer le maximum possible de connaissances et de renseignements sur la profession.

Un grand merci à ma famille pour leur amour, leurs conseils ainsi que leur soutien inconditionnel, à la fois moral et économique, qui m’a permis de réaliser ce mémoire.

Enfin, j’adresse mes plus sincères remerciements à tous mes proches et amis, qui m’ont toujours soutenu et encouragé tout au cours de notre démarche.

Merci à tous.

### Résumé

Dans le cadre de mon projet de fin d’études à École marocaine des sciences de l'ingénieur (EMSI), en vue de l’obtention du diplôme de Master en ingénierie informatique et réseaux, nous sommes tenus de réaliser un projet de fin d'études au sein d'une organisation qui nous accueille, afin d'approfondir et d'appliquer les connaissances acquises durant notre formation et de promouvoir le travail confié.

Le présent document constitue la synthèse de mon travail dans le cadre de ce projet de stage, effectué au sein de CBI, à Casablanca.

On a eu la chance durant cette expérience de travailler sur un projet de modernisation des applications au niveau du Cloud et On-premise.

Notre projet consiste sur le DevOps, qui pourrait être considéré comme une méthodologie Agile portée à un niveau supérieur de flexibilité et d'efficacité. L’adaptation de DevOps par une entreprise peut avoir plusieurs impacts positifs sur son fonctionnement et ses résultats comme l’accélération du cycle de développement et amélioration de la qualité du logiciel.

Nous mettrons en place un système de contrôle de version et des repos de code, ainsi que des pipelines CI/CD pour assurer une intégration et une livraison continues de notre application web. Cette application sera basée sur une architecture de microservices et nous utiliserons des technologies telles que Java, React, Python et MySQL pour son développement.

Pour concrétiser nos discussions précédentes, nous faisons appel à Azure DevOps, Jenkins et ArgoCD. Cela nous permettra de mettre en œuvre nos processus de développement et de déploiement de manière efficace et cohérente.

Nous utiliserons Azure Cloud pour utiliser les services de Microsoft et les technologies telles que Docker, Kubernetes, Prometheus et Grafana afin de déployer et de surveiller notre application.

**Mots Clés :** DevOps, CI/CD, Java, React, Python, MySQL, Azure DevOps, ArgoCD, Jenkins, Azure Cloud, Docker, Kubernetes, Prometheus, Grafana

### Abstract

Within the end of my studies at the Moroccan School of Engineering (EMSI), in order to obtain the diploma of Master in computer networks engineering and intelligent telecommunications systems, we are required to carry out an end of studies project within an organization which welcomes us, in order to deepen and apply the knowledge acquired during our training and to promote the entrusted work.

The present document is the synthesis of my work within this internship project, carried out within CBI, in Casablanca.

We had the chance during this experience to work on a project of modernization of applications at the level of Cloud and On-premise.

Our project consists of DevOps, which could be considered as an Agile methodology taken to a higher level of flexibility and efficiency. The adaptation of DevOps by a company can have several positive impacts on its operations and results such as accelerating the development cycle and improving software quality.

We will implement a version control system and code repos, as well as CI/CD pipelines to ensure continuous integration and delivery of our web application. This application will be based on a microservices architecture and we will use technologies such as Java, React, Python and MySQL for its development.

To bring our previous discussions to life, we are using Azure DevOps, Jenkins and ArgoCD. This will allow us to implement our development and deployment processes in an efficient and consistent manner.

We will use Azure Cloud to leverage Microsoft services and technologies such as Docker, Kubernetes, Prometheus and Grafana to deploy and monitor our application.

**Key Words:** DevOps, CI/CD, Java, React, Python, MySQL, Azure DevOps, Jenkins, ArgoCD, Azure Cloud, Docker, Kubernetes, Prometheus, Grafana

### Table des figures

[Figure 1 : Les 5 métiers CBI 12](#_bookmark5)

[Figure 2 : CBI en chiffres 13](#_bookmark7)

[Figure 3 : Marque EasyCloud de CBI 13](#_bookmark9)

[Figure 4 : CBI Cisco Gold Partner 14](#_bookmark11)

[Figure 5 : CBI North West Africa Growth Partner of the Year 2021 15](#_bookmark12)

[Figure 6 : CBI Microsoft Gold Partner 15](#_bookmark13)

[Figure 7 : Implantations de CBI 16](#_bookmark15)

[Figure 8 : Partenaires de CBI 17](#_bookmark17)

[Figure 9 : Clients de CBI 17](#_bookmark19)

[Figure 10 : Axes du projet 18](#_bookmark23)

[Figure 11 : Diagramme de Gantt 20](#_bookmark26)

[Figure 12 : Cycle de vie DevOps 26](#_bookmark40)

[Figure 13 : Logo K8s 28](#_bookmark50)

[Figure 14 : Architecture d'InnoDB Cluster 30](#_bookmark61)

[Figure 15 : Logo Microsoft Azure 32](#_bookmark64)

[Figure 16 : Logo Azure DevOps 32](#_bookmark66)

[Figure 17 : Logo Azure Boards 33](#_bookmark68)

[Figure 18 : Logo Azure Repos 33](#_bookmark70)

[Figure 19 : Logo Test Plans 33](#_bookmark72)

[Figure 20 : Logo Azure Artifact 34](#_bookmark74)

[Figure 21 : Logo Azure Pipelines 34](#_bookmark76)

[Figure 22 : Architecture Azure 35](#_bookmark79)

[Figure 23 : Architecture K8s Azure 35](#_bookmark81)

[Figure 24 : Architecture On-premise 37](#_bookmark87)

[Figure 25 : Architecture K8s On-premise 37](#_bookmark89)

[Figure 26 : Logo Spring Java 40](#_bookmark97)

[Figure 27 : Logo ReactJS 40](#_bookmark99)

[Figure 28 : Logo Python 40](#_bookmark101)

[Figure 29 : Logo MySQL 41](#_bookmark103)

[Figure 30 : Logo Visual Studio 41](#_bookmark106)

[Figure 31 : Logo PyCharm 41](#_bookmark108)

[Figure 32 : DockerFile 44](#_bookmark114)

[Figure 33 : Operations du document 48](#_bookmark122)

[Figure 34 : Operations d"OCR 48](#_bookmark123)

[Figure 35 : Operations de report 48](#_bookmark124)

[Figure 36 : Logo d'Ansible 50](#_bookmark131)

[Figure 37 : Logo de Github 51](#_bookmark133)

[Figure 38 : Logo de Jenkins 52](#_bookmark135)

[Figure 39 : Logo d'ArgoCD 54](#_bookmark139)

[Figure 40 : Interface Login 55](#_bookmark142)

[Figure 41 : Interface d'authentification de Microsoft 56](#_bookmark144)

[Figure 42 : Interface d'accueil 56](#_bookmark146)

[Figure 43 : Interface d'Upload Facture 57](#_bookmark148)

[Figure 44 : Interface Tableau de bord 57](#_bookmark150)

[Figure 45 : Interface Chatbot 58](#_bookmark152)

**Liste des tableaux**

[Tableau 1 : Details de chaque métier CBI 12](#_bookmark6)

[Tableau 2 : Les avantages et les inconvénients du Cloud Azure 36](#_bookmark83)

[Tableau 3 : Les avantages et les inconvénients d'on-premise 38](#_bookmark91)

[Tableau 4 : Les taches de notre projet 43](#_bookmark112)

### Table de matière

[Résumé 1](#_bookmark0)

[Abstract 2](#_bookmark1)

[Introduction générale 10](#_bookmark2)

[Chapitre 1 11](#_bookmark3)

1. [Présentation générale de l’entreprise d’accueil CBI 12](#_bookmark4)
   1. [Marque EasyCloud 13](#_bookmark8)
   2. [Certifications 14](#_bookmark10)
   3. [Implantations 15](#_bookmark14)
   4. [Partenaires 16](#_bookmark16)
   5. [Clients 17](#_bookmark18)
2. [Cadre du projet 18](#_bookmark20)
   1. [Fiche du projet 18](#_bookmark21)
   2. [Axes du projet 18](#_bookmark22)
   3. [Méthodologie adaptée 19](#_bookmark24)
   4. [Planning du projet 19](#_bookmark25)
3. [Conclusion 20](#_bookmark27)

[Chapitre 2 21](#_bookmark28)

1. [Analyse de l’existant 22](#_bookmark29)
2. [Problématique 22](#_bookmark30)
3. [Solution proposée 22](#_bookmark31)
4. [Description du projet 23](#_bookmark32)
5. [Pourquoi Cloud Azure et On-premise 23](#_bookmark33)
6. [Conclusion 23](#_bookmark34)

[Chapitre 3 24](#_bookmark35)

1. [Conteneurisation 25](#_bookmark36)
2. [le DevOps 25](#_bookmark37)
   1. [Définition 25](#_bookmark38)
   2. [Cycle de vie des applications 26](#_bookmark39)
      1. [Plan 26](#_bookmark41)
      2. [Code 26](#_bookmark42)
      3. [Build 26](#_bookmark43)
      4. [Test 26](#_bookmark44)
      5. [Release & deploy 27](#_bookmark45)
      6. [Operate & monitor 27](#_bookmark46)
3. [CI/CD 27](#_bookmark47)
4. [Kubernetes 28](#_bookmark48)
   1. [Définition 28](#_bookmark49)
   2. [Les avantages de K8s 28](#_bookmark51)
   3. [Cluster K8s 29](#_bookmark52)
   4. [Les composants d’un Cluster K8s 29](#_bookmark53)
      1. [Node 29](#_bookmark54)
      2. [Namespace 29](#_bookmark55)
      3. [Deployment 29](#_bookmark56)
      4. [Pod 29](#_bookmark57)
      5. [Service 29](#_bookmark58)
      6. [Ingress 30](#_bookmark59)
      7. [InnoDB Cluster 30](#_bookmark60)
5. [Cloud computing 31](#_bookmark62)
6. [Microsoft Azure 32](#_bookmark63)
   1. [Azure DevOps 32](#_bookmark65)
      1. [Azure Boards 33](#_bookmark67)
      2. [Azure Repos 33](#_bookmark69)
      3. [Azure Test Plans 33](#_bookmark71)
      4. [Azure Artifact 33](#_bookmark73)
      5. [Azure Pipelines 34](#_bookmark75)
   2. [Azure Portal 34](#_bookmark77)
   3. [Architecture Azure 35](#_bookmark78)
   4. [Architecture Kubernetes 35](#_bookmark80)
   5. [Avantages et Inconvénients 36](#_bookmark82)
7. [On-premises 36](#_bookmark84)
   1. [Definition 36](#_bookmark85)
   2. [Architecture On-Premises 37](#_bookmark86)
   3. [Architecture microK8s 37](#_bookmark88)
   4. [Avantages et Inconvénients 38](#_bookmark90)
8. [Conclusion 38](#_bookmark92)

[Chapitre 4 39](#_bookmark93)

1. [l’application 40](#_bookmark94)
   1. [Languages et Framework 40](#_bookmark95)
      1. [Spring java 40](#_bookmark96)
      2. [ReactJS 40](#_bookmark98)
      3. [Python 40](#_bookmark100)
      4. [MySQL 41](#_bookmark102)
   2. [les outils utilisés: 41](#_bookmark104)
      1. [Visual Studio 41](#_bookmark105)
      2. [PyCharm 41](#_bookmark107)
2. [Cloud Azure 42](#_bookmark109)
   1. [Azure Devops 42](#_bookmark110)
      1. [Azure Board 42](#_bookmark111)
      2. [Azure repos 43](#_bookmark113)
      3. [Azure pipelines 45](#_bookmark115)
      4. [Service connection 45](#_bookmark116)
   2. [Azure portal 45](#_bookmark117)
      1. [ACR 45](#_bookmark118)
      2. [AKS 46](#_bookmark119)
      3. [Azure AD 47](#_bookmark120)
      4. [API management 47](#_bookmark121)
      5. [Azure App Service 48](#_bookmark125)
      6. [Chatbot 49](#_bookmark126)
      7. [Key Vaults 49](#_bookmark127)
      8. [Monitoring (Grafana & Prometheus) 49](#_bookmark128)
3. [On-premise 50](#_bookmark129)
   1. [Ansible 50](#_bookmark130)
   2. [Github 51](#_bookmark132)
   3. [Jenkins 51](#_bookmark134)
   4. [Docker Hub 52](#_bookmark136)
   5. [MicroK8s 53](#_bookmark137)
   6. [ArgoCD 53](#_bookmark138)
4. [Application dans Azure 54](#_bookmark140)
   1. [Interface login 55](#_bookmark141)
   2. [Interface d’authentification 55](#_bookmark143)
   3. [Interface D’accueil 56](#_bookmark145)
   4. [Interface Upload Facture 56](#_bookmark147)
   5. [Interface tableau de bord 57](#_bookmark149)
   6. [Interface chatbot 57](#_bookmark151)
5. [Conclusion 58](#_bookmark153)

[Conclusion Générale 59](#_bookmark154)

[Webographie & Bibliographie 60](#_bookmark155)

### Introduction générale

Dans un monde de plus en plus connecté et axé sur les données, l’IT est devenue une fonction vitale pour les entreprises de toutes tailles et de tous les domaines . Les entreprises doivent rester à jour avec les meilleures pratiques et les dernières tendances pour tirer la meilleure partie de ces approches et rester compétitives sur le marché technologique en constante évolution.

Le domaine de l’IT englobe de nombreux sous-domaines, tels que le cloud computing. C’est un modèle de fourniture de services informatiques via Internet. Il offre une alternative souple et rentable à la gestion traditionnelle des infrastructures informatiques. Il permet aux entreprises et aux utilisateurs finaux d’accéder a des ressources informatiques puissantes, flexibles et évolutives, tout en réduisant les couts et en améliorant l’efficacité opérationnelle.

D’autre part, l’infrastructure sur site (on-premise) implique que l’entreprise possède et gère elle-même ses propres serveurs, équipements réseau et systèmes de stockage. Cela signifie que les ressources informatiques sont installées et maintenues localement dans les locaux de l’entreprise.

Le cloud computing offre une approche flexible et évolutive basée sur des services en ligne, tandis que on-premise implique la possession et la gestion directe de l’infrastructure informatique. Le choix entre les deux dépend des besoins, des exigences et des objectifs spécifiques de chaque entreprise.

Hormis, l’introduction générale et la conclusion, notre travail est divisé en quatre chapitres :

Tout d´abord, je présenterais l'entreprise d'accueil. Par la suite je situerai le projet dans son contexte général et je parlerai d’une étude descriptive de la conception du projet. Puis, je vais décrire les solutions et les outils utilisés pour atteindre les objectifs du projet. Enfin, j’aborderai la phase de réalisation qui sera consacrée à l’environnement de travail et aux apports du projet.

# Chapitre 1

## PRESENTATION DU CADRE DE

**STAGE**

Le but du premier chapitre est de mettre le projet dans son contexte organisationnel et contextuel. Il comporte une présentation de l’organisme d’accueil CBI où j’ai effectué mon stage de fin d’étude, une description du cadre du projet ainsi que la partie de planification adoptée pour la réalisation du projet

#### Présentation générale de l’entreprise d’accueil CBI

CBI, créé en 1970, est un intégrateur global de solutions IT, expert à toutes les étapes de cycle de vie de l’information, est une société de services informatiques qui déploie des solutions à forte valeur ajoutée en 5 métiers : Télécoms, Sécurité, Systèmes, Software et Editique.



**Figure 1 : Les 5 métiers CBI**

|  |  |
| --- | --- |
| **Télécoms** | La mise en réseau et en collaboration de toutes les ressources informatiques. |
| **Sécurité** | La sécurité et la fiabilité de l’ensemble du système d’information et des données. |
| **Système** | La mise en place de toute l’infrastructure informatique qu’elle soit physique, hybride, virtualisée ou convergée. |
| **Software** | L’intégration de progiciels et briques métiers pour gérer, optimiser et générer de la valeur, notamment, mieux satisfaire les clients, mieux  produire et mieux investir. |
| **Editique** | L’édition papier et/ou numérique de vos documents et livrables, la gestion de votre politique documentaire et de vos flux d’impression. |

**Tableau 1 : Details de chaque métier CBI**

CBI est un capital humain hautement qualité compte plus de 270 salariés qui conseillent et accompagnent ses clients pour optimiser leur IT, maximiser leur productivité et performer, ainsi plus de 530 certifications auprès des constructeurs/éditeurs leaders mondiaux du marché.

**Figure 2 : CBI en chiffres**

De l'audit et du conseil jusqu'à la gestion des clés en main de toute l'infrastructure informatique, en passant par l'intégration et la formation, l'entreprise intervient à toutes les étapes du cycle de vie des projets.

CBI déjà certifié Microsoft gold Partner, est le premier intégrateur Cloud Solutions Provider (CSP) Tier1 au Maroc. CBI est fournisseur direct de toutes les briques Cloud Microsoft, aussi bien la messagerie Microsoft Exchange Online, la suite bureautique Office 365, le portail SharePoint, les outils de communication Skype et Teams ou encore les scénarios adaptés Microsoft Azure. Grâce à ce statut avantageux, CBI fournit les licences Cloud à ses clients à un tarif très préférentiel, le plus bas du marché.

##### Marque EasyCloud

Fort de ce positionnement et de cette expertise uniques, CBI a lancé sa marque Cloud ;

« EasyCloud », et ses boutiques en ligne : Easycloud.ma, Easycloud.sn et Easycloud.ci.



**Figure 3 : Marque EasyCloud de CBI**

Leurs Marketplace permettent :

* D’accéder à des solutions Cloud avec un support et un accompagnement local à des *sla*

très *exigeants ;*

* De bénéficier de solutions performantes, sans investissement lourd, et donc de rester

*agile* et *flexible* ;

* De pérenniser son entreprise en sécurisant son capital le plus précieux, ses *données ;*
* D’améliorer sa productivité en utilisant des solutions de collaboration efficaces et hautement disponibles ;

Avec EasyCloud en tant que CSP Tier 1, l’entreprise prend en charge les projets Cloud de bout en bout de ses clients, de l’attribution des licences jusqu’à leur réalisation ﬁnale.

Pour simplifier la gestion de leurs services Cloud pour les clients, CBI a créé "Easy Cloud Platform", une plateforme dotée de nombreuses fonctionnalités, y compris :

* Création, modification, suppression des utilisateurs ;
* Affectation des services aux nouveaux utilisateurs ;
* Gestion des licences des clients ;
* Accès à la Marketplace ;
* Accès au portail Office 365.

##### Certifications :

**En septembre 2021**, Cisco a reconfirmé sa certification GOLD de CBI pour la neuvième année consécutive, confirmant leur expertise dans l'accompagnement de leurs clients dans les domaines de la sécurité, de la collaboration et de la technologie du réseau. Il démontre leur savoir-faire et démontre leur capacité à intégrer, à supporter, à exploiter et à gérer des projets structurants basés sur ces technologies.

**Figure 4 : CBI Cisco Gold Partner**

**En octobre 2021**, CBI a été élu « North West Africa Growth Partner of the Year 2021 » par Nutanix, Ce titre vient couronner un **partenariat fructueux** de longue date entre CBI & Nutanix qui a débuté en 2015 et durant lequel CBI s’est distinguée à plusieurs reprises. En effet, en 2018 CBI a reçu le prix du « **Premier Partner** » pour atteindre en 2020 le niveau de partenariat le plus élevé « **Cloud Champion** ».



**Figure 5 : CBI North West Africa Growth Partner of the Year 2021**

**En novembre 2021**, a été la seule entreprise du Maroc qui a été choisie comme « Microsoft Partner of The Year ». Ce titre met en valeur l'expertise exceptionnelle de leurs équipes dans toute la technologie Microsoft. De nombreux clients ont bénéficié de la mise en œuvre de solutions innovantes qui leur ont permis de gagner en productivité et en agilité. Depuis 2018, CBI entretient une relation de partenariat solide avec Microsoft pour développer une expertise solide au service de ses clients.



##### Implantations :

**Figure 6 : CBI Microsoft Gold Partner**

CBI dispose d’un vaste réseau d’agences qui quadrille l’ensemble du territoire national :

* Le siège de CBI et l'agence Casablanca se trouvent dans la région du Grand Casablanca. Zola, Emile
* Agence Rabat Hay Ryad dans la région de Rabat-Salé-Kenitra.
* La région nord et orientale est représentée par l'agence de Tanger.
* La région centre est représentée par l'agence de Fès.
* La région sud est représentée par les agences de Marrakech et Agadir.

Grâce à ses implantations au Sénégal et en Côte d'Ivoire, il est en mesure de mettre en œuvre des projets dans ces deux pays, ce qui lui permet de couvrir toute l'Afrique de l'Ouest et de l'Afrique Centrale.

* Le Sénégal est couvert par la Mauritanie, le Mali, la Guinée, la Guinée-Bissau, la Sierra Leone et le Niger.
* La Côte d'Ivoire est couverte par les pays suivants : Burkina Faso, Ghana, Togo et Bénin.



**Figure 7 : Implantations de CBI**

CBI est également présent au Gabon depuis de nombreuses années, pays où il installera très bientôt une nouvelle implantation qui lui permettra de mener des projets dans les pays suivants : le Cameroun, le Congo Brazzaville et le Centrafrique.

##### Partenaires :

CBI a élaboré plusieurs relations de partenariat avec des dizaines des entreprises IT notamment : Cisco, IBM, Lenovo …, La figure ci-dessous représente une liste non exhaustive des partenaires CBI :



**Figure 8 : Partenaires de CBI**

##### Clients :

CBI dispose d’un portefeuille client regroupant des dizaines des entreprises des secteurs privés ainsi que les organismes publics, on peut citer à titre d’exemple : Atos, Amendis, Al barid bank, bank al maghreb. La figure ci-dessous représente une liste non exhaustive des clients CBI :

**Figure 9 : Clients de CBI**

#### Cadre du projet :

##### Fiche du projet

Dans le cadre de ma dernière année de Master Ingénierie des Réseaux Informatiques et des Systèmes de Télécommunications Intelligents à la Faculté des Sciences et Techniques de Settat, je dois effectuer un stage de fin d’étude d’une durée de 6 mois. Ce stage vise à clôturer mon cursus. Il me permet d'être formé dans une entreprise afin d'acquérir des connaissances sur un secteur d'activité tout en me permettant de mettre en pratique les connaissances théoriques que j'ai acquises lors de mon cursus. Dans ce rapport, je présente mon environnement de travail et la mission principale que j'ai réalisée au sein de l'entreprise CBI présenté dans la partie précédente.

Mon stage se porte sur le DevOps pour augmenter la disponibilité des services offerts. En quelques mots, l'objectif de mon travail est de rendre une application plus résistante aux pannes et aux fortes affluences.

##### Axes du projet

Le projet est divisé en trois axes principaux en se basant sur une méthodologie gestion de projet :

**Gestion e projet**

**Etude comparative et choix des technologies**

**Migration vers le cloud (Azure)**

**Et on-premise**

**Conception &**

**Développement**

**Figure 10 : Axes du projet**

Après avoir défini le mode de gestion du projet, l'avancement du projet commence par une étude comparative des technologies pour sélectionner la technologie la mieux adaptée aux besoins du projet, puis met en œuvre une étude conceptuelle pour s'assurer que l'architecture d'information structurée est utilisée pour présenter le système de manière compréhensible, ce qui permet de démarrer le développement avec une idée claire des composantes du projet. La phase finale du projet comprenait la migration de l'application développée vers le cloud Azure et on-premise.

##### Méthodologie adaptée

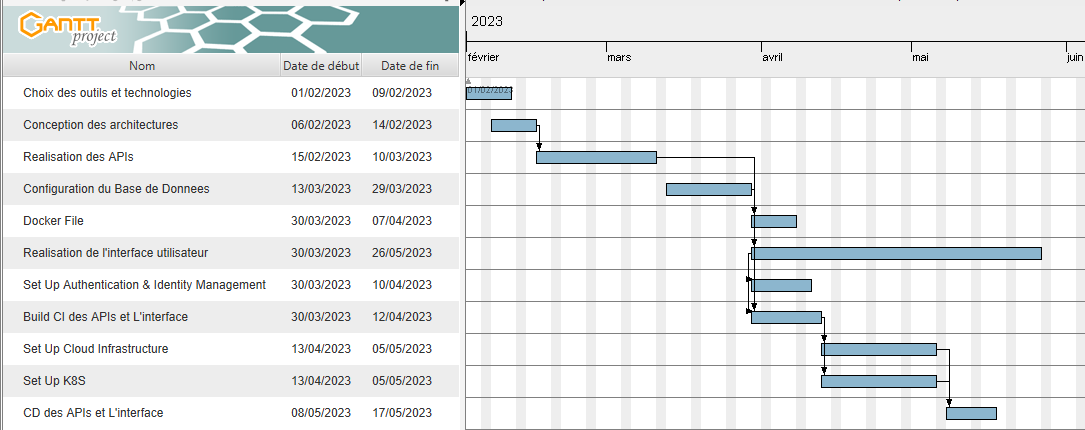
La méthodologie de travail chez CBI se distingue par des réunions et un suivi continue sur l’avancement de n’importe quel projet ; le tout pour assurer une certaine qualité et conformité au besoin et aux attentes. Ainsi, l'ensemble des bonnes pratiques suivantes sera utilisé :

* L’extraction des fonctionnalités à développer après Brainstorming,
* Le découpage du projet en plusieurs sprints,
* Le découpage de chaque sprint en une série de tâches,
* Une réunion quotidienne avec le superviseur pour suivre l'avancement,
* Chaque vendredi, il y a une réunion hebdomadaire où nous essayons de répondre à ces trois questions :
  + Quelles tâches sont effectuées?
  + Quelles sont les obstacles rencontrés?
  + Quelles tâches doivent être accomplies à l'avenir ?

##### Planning du projet

La planification du projet consiste à organiser le déroulement des étapes à réaliser dans le temps afin d’avoir une vision claire du temps et avoir une compréhension de l'ensemble des tâches du projet, de leurs dépendances et de leur organisation chronologique.

Dans cette partie de gestion de projet, on découpe le projet en actions à exécuter pour répondre aux besoins de l’entreprise et on réalise un plan des capacités temps et ressources en donnant une estimation de la durée des tâches.



#### Conclusion

**Figure 11 : Diagramme de Gantt**

La présentation du projet a commencé dans ce chapitre en présentant l'organisme d'accueil, le cadre et la planification de sa mise en œuvre. Le chapitre suivant est réservé à la présentation du problématique et la solution proposée.

# Chapitre 2

## ETUDE DE L’EXISTANT

Ce chapitre comporte une description de la phase spécification et de l’étude de l’existant dans laquelle on présentera une critique et proposera une solution à ces problèmes.

#### Analyse de l’existant

L'équipe de développement utilise actuellement la méthode AGILE SCRUM. Après cela, l'équipe opérationnelle intervient pour fournir une version finale sans avoir contrôlé le code. Il n'y a aucun moyen de contrôler ou de surveiller ce processus, donc l'automatisation de ces tâches et la synchronisation des équipes deviennent nécessaires car le temps gaspillé est important et affecte la productivité.

#### Problématique

La gestion des équipes peut être difficile dans certaines entreprises, en particulier la relation entre l'équipe de développement et l'équipe opérationnelle.

Le main objectif d'une équipe opérationnelle est de garantir la stabilité du système. La meilleure façon d'atteindre cet objectif est de contrôler sévèrement la qualité des modifications apportées aux systèmes qu'ils maintiennent. L'objectif de l'équipe de développement est de réaliser les modifications nécessaires au meilleur coût et le plus rapidement possible. Ensuite, la division des tâches entre les deux équipes a rapidement conduit à des conflits constants en raison de l'incompatibilité des objectifs. Les trois problèmes de la gestion de projet sont le temps, la qualité et le coût.

Un autre problème c’est la non-disponibilité du système. Alors la disponibilité du système devient obligatoire.

#### Solution proposée

Par conséquent, la principale préoccupation est d'avoir un processus pour automatiser l'intégration, les tests et le contrôle de la qualité du code et le processus de livraison continue de l'application. Il faut donc une méthode qui permet d'abord une conception et un déploiement rapides d'applications de haute qualité, puis assure que des modifications plus ou moins importantes soient disponibles en quelques minutes. Pour cela, on va utiliser la notion de DevOps et Kubernetes pour assurer la disponibilité de l’application

Le projet se compose de 3 axes principaux :

-CI qui est une approche de développement logiciel qui vise à intégrer régulièrement et automatiquement les modifications de code dans un référentiel central.

-CD qui est une pratique de développement logiciel qui consiste a livrer automatiquement et de manière fiable les applications aux utilisateurs finaux.

-Monitoring est le processus de surveillance continue des systèmes afin de détecter les problèmes.

#### Description du projet

Le Sujet Modernisation des applications au niveau du cloud consiste à développer une solution qui va permettre de collecter les données importantes à partir des factures en format papier grâce à la technologie de l’OCR (Optical Characters recognition) et les introduire dans un système d’information qui facilitera en suite les différents traitements que ces données peuvent subir. Il s’agit donc d’une application distribuée qui contient plusieurs micro services (Authentification, Upload des factures scannés, service Analytics etc.…).

La méthodologie du développement sera le DevOps qui est une approche de développement de logiciels qui combine les pratiques de développement et d'exploitation pour améliorer la collaboration et la communication entre les équipes d'exploitation et de développement. L'objectif est de permettre une livraison plus rapide et plus fiable des logiciels en automatisant les processus de développement, de test, de déploiement et de surveillance. Durant le développement il y a une utilisation des différents langages de programmation ainsi que technologies (Java, Spring, ReactJS, Python, Docker, Kubernetes etc....) selon le besoin en assurant l’intégration et le déploiement continue CI/CD grâce au concept du Pipeline ainsi que les différents outils de déploiement notamment Docker et Kubernetes.

Ce système va être déployer en deux environnements selon le besoin du client :

-Cloud Azure

-on premise

#### Pourquoi Cloud Azure et On-premise

Dans notre cas, le système va être déployer vers Azure pour les clients qui veulent que leurs données être enregistrées dans le cloud, et on-premise pour les clients qui veulent que leurs données soient hébergées sur leur propre infrastructure informatique et installé sur les différents postes de travail.

#### Conclusion :

Dans ce chapitre, on a tout d’abord exposé l’étude de l’existant avant de définir la problématique, sa solution et le cahier de charge de notre projet.

Dans le chapitre suivant, on va voir l’étude fonctionnelle du projet.

# Chapitre 3

## Analyse et conception du projet

Après avoir achevé les phases de spécifications et d’analyse nous passons à l’étude fonctionnelle du projet.

Mon projet se consiste sur des notions très importantes dans le domaine de l’IT. Plusieurs entreprises se dirigent vers eux. Parmi ces notions on trouve la Conteneurisation. Alors une question se pose, C’est quoi la Conteneurisation ?

#### Conteneurisation

La conteneurisation est une approche moderne pour le déploiement d’applications, qui simplifie la gestion des applications, améliore l’efficacité opérationnelle et favorise la portabilité des applications entre les environnements. Un conteneur est une unité logicielle légère et autonome qui contient tout ce dont une application a besoin pour s’exécuter, y compris le code, les bibliothèques et les fichiers de configuration. La conteneurisation utilise des technologies telles que Docker et Kubernetes.

Les conteneurs permettent de créer des environnements d’exécution préconfigures et reproductibles, ce qui facilite le déploiement rapide et cohérent des applications. Cela s’aligne avec les principes DevOps visant à automatiser les déploiements et à réduire les erreurs humaines.

#### le DevOps :

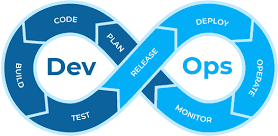
##### Définition

Le mot DevOps est la contraction des mots Operations et Development. Cependant, il représente une variété d'idées et de méthodes bien plus étendue que ces deux mots combinés ou séparément. Le DevOps suit des principes tels que la sécurité, le travail en équipe, l'analyse des données et bien d'autres.

Le DevOps permet aux équipes de travailler ensemble et de collaborer pour créer des produits plus performants et plus fiable. Les équipes peuvent mieux répondre aux besoins des clients, accroître la confiance dans les applications qu'elles développent et atteindre plus rapidement les objectifs de leur entreprise en adoptant une culture DevOps ainsi que les pratiques et les outils DevOps.

##### Cycle de vie des applications

Ce workflow se présente comme une boucle infinie (∞) permettant d’atteindre une fluidité entre le développement et l’exploitation d’un produit**.** Ce workflow compose les étapes suivantes :



**Figure 12 : Cycle de vie DevOps**

##### Plan

La première étape du workflow DevOps consiste à mettre en place un plan de production. C’est sur ce plan que repose l’efficacité du concept DevOps. Ce plan a également pour objectif de diviser le travail en petit morceau, cela afin d’améliorer la vitesse et la qualité de production.

##### Code

La deuxième étape consiste à concevoir des logiciels et à procéder à la création du code logiciel.

##### Build

Cette étape consiste à gérer les versions de logiciel et utiliser des outils automatisés pour aider à compiler et à générer le code pour la future version en production. Il est possible d’utiliser des référentiels de code source qui permettent de générer l’infrastructure nécessaire pour la sortie du produit.

##### Test

Cette partie permet de déceler des erreurs ou des échecs dans le code. Ces tests sont automatisés et sont exécutés dans le référentiel.

Si des problèmes sont détectés, le code posant problème est renvoyé à l’équipe en charge pour qu’elle le corrige et le teste à nouveau. Si aucun problème n’est détecté sur le code, les modifications de code passent à la partie suivante du workflow.

##### Release & deploy

Si les tests relatifs au produit sont positifs, il est alors temps de passer à l’étape suivante : celle de la livraison et du déploiement de l’application. La livraison consiste en une publication du code dans l’environnement de production.

##### Operate & monitor

Le concept DevOps met également l’accent sur le feedback continu des clients.

Il s’agit là d’assurer l’évaluation de chaque version. Grâce à un feedback continu, les équipes peuvent ainsi améliorer leurs processus de production, mais également apporter de nouvelles mises à jour correspondant davantage aux attentes des clients.

DevOps favorise la publication régulière et fréquente des versions logicielles en utilisant les principes Agile. L'automatisation de l'intégration et du déploiement du code est un processus communément appelé intégration continue/déploiement continu (CI/CD). Alors C’est quoi le CI/CD ?

#### CI/CD :

La première notion c’est l’intégration continue (CI). Cette notion désigne l’intégration continue, à savoir un processus automatisé pour les développeurs. Les développeurs doivent modifier régulièrement le code de leur application, la tester, puis la fusionner dans un référentiel partagé dans le cadre de cette intégration continue. Cette solution permet de fluidifier le processus de développement mais aussi d’éviter les régressions.

La deuxième notion est le déploiement continu (CD). Le CD fonctionne de pair avec la CI, en complétant le cycle d'automatisation et en réduisant ou en supprimant l'intervention humaine dans le processus de déploiement. Les outils DevOps automatisés surveillent les mises à jour du code source et les déploient automatiquement dans l'environnement de production une fois qu'elles ont passé la phase de test, ce qui permet d'économiser du temps et d'améliorer la satisfaction des utilisateurs.

L'orchestration joue un rôle essentiel dans la mise en œuvre efficace d'un pipeline CI/CD. Elle permet de gérer et de coordonner les différentes étapes du pipeline CI/CD, en s'assurant que les différentes tâches sont exécutées dans le bon ordre et avec les ressources nécessaires. Cela inclut la gestion des conteneurs, des ressources réseau et de la mise en place des

environnements de test. Deux outils populaires utilisés pour l'orchestration sont Kubernetes et Docker Compose.

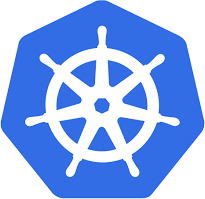
La principale différence entre Kubernetes et Docker Compose réside dans leur capacité à gérer des déploiements à grande échelle et des environnements complexes. Kubernetes offre une orchestration avancée et une gestion automatisée des ressources, ce qui en fait un choix privilégié pour les déploiements en production.

#### Kubernetes

##### Définition

Le nom Kubernetes tire son origine du grec ancien, signifiant *capitaine* ou *pilote* et est la racine de *gouverneur* et [cybernetic](http://www.etymonline.com/index.php?term=cybernetics). *K8s* est l'abréviation dérivée par le remplacement des 8 lettres "ubernete" par "8".

Kubernetes est une plateforme d'orchestration des conteneurs Open Source qui automatise de nombreux processus manuels liés au déploiement, à la gestion et à la mise à l'échelle des applications conteneurisées.



**Figure 13 : Logo K8s**

##### Les avantages de K8s

Les tâches suivantes peuvent être réalisées avec Kubernetes :

* Orchestrer les conteneurs sur divers hôtes
* Améliorer l'utilisation du matériel pour optimiser les ressources nécessaires pour exécuter les applications d'entreprise
* Contrôle et automatisation des déploiements et mises à jour des applications
* Montage et ajout de stockage pour l'exécution des applications stateful
* Mise à l'échelle des applications conteneurisées et de leurs ressources à la volée
* Gestion déclarative des services pour garantir le bon fonctionnement des applications déployées
* Contrôle de l'intégrité et autoréparation des applications grâce au placement, au redémarrage, à la réplication et à la mise à l’échelle automatiques

##### Cluster K8s

Un déploiement fonctionnel de Kubernetes est appelé « cluster ». Les deux principales parties d'un cluster Kubernetes sont le plan de contrôle et les machines de computation ou nœuds.

C'est un ensemble des nœuds qui constituent les principales ressources de calcul gérées par Kubernetes et qui sont protégés du trafic Internet par un pare-feu.

##### Les composants d’un Cluster K8s

##### Node

Une machine Kubernetes qui exécute les actions conteneurisées dans un cluster Kubernetes est un Node Kubernetes. Cela peut être une machine physique ou virtuelle hébergée sur site ou dans le cloud.

##### Namespace

Kubernetes prend en charge plusieurs clusters virtuels sur un seul cluster physique. Namespaces est le nom de ces clusters virtuels.

##### Deployment

Un Déploiement fournit des mises à jour déclaratives pour les [Pods](https://kubernetes.io/fr/docs/concepts/workloads/pods/pod/) et les [ReplicaSets](https://kubernetes.io/fr/docs/concepts/workloads/controllers/replicaset/).

##### Pod

Un pod est une composante essentielle de Kubernetes qui comprend un ensemble de conteneurs qui sont déployés ensemble sur un seul noeud. Tous les conteneurs d'un pod partagent la même adresse IP, IPC, nom d'hôte et autres ressources du système.

Dans notre cas, chaque pod a un ReplicaSet pour garantir la disponibilité de l’application.

##### Service

Un service est un élément qui dissocie les définitions de tâche des pods. Les proxies de service Kubernetes reçoivent automatiquement des demandes de service dans le bon pod, même s'il a été déplacé dans le cluster ou s'il a été remplacé.

##### Ingress

Un Ingress est un objet K8s qui expose les routes HTTP et HTTPS des services au sein du cluster aux services externes.

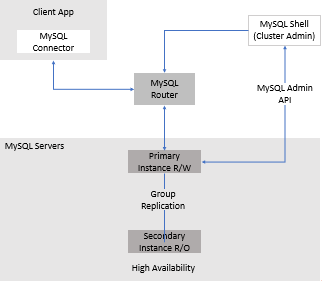
##### InnoDB Cluster

InnoDB est un moteur de stockage clé dans le système de gestion de bases de données MySQL, qui est une base de données open source prenant en charge différents moteurs de stockage. InnoDB est un sous-système de stockage spécifique qui a gagné en popularité ces dernières années en raison de ses performances élevées et de sa fiabilité supérieure par rapport à ses concurrents.

InnoDB Cluster utilise les technologies MySQL suivantes :

* MySQL Shell, qui est un client avancé et un éditeur de code pour MySQL.
* MySQL Server, et la réplication de groupe, qui permet à un ensemble d'instances MySQL d'assurer une haute disponibilité. InnoDB Cluster fournit une alternative programmatique facile à utiliser pour travailler avec la réplication de groupe.
* MySQL Router, un middleware léger qui assure un routage transparent entre l'application et le cluster InnoDB.

Voilà l’architecture d’InnoDB Cluster du projet :



**Figure 14 : Architecture d'InnoDB Cluster**

Alors on peut dire que Kubernetes est une solution open-source qui permet de tirer parti des infrastructures qu'elles soient on-permises ou en Cloud. Alors c’est quoi le cloud computing ?

#### Cloud computing

Le cloud computing est la fourniture de services informatiques via Internet dans le but d’accélérer l’innovation, d’offrir des ressources flexibles et de profiter d’économies d’échelle.

Actuellement, le marché est clairement dominé par 3 acteurs : AWS (Amazon Web Services), Microsoft Azure et Google Cloud Platform. Dans notre projet on va utiliser Microsoft Azure. Chacun de ces fournisseurs de services cloud possède ses propres avantages et caractéristiques uniques, mais je vais citer quelques raisons pour lesquelles j'ai choisi Microsoft Azure pour mon projet :

* Large gamme de services : Azure offre une vaste gamme de services cloud, allant du calcul et du stockage à l'IA et à l'IoT. On peut facilement trouver les services dont on a besoin pour répondre aux exigences spécifiques de notre projet.
* Intégration avec l'écosystème Microsoft : notre projet repose sur des technologies Microsoft existantes, Azure offre une intégration transparente avec ces outils et plateformes. On peut bénéficier d'une continuité dans nos processus de développement et de gestion, notamment avec des produits tels qu’Active Directory,
* Sécurité et conformité : Azure accorde une grande importance à la sécurité des données et propose des fonctionnalités de sécurité avancées pour protéger nos applications et nos informations sensibles.
* Prise en charge multiplateforme : Azure prend en charge divers langages de programmation, frameworks et systèmes d'exploitation. Cela nous offre une flexibilité pour développer notre projet avec les technologies de notre choix.
* Forte présence mondiale : Azure dispose d'un réseau mondial d'emplacement de centres de données, ce qui nous permet de déployer nos applications et services dans différentes régions du monde. Cela peut être bénéfique si notre projet nécessite une présence géographique étendue ou une réduction de la latence pour les utilisateurs dans des régions spécifiques.

Cependant, il est important de noter que le choix du fournisseur de services cloud dépend également de l’entreprise, CBI est partenaire de Microsoft, et de son budget.

Ainsi, Microsoft Azure propose en effet un niveau gratuit, connu sous le nom de "Free Tier", qui permet aux utilisateurs de commencer à utiliser certains services Azure sans frais pendant une période limitée. Cela peut être un avantage significatif, notamment pour les projets de petite taille ou les développeurs souhaitant explorer les fonctionnalités d'Azure sans engagement financier initial.

Il est important de noter que bien que AWS (Amazon Web Services) et Google Cloud Platform (GCP) ne proposent pas de niveaux gratuits aussi généreux que celui d'Azure.

#### Microsoft Azure :

Azure est une plateforme Cloud intelligente qui propose une large gamme de services de plateforme (PaaS) conçus spécifiquement pour les développeurs et les applications.



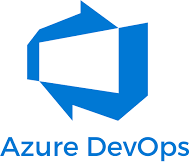
**Figure 15 : Logo Microsoft Azure**

Azure comporte de deux plateformes :

##### Azure DevOps :

La solution d'intégration continue proposée par Microsoft pour remplacer Team Foundation Server est Azure DevOps.

Cet environnement hébergé dans le cloud Azure qui permettra de planifier les développements, versionner le code source avec Git, automatiser la construction et le déploiement des applications, lancer des scénarios de test et tracer les anomalies rencontrées dans ces processus.

Azure DevOps aide à Planifier plus intelligemment, collaborer mieux et livrer plus rapidement avec un ensemble de services de développement modernes.

**Figure 16 : Logo Azure DevOps**

Cette plateforme fournit ces fonctionnalités suivantes :

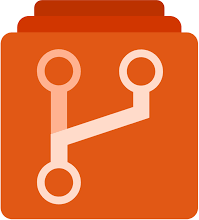
##### Azure Boards

Cette fonctionnalité aide à planifier et effectuer un suivi des taches. Azure Boards prend en charge les approches agiles telles que Scrum et Kanban. Avec des tableaux de bord, il offre une variété de fonctionnalités et d'intégrations pour aider les équipes à collaborer et à rester organisées.



**Figure 17 : Logo Azure Boards**

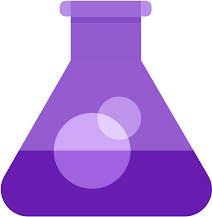
##### Azure Repos

Azure Repos est utilisé pour stocker le code. C’est un ensemble d’outils de gestion de versions utilise pour gérer le code source.

**Figure 18 : Logo Azure Repos**

##### Azure Test Plans

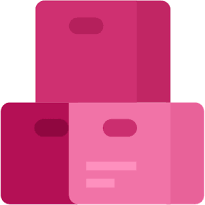
Les outils de test des applications, y compris les tests continus, sont fournis par Azure Test Plans.

Ils permettent également la création de plans de test dans lesquels les étapes, les actions et les résultats attendus peuvent être enregistrés.

**Figure 19 : Logo Test Plans**

##### Azure Artifact

Azure Artifact aide à partager le code sans effort en stockant ensemble des packages Maven,Npm,NuGet et Python.



**Figure 20 : Logo Azure Artifact**

##### Azure Pipelines

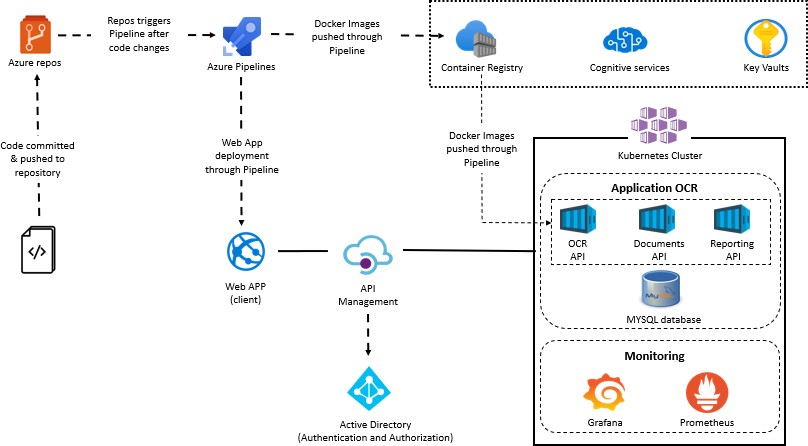
Les projets de code sont créés et testés automatiquement par la solution Azure Pipelines. Il prend en charge tous les principaux langages et types de projets et combine l'intégration, la livraison et les tests continus pour générer, tester et livrer du code vers Azure Portal.

##### Azure Portal

**Figure 21 : Logo Azure Pipelines**

Azure Portal est une interface web unifiée qui fournit une alternative conviviale aux outils en ligne de commande. C’est une plateforme qui nous permet d’accéder à toutes nos applications. Cela permet de les gérer en un seul endroit. Nous pouvons créer, gérer et surveiller toutes les applications, des simples applications Web aux applications complexes en cloud.

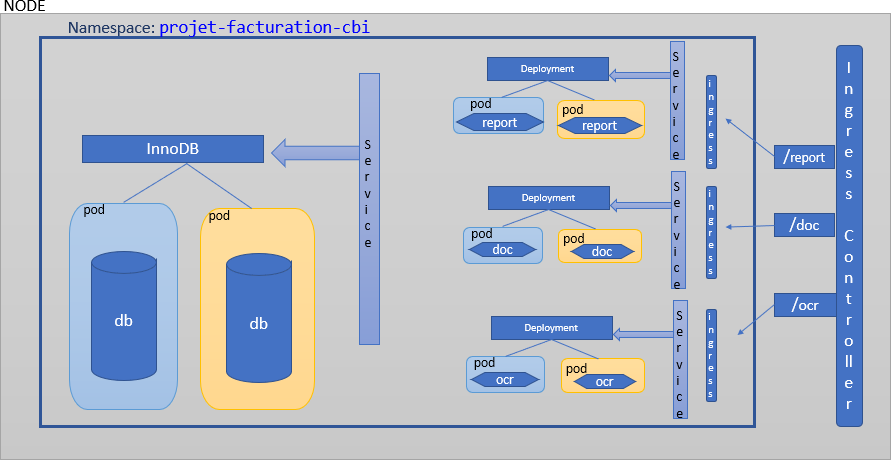
##### Architecture Azure :



**Figure 22 : Architecture Azure**

On va adopter cette architecture pour réaliser le projet sur Azure.

##### Architecture Kubernetes :



**Figure 23 : Architecture K8s Azure**

Pour le déploiement en utilisant K8s, on va créer un namespace : Projet-facturation-cbi . Ce namespace contient nos déploiements, les pods pour chaque api et ses replicas, InnoDB cluster qui contient un pod pour la base de données et son réplica, un service pour chaque déploiement et un Ingress Controller pour que notre application être accessible de l’extérieur.

Les ReplicaSet sont utilisés pour garantir la haute disponibilité et la résilience. Il présente le nombre de copies du pod principale.

##### Avantages et Inconvénients

|  |  |
| --- | --- |
| Avantages | Inconvénients |
| Optimisation des couts Flexibilité et scalabilité Conformité  Sauvegarde, récupération et basculement  Gestion et surveillance Simplifiées | Les pannes Sécurité  La confidentialité des données |

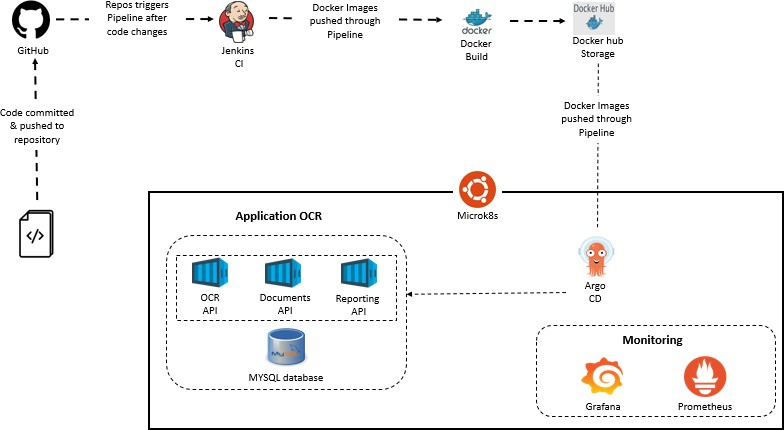
**Tableau 2 : Les avantages et les inconvénients du Cloud Azure**

#### On-premises :

##### Definition

On-Premises signifie sur site. Cette définition d'On-Premises fait référence à l'utilisation du serveur et de l'environnement informatique de l'entreprise. Le client, ou licencié, achète ou loue un logiciel basé sur serveur qui sera installé sur son propre serveur ou sur un serveur loué dans ce modèle d'utilisation.

##### Architecture On-Premises

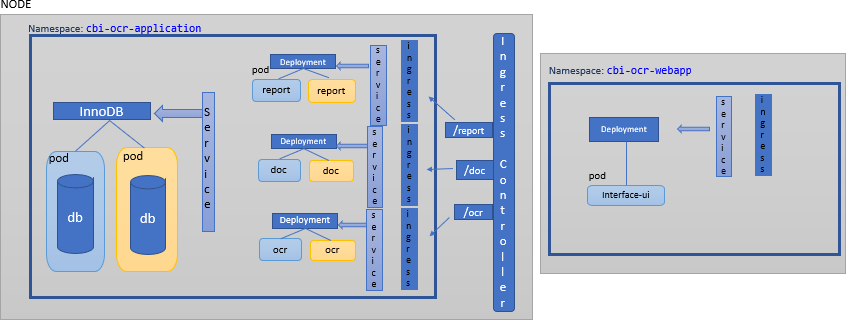


**Figure 24 : Architecture On-premise**

Cette architecture est adoptée pour réaliser le projet sur site.

##### Architecture microK8s

MicroK8s est utilisé pour les environnements de développement, les tests locaux et l’exécution de petits clusters Kubernetes sur une seule machine.



**Figure 25 : Architecture K8s On-premise**

Dans ce cas de déploiement su site, on a besoin de 2 namespaces :

* Cbi-ocr-application : qui contient notre InnoDB cluster, les déploiements des APIs et le service Ingress.
* Cbi-ocr-webapp : qui contient le déploiement de notre interface utilisateur et le service Ingress.

##### Avantages et Inconvénients :

|  |  |
| --- | --- |
| Avantages | Inconvénients |
| Sécurité renforcée  Contrôle sur les serveurs Customisation | Besoin d’un support IT supplémentaire  Accroitre l’investissement dans le capital Limiter la capacite de mise à l’échelle |

**Tableau 3 : Les avantages et les inconvénients d'on-premise**

#### Conclusion :

Dans ce chapitre j’ai présenté une vue conceptuelle du projet. Le chapitre qui suit détaille les outils utilisés pour la réalisation du projet.

# Chapitre 4

## Réalisation et outils

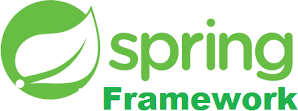
Dans ce dernier chapitre comporte en premier lieu les outils, les technologies utilisés pour répondre au besoin du projet, en deuxième lieu la modernisation du système dans le cloud azure et on-premise.

#### l’application :

##### Languages et Framework :

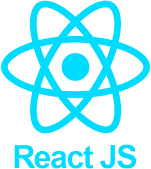
##### Spring java

Le framework Spring Java est le plus populaire pour faciliter le développement et le test d'applications Java. C'est un Framework léger et open source qui permet aux développeurs Java et Java EE de créer des applications d’entreprise.



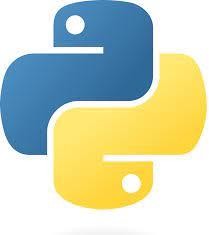
##### ReactJS

**Figure 26 : Logo Spring Java**

ReactJS est une bibliothèque JavaScript. Il permet de construire des interfaces utilisateur composables. Elle favorise la création de composants d’interface utilisateur réutilisables qui présentent des données qui évoluent dans le temps.

##### Python

**Figure 27 : Logo ReactJS**

Python est un langage de programmation puissant et facilement accessible à l'apprentissage. Il offre des structures de données de haut niveau et permet une technique de programmation orientée objet simple mais efficace..

**Figure 28 : Logo Python**

##### MySQL

MySQL est un système de gestion de bases de données relationnelles open source largement utilisé. Il est réputé pour sa simplicité, sa performance, sa fiabilité et sa compatibilité avec diverses plateformes. C'est un service de base de données entièrement géré qui utilise la base de données open source la plus populaire au monde pour déployer des applications natives du cloud.

**Figure 29 : Logo MySQL**

##### les outils utilisés:

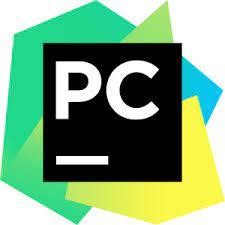
* + 1. **Visual Studio**

L'IDE (Integrated Development Environment) Visual Studio est une plateforme de développement complète qui offre aux utilisateurs un environnement créatif pour modifier, déboguer, générer et publier des applications. En plus des fonctionnalités d'édition et de débogage standard que l'on trouve dans la plupart des IDE, Visual Studio propose des compilateurs, des outils d'achèvement de code, des concepteurs graphiques et de nombreuses autres fonctionnalités pour améliorer le processus de développement de logiciels.

##### PyCharm

**Figure 30 : Logo Visual Studio**

PyCharm est un environnement de développement intégré (IDE) couramment utilisé pour la programmation en Python. Pour améliorer le processus de développement et faciliter la création d'applications Python, cet outil offre une multitude de fonctionnalités.



**Figure 31 : Logo PyCharm**

#### Cloud Azure :

Concernant le déploiement vers le Cloud Azure, On va travailler avec les services d’Azure

##### Azure Devops :

* + 1. **Azure Board :**

Azure Boards contient toutes les taches pour réaliser le projet. Il contient 32 taches débutantes par la conception de l’architecture du projet et la réalisation des APIs.

Azure Boards nous aide à planifier et effectuer un suivi des taches pour réaliser un produit de manière plus efficace.

|  |
| --- |
| Les taches de notre projet |
| * Design a globale architecture for our components on Azure * create API CRUD named Doc\_API using Spring Java * configure Database * Test API CRUD Doc\_API * Create DockerFile for Doc\_API * Test API CRUD Docker Image * Create API\_Reporting using Spring Java * Test API-Reporting * Create DockerFile for API-Reporting * Test API-Reporting Docker Image * Create API\_OCR using Machine learning * test API OCR * create dockerfile for API OCR * test API OCR docker image * Create Basic features User Interface using ReactJS * Link User Interface to API OCR * Link User Interface to API DOCS * Link user Interface to API Repporting * Test user interface * Configure Azure Active Directory * Create app registration for OCR APPLICATION * add authentication via ms azure ad to OCR APPLICATION |

* Add chatBot to the application
* Use Azure KeyVaults to stock the secrets
* Create ACR to stock the Docker Images
* Create Pipeline for each of the APIs
* AKS Architecture
* Configure AKS in GitOps Repo
* Create Kubernetes Deployment manifests
* Create 2 Pipelines: create AKS and delete AKS
* Setup Api Management and link it with Azure AD
* SetUp Monitoring

**Tableau 4 : Les taches de notre projet**

Azure Boards s’intègre de manière transparente avec d’autres outils Azure, tels qu’Azure Repos.

##### Azure repos :

Dans Azure Repos, On stocke le code source. Il y a 5 Repos qu’on a travaillé avec :

Document\_API : c’est une api CRUD, qui créer, mis à jour , lire et supprimer les factures. Cette api est réalisée en utilisant Spring Java.

Reporting\_API : cette api retourne le nombre de factures par date et génère un PDF qui contient les informations des factures. Elle est réalisée en utilisant Spring Java

OCR\_API : c’est une api d’océrisation, elle convertit la facture du format PDF en format Json pour la stocker dans la base de données. Elle est réalisée en utilisant le Machine Learning spécialement Tesseract et opencv.

User\_Interface : c’est l’interface utilisateur réalisée en utilisant ReactJS.

GitOps : c’est un repo qui contient l’architecture du K8s pour la rendre automatisée.

Chaque Repo a un dockerfile qui est un fichier texte utilise pour créer une image Docker. Il contient une série d’instructions qui spécifient les étapes nécessaires pour construire une image Docker complète et prête à être exécutée en tant que conteneur.

On prend le dockerfile du Document\_API :



**Figure 32 : DockerFile**

* From openjdk :17

Cette instruction indique que l’image de base sera basée sur l’image openjdk :17 qui est une image officielle OpenJDK avec la version 17 de java.

* ADD . Document-API/

Cette instruction ajoute les fichiers de l’application dans le répertoire courant ou se trouve DockerFile au répertoire /Document-API/ dans l’image.

* WORKDIR /Document-API

Cette instruction définit le répertoire de travail a /Document-API. Toutes les instructions suivantes seront exécutées dans ce répertoire.

* EXPOSE 8080

Cette instruction expose le port 8080 dans le conteneur Docker.

* ENTRYPOINT [“java”, “-jar”, “./target/gestion\_factures-0.0.1-SNAPSHOT.jar”]

Cette instruction définit la commande d’entrée par défaut qui sera exécutée lorsque le conteneur démarre.

Ces instructions Docker permettent de créer une image Docker à partir du DockerFile pour exécuter l’API Java basée sur Spring Boot dans un conteneur Docker.

Azure repos s’intègre avec Azure Pipelines permettant une intégration et une automatisation continues.

##### Azure pipelines :

Azure pipelines est un service de CI/CD qui permet l’automatisation du processus de génération, de test et de déploiement d’application.

Dans notre cas, on a 6 pipelines :

Interface\_CI : c’est le pipeline lié à User\_interface. Cette pipeline build le code et le met dans l’artifact, c.-à-d. CI. Et puis on utilise CD pour mettre l’artifact dans le web App. Les artifacts sont les résultats de la génération d’un projet logiciel, tels que les fichiers binaires, les packages et les fichiers de configuration.

Reporting\_API et Document\_API : ces 2 pipelines build le code et push docker image vers ACR.

OCR\_APP\_API : ce pipeline met docker image dans ACR

Create\_AKS et Delete\_AKS: ces pipelines sont utilisés pour créer le cluster Kubernetes vers 9h matin et le supprimer vers 18h pour optimiser le cout de consommation.

Pour lier la plateforme D’azure DevOps avec Azure Portal pour mettre les Docker images dans ACR, on a besoin d’un Service Connection.

##### Service connection :

Un service connection est une entité dans Azure DevOps qui permet une connexion sécurisée à des services externes tels que des référentiels de code source, des fournisseurs de cloud, des bases de données et des outils de test. Elle fournit les informations d’authentification et d’accès nécessaires pour que les pipelines et les taches puissent interagir avec ces services.

##### Azure portal :

* + 1. **ACR**

Azure Container Registery est un service de registre de conteneurs géré et privé fourni par Microsoft Azure. Il permet de stocker et gérer les images de conteneurs Docker.

ACR peut être un choix préférable car on utilise déjà Azure.

Dans notre cas, on créer un container registery qui contient 3 images de conteneurs Docker :

* Doc\_api\_image : est l’image de l’API\_Doc
* Report\_api\_image : est l’image de l’API\_reporting
* Ocr\_api\_image : est l’image de l’API\_OCR

Il s’intègre de manière transparente avec d’autres services Azure, tels que AKS et Azure App Service.

##### AKS

AKS est l’abréviation d’Azure Kubernetes Service, c’est un service de conteneurisation managé proposé par Microsoft Azure. Il facilite le déploiement, la gestion et la mise en œuvre des applications conteneurisées à l’aide de l’orchestrateur de conteneurs Kubernetes.

Il simplifie la création et l’exploitation d’un environnement Kubernetes en automatisant les taches de gestion complexes.

Dans notre cas, on a créé d’abord 4 namespaces :

* Projet-facturation-cbi :

Ce namespace contient les pods et ses réplicas, les déploiements et les services.

Il existe un pod pour chaque api et son réplica, la base de données, son réplica et son routeur pour la démarrer.

Concernant les déploiements, on a un déploiement pour chaque api pour fournir les mises à jour au pods.

Chaque Api a un service et une adresse IP pour la communication avec les autres Apis. Chaque Api a un service ingress pour la rendre accessible de l’extérieur.

* Mysql-operator

Ce namespace contient les opérateurs pour la création de la base de données. Il fournit un moyen automatise de déployer, gérer et mettre à l’échelle des clusters MySQL dans un environnement Kubernetes. Il est conçu pour simplifier l’exploitation et l’administration des bases de données MySQL en utilisant les concepts et les fonctionnalités de kubernetes.

* Ingress-ocr-controller

C’est un namespace qui contient Ingress Controller qui est un composant Kubernetes essentiel qui facilite la gestion du trafic entrant vers les services déployés dans un cluster Kubernetes, en offrant des fonctionnalités avancées de routage et d’équilibrage de charge.

* Monitoring

Ce namespace contient le monitoring. C’est un processus qui consiste à collecter et à analyser des données afin de détecter, mesurer et de signaler les performances, les anomalies et les erreurs. Son Objectif est de garantir la disponibilité, la performance et la fiabilité du système et du service.

Ce namespace contient les outils de surveillance Prometheus ,qui est un système base sur la collecte de métriques, et Grafana qui est une plateforme de visualisation de données et de tableaux de bord.

Ce namespace des pods et des déploiements pour Prometheus et Grafana, et des services ingress pour le rendre accessible de l’extérieur.

##### Azure AD

Azure Active Directory est un service de gestion des identités base sur le cloud. Il permet la création et le gestion des utilisateurs, groupes et rôles. On peut sécuriser l’accès à nos ressources Azure, y compris AKS, en utilisant l’authentification et l’autorisation basées sur les identités Azure AD.

Pour créer et configurer des applications dans Azure AD , on a utilisé Azure App Registrations qui est un service spécifique d’Azure AD. Azure App Registrations est utilisé pour l’authentification dans l’application qui est gérée par Microsoft.

Il est utilisé également pour l’authorization. On a deux rôles :

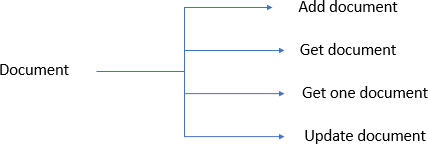
* Admin : qui peut consulter le tableau de bord dans l’application
* User : qui peut upload les factures et voir l’accueil

Il est nécessaire d’avoir à la fois le rôle d’administrateur et le rôle d’utilisateur pour accéder à l’ensemble des fonctionnalités de l’application.

##### API management

Azure API Management est un service qui permet de gérer, sécuriser, publier et analyser les API. Il agit comme une passerelle entre les utilisateurs qui consomment les API et les services qui les fournissent.

On a créé 3 APIs. Document , Ocr et report. Chaque API a des opérateurs.



**Figure 33 : Operations du document**



**Figure 34 : Operations d"OCR**



**Figure 35 : Operations de report**

L’intégration entre Azure AD et Azure API Management permet de sécuriser et de contrôler l’accès aux API exposées via Azure API Management en utilisant les fonctionnalités d’authentification et d’autorisation d’Azure AD.

##### Azure App Service

Azure App Service est un service de plateforme en tant que service (PaaS) qui permet de créer, déployer et héberger des applications web. Elle facilite le déploiement des applications web et elle garantit une haute disponibilité du système.

Dans notre cas, on a créé deux App Services :

* Interface-u : c’est un web app qui contient notre application d’océrisation.
* Interface-chat : c’est le web app qui contient le Chatbot

##### Chatbot

Le chatbot est un programme informatique conçu pour simuler une conversation humaine et interagir avec les utilisateurs via une interface de chat.

Dans notre projet, le chatbot est réalisé pour aider les utilisateurs à circuler dans l’application.

Pour créer ce chatbot, On a utilisé Azure Bot Service, qui permet de créer, déployer et gérer des chatbots. Parmi les fonctionnalités d’Azure Bot Service, on trouve :

* Développement base sur des modèles : il propose des modèles de démarrage pour faciliter le développement des chatbots.
* Intégration avec les services cognitifs : comme le traitement du langage naturel (NLP) avec Azure Language Understanding (LUIS) qui sont bases sur Machine Learning.
* Déploiement et hébergement faciles : il est possible de déployer le chatbot vers Azure App Service à partir de Visual Studio.

##### Key Vaults

Azure key Vault est un service de gestion des secrets offert par Microsoft Azure. Il permet de stocker, gérer et protéger les mots de passe et les clés de chiffrement.

##### Monitoring (Grafana & Prometheus)

Prometheus et Grafana sont deux outils populaires utilisés pour la surveillance et la visualisation des systèmes et des métriques. Ils offrent des fonctionnalités avancées et complémentaires, ce qui en fait un choix courant pour la surveillance et la visualisation des applications.

Voici quelques raisons qui peuvent expliquer le choix de Prometheus et Grafana :

* Architecture orientée service : Prometheus est conçu pour fonctionner dans une architecture orientée service, où il collecte et stocke les métriques des différentes parties d'un système distribué. Il utilise un modèle de collecte de données basé sur les expositions HTTP, ce qui le rend compatible avec une grande variété de systèmes et de langages de programmation.
* Collecte flexible des métriques : Prometheus prend en charge une collecte flexible des métriques. Il peut collecter des métriques sur les performances, l'utilisation des ressources et les erreurs.
* Stockage et requêtage des métriques : Prometheus dispose d'un stockage intégré des métriques à long terme, ce qui permet de stocker les données historiques et de les interroger facilement. Il utilise une base de données de séries temporelles pour stocker les données, ce qui permet des requêtes rapides et efficaces sur les métriques.
* Visualisation des métriques : Grafana est un outil de visualisation de données puissant et flexible. Il permet de créer des tableaux de bord personnalisés pour visualiser les métriques collectées par Prometheus. Grafana offre une large gamme de types de graphiques, de panneaux et de fonctionnalités d'exploration des données pour créer des visualisations attrayantes et informatives.

#### On-premise :

##### Ansible

Ansible est un moteur d’automatisation Open Source qui automatise la gestion des configurations et l’installation des logiciels en utilisant des playbooks qui sont des modèles de taches d’automatisation.

Dans notre cas, on a utilisé ansible pour l’installation de Jenkins, Docker et Kubernetes.

Ansible est un choix approprié pour l'installation de Jenkins, Argo CD, Docker et MicroK8s en raison de son automatisation, de sa reproductibilité, de sa compatibilité multiplateforme, de sa communauté active et de son extensibilité. Il facilite le déploiement et la configuration cohérente de ces logiciels, tout en offrant une flexibilité pour s'adapter à des besoins spécifiques.

Ainsi si on souhaite installer Jenkins, Argo CD, Docker et MicroK8s sur d'autres machines, Ansible nous facilite la tâche.



**Figure 36 : Logo d'Ansible**

##### Github

Github est une plateforme de développement collaboratif basée sur Git. Elle offre un service d’hébergement de code source, de gestion de versions et de collaboration pour les projets logiciels.

le choix de GitHub pour la gestion d'un projet de développement logiciel peut être motivé par ses fonctionnalités avancées de contrôle de version et sa facilité de collaboration.

On a créé une organisation qui contient notre repositories, les mêmes qui sont dans Azure Repos. Ces repos sont liés avec Jenkins.



##### Jenkins

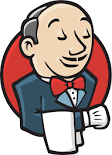
**Figure 37 : Logo de Github**

Lorsqu'il s'agit d'outils de CI/CD, il existe plusieurs options populaires, notamment Jenkins, GitLab CI/CD, Travis CI, CircleCI et GitHub Actions. Chaque outil a ses propres caractéristiques et avantages. Voici une comparaison générale des outils de CI :

* Jenkins : Jenkins est un outil CI/CD très flexible et personnalisable. Il est connu pour sa vaste gamme de plugins et son support pour de nombreuses intégrations et configurations personnalisées. Jenkins est largement adopté dans les grandes organisations et les projets complexes en raison de sa polyvalence et de sa capacité à s'adapter à des cas d'utilisation variés.
* GitLab CI/CD : GitLab CI/CD est intégré directement à la plateforme GitLab, offrant une expérience de bout en bout pour le contrôle de version, l'intégration continue et le déploiement continu. Cependant, il n’est pas gratuit.
* Travis CI : Travis CI est un outil de CI/CD cloud qui offre une configuration facile et une intégration transparente avec GitHub. Il est simple à configurer et convient particulièrement bien aux projets open source et aux projets de petite à moyenne envergure.
* CircleCI : CircleCI est un outil de CI/CD cloud qui propose une configuration simple et rapide. Il prend en charge plusieurs langages de programmation et offre une intégration transparente avec GitHub.
* GitHub Actions : GitHub Actions est intégré directement à la plateforme GitHub et offre des fonctionnalités de CI/CD. Il permet de créer des workflows personnalisés pour l'intégration continue, le déploiement continu et d'autres tâches automatisées.

Jenkins est un outil de CI/CD open source et gratuit, ce qui en fait un choix attrayant du point de vue financier. L'utilisation de Jenkins ne nécessite pas de frais de licence, et on peut l'installer et l'utiliser sur nos propres serveurs ou machines. Cela peut être avantageux pour les organisations ayant des contraintes budgétaires ou qui préfèrent une solution gratuite.

On a utilisé Jenkins pour les Pipeline CI. Il s’intègre aux systèmes de gestion de version comme Git et il est configure pour déclencher automatiquement des builds des images des conteneurs Docker vers Docker Hub.



On a créé 4 pipelines :

**Figure 38 : Logo de Jenkins**

* Build\_OCR\_API: c’est une pipeline pour l’API d’ocerisation
* Build\_user\_interface: cette pipeline est pour l’interface utilisateur
* Doc\_API: cette pipeline est pour l’API CRUD
* Reporting\_API: cette pipeline est pour l’API Reporting Ces pipelines push les images vers Docker Hub.

##### Docker Hub

Docker Hub est un service de registre de conteneurs cloud public qui joue un rôle essentiel dans l’écosystème Docker en offrant un emplacement centralise pour stocker, partager et distribuer des images de conteneurs Docker.

##### MicroK8s

MicroK8s est une distribution légère et simplifiée de K8s. Il facilite l’installation, le déploiement et la gestion d’un cluster kubernetes sur des environnements de petite taille.

Le choix de travailler avec MicroK8s peut être justifié en raison des contraintes spécifiques liées à notre environnement . Voici pourquoi MicroK8s peut être une solution appropriée dans notre situation :

* Limitations de ressources : On dispose d'une machine virtuelle avec des ressources limitées, l'installation et la configuration complète de Kubernetes peuvent être gourmandes en ressources et ne pas convenir à notre environnement. MicroK8s est une distribution légère de Kubernetes conçue pour les environnements à ressources limitées, ce qui le rend adapté aux machines virtuelles avec des ressources restreintes.
* Machine virtuelle unique : La configuration de Kubernetes nécessite généralement plusieurs machines virtuelles pour créer un cluster Kubernetes. Cependant, MicroK8s permet d'exécuter un cluster Kubernetes sur une seule machine virtuelle, ce qui facilite grandement l'installation et la configuration de Kubernetes dans notre environnement.

MicroK8s intègre tous les composants essentiels de K8s et fournit une interface de ligne de commande pour la gestion du cluster.

Il peut servir de plateforme de déploiement pour exécuter un cluster K8s local, tandis qu’Argo CD peut être utilise pour automatiser et simplifier le déploiement du système sur ce cluster.

##### ArgoCD

Le choix entre Argo CD et Jenkins pour la gestion du déploiement continu (CD) dépend des besoins spécifiques de notre projet et de notre infrastructure. Voici quelques raisons pour lesquelles Argo CD peut être préféré par rapport à Jenkins dans le contexte du déploiement continu :

* Orientation spécifique à Kubernetes : Argo CD est conçu spécifiquement pour la gestion des déploiements dans des environnements Kubernetes. Il est étroitement intégré à Kubernetes et permet une gestion et un déploiement simplifiés des ressources Kubernetes, tels que les déploiements et les services.
* Interface utilisateur conviviale : Argo CD dispose d'une interface utilisateur conviviale qui facilite la visualisation, la surveillance et la gestion des déploiements. L'interface intuitive permet aux équipes de développement et d'exploitation d'avoir une vue d'ensemble claire des déploiements en cours et de suivre les changements apportés aux ressources.

C’est un outil d’automatisation des déploiements et de gestion des flux de travail dans le cluster K8s. Il utilise des fichiers YAML pour définir les applications à déployer. Il servit une plateforme pour la visualisation du déploiement du système.



**Figure 39 : Logo d'ArgoCD**

#### Application dans Azure

L’URL de l’application est : <https://interface-u.azurewebsites.net/>. Ce URL est généré par Azure Web App.

Pour sécuriser notre environnement, on a utilisé une connexion sécurisée en activant HTTPS pour chiffrer les communications entre l'application et les utilisateurs.

La gestion des accès et des identités est une autre priorité. On a utilisé Azure Active Directory pour gérer l'authentification et l'autorisation des utilisateurs. Ainsi on a appliqué des principes de moindre privilège en accordant des autorisations appropriées aux utilisateurs et aux ressources.

On a utilisé également API Management pour sécuriser notre application en contrôlant l'accès aux API, en gérant les autorisations, en validant les données d'entrée et en fournissant des fonctionnalités de journalisation avancées.

On a configuré des points de terminaison privés pour nos ressources, tels que des bases de données et nos APIs, afin de limiter l'accès uniquement aux composants nécessaires.

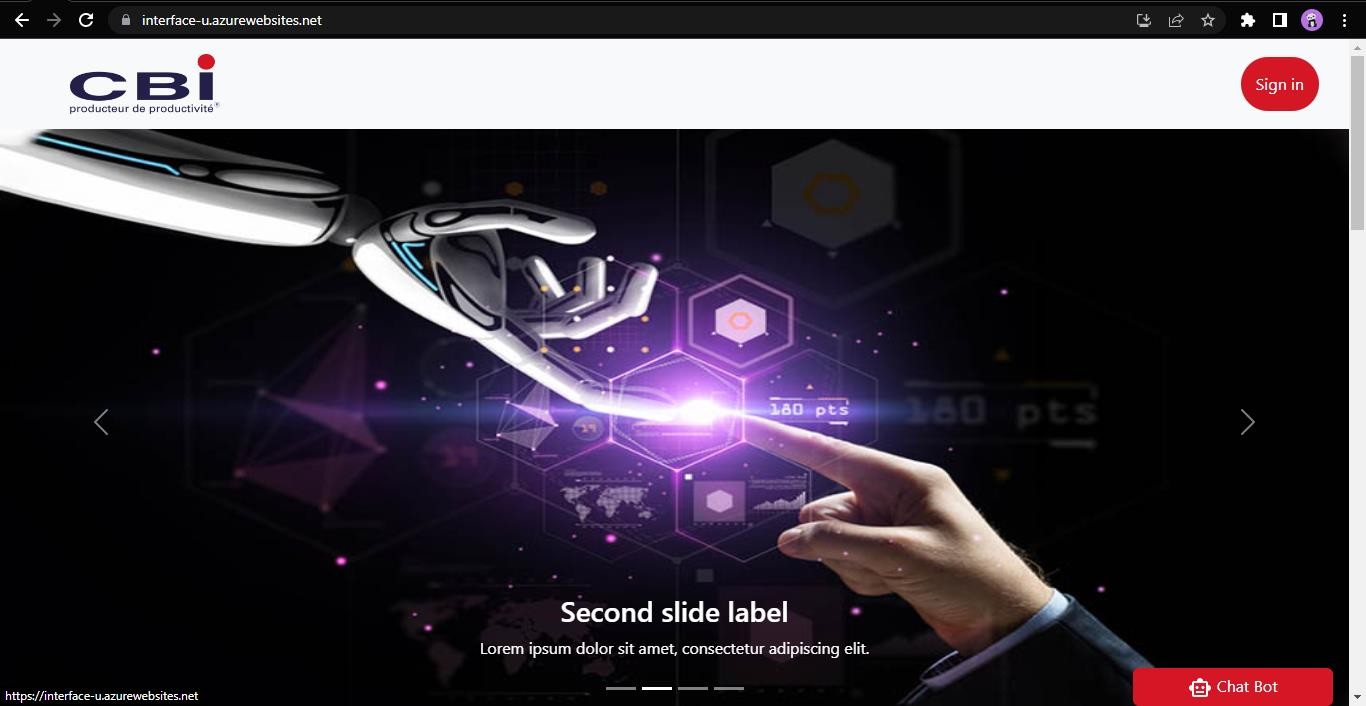
Ainsi, on a utilisé Azure Key Vault pour stocker et gérer en toute sécurité les secrets utilisés par notre application.

l'API Microsoft Graph offre des fonctionnalités de sécurité avancées pour protéger les données et les interactions avec les services Microsoft. En utilisant les mécanismes d'authentification, d'autorisation et de protection des données fournis par l'API, on peut développer des applications sécurisées qui interagissent de manière fiable et protégée avec les services Microsoft.

Il est important de combiner ces mesures de sécurité pour assurer une protection globale de notre application.

##### Interface login :

La première interface de l’application est l’interface login.

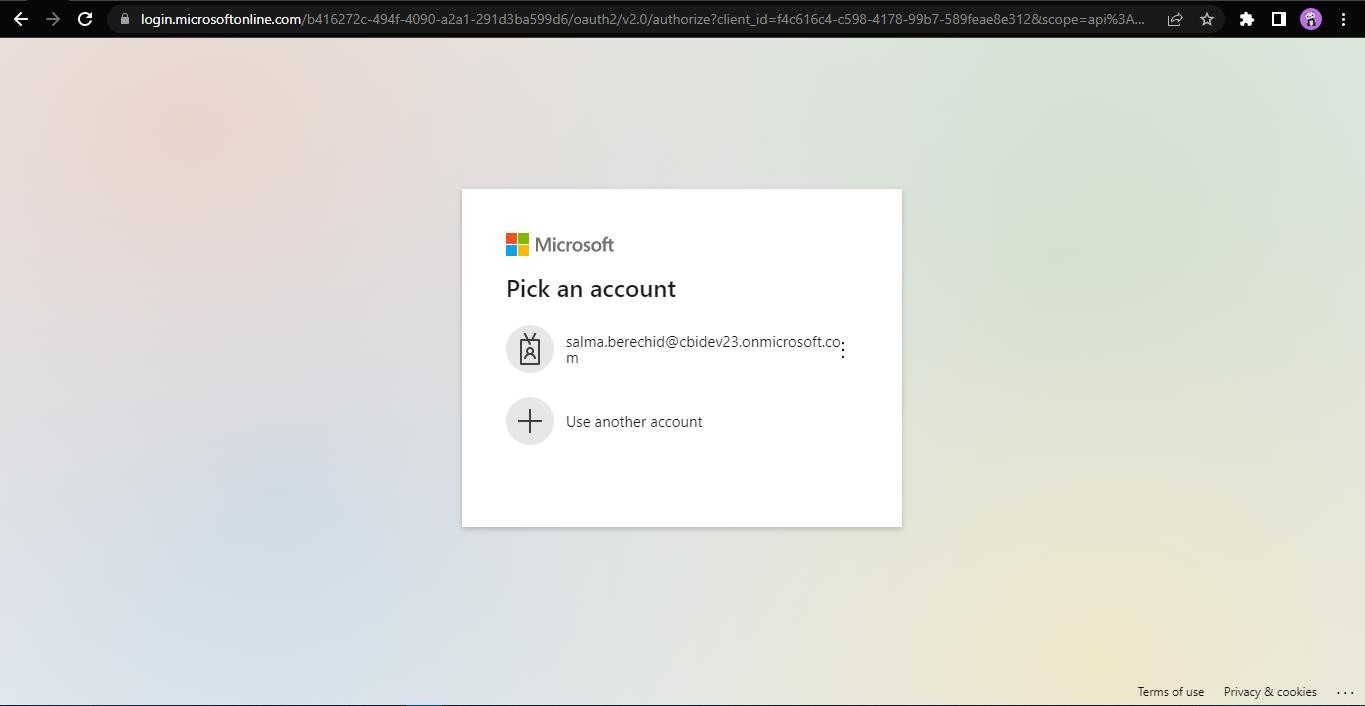


**Figure 40 : Interface Login**

En cliquant sur ‘sign in’, une interface d’authentification s’affiche.

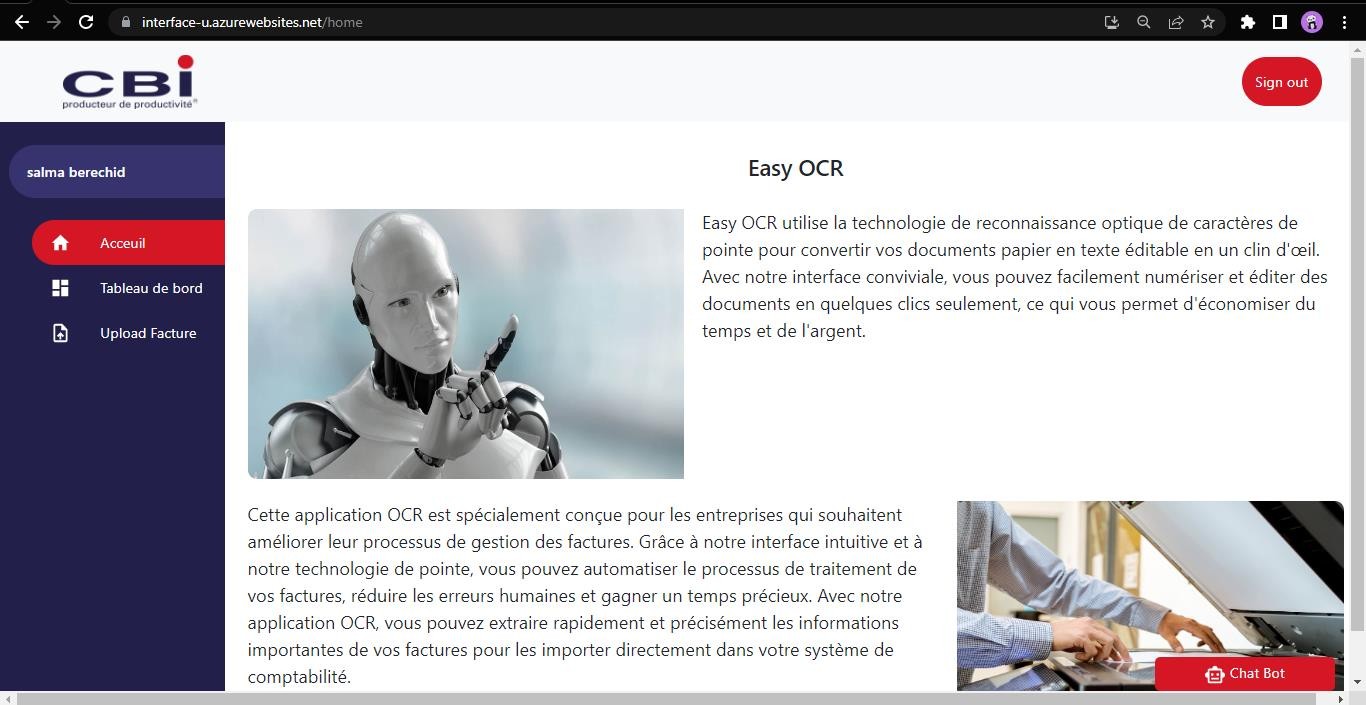
##### Interface d’authentification :

Cette interface redirige vers la page de Microsoft pour authentifier. On a utilisé L’API Microsoft Graph qui consiste à obtenir un jeton d’accès pour authentifier l’utilisateur et lui permettre d’accéder et d’utiliser les fonctionnalités de l’application.



**Figure 41 : Interface d'authentification de Microsoft**

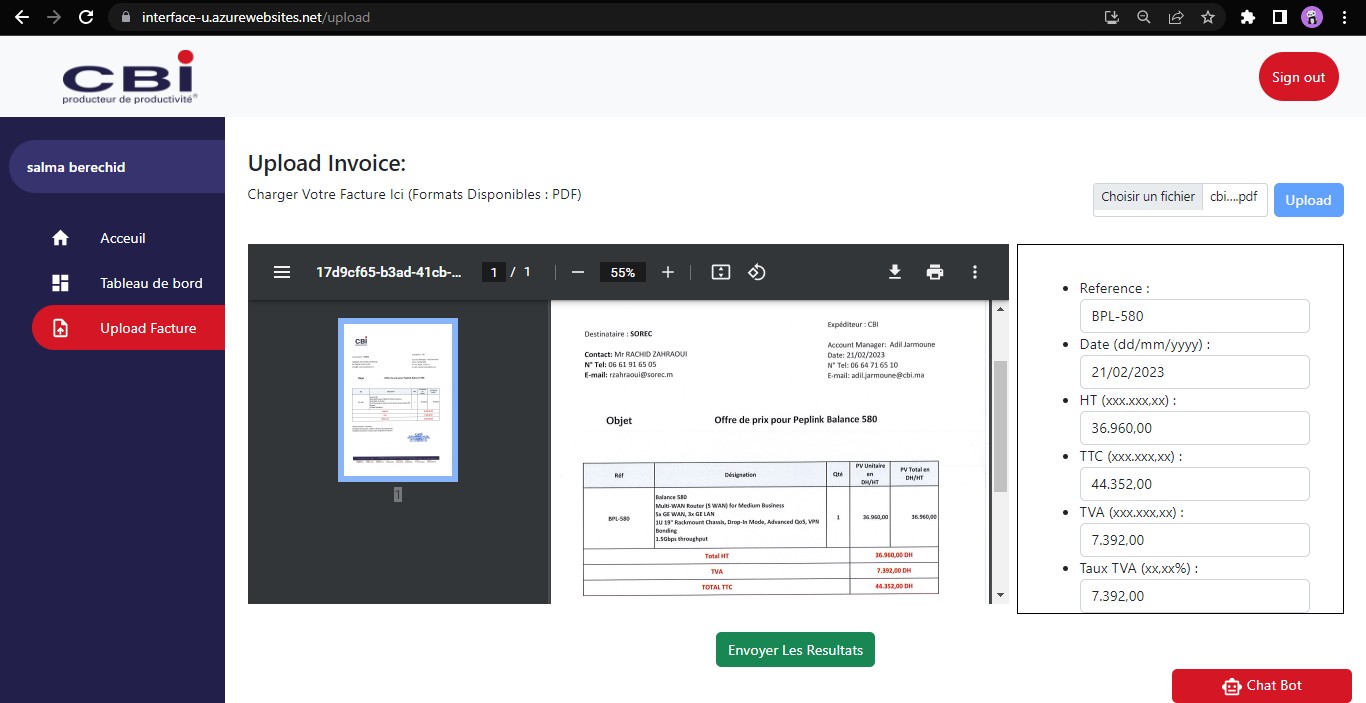
##### Interface D’accueil :

La première page après l’authentification est la page d’accueil. Elle contient une explication concernant l’OCR et le but de l’application.

**Figure 42 : Interface d'accueil**

##### Interface Upload Facture :

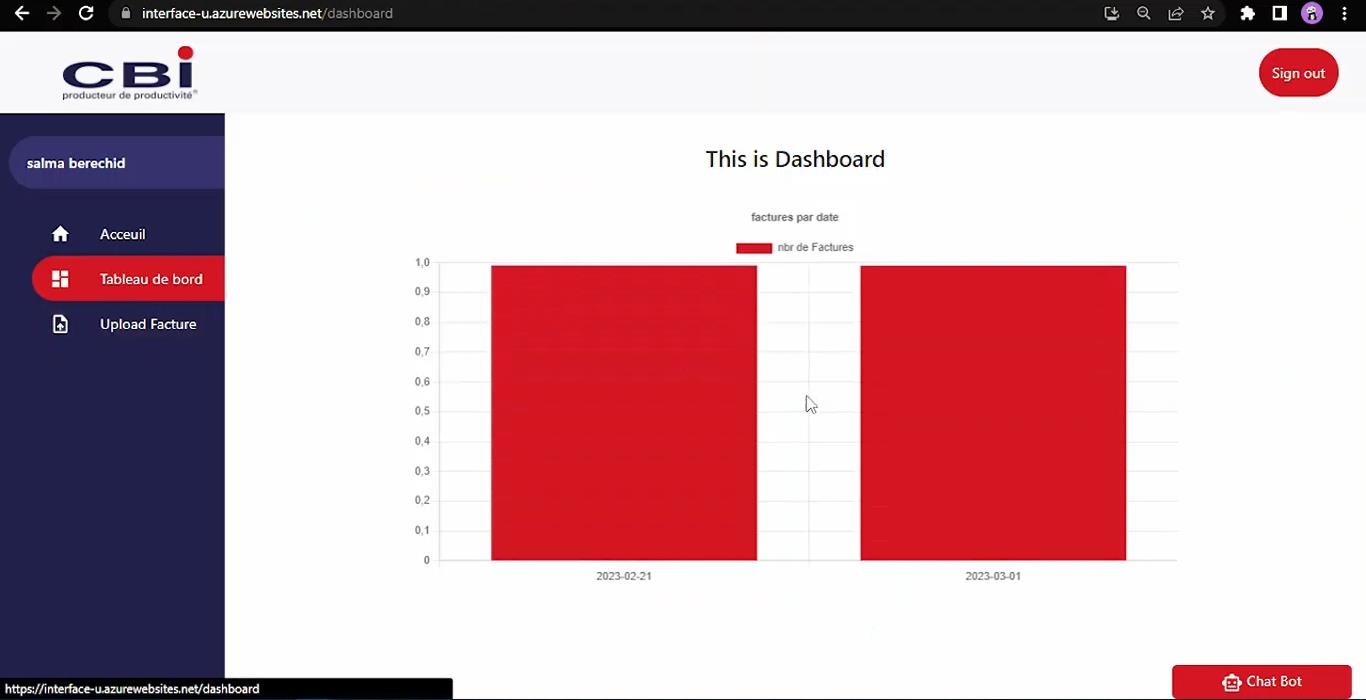
Cette interface est pour upload les factures, les océriser et stocker les informations dans la base de données.



**Figure 43 : Interface d'Upload Facture**

##### Interface tableau de bord :

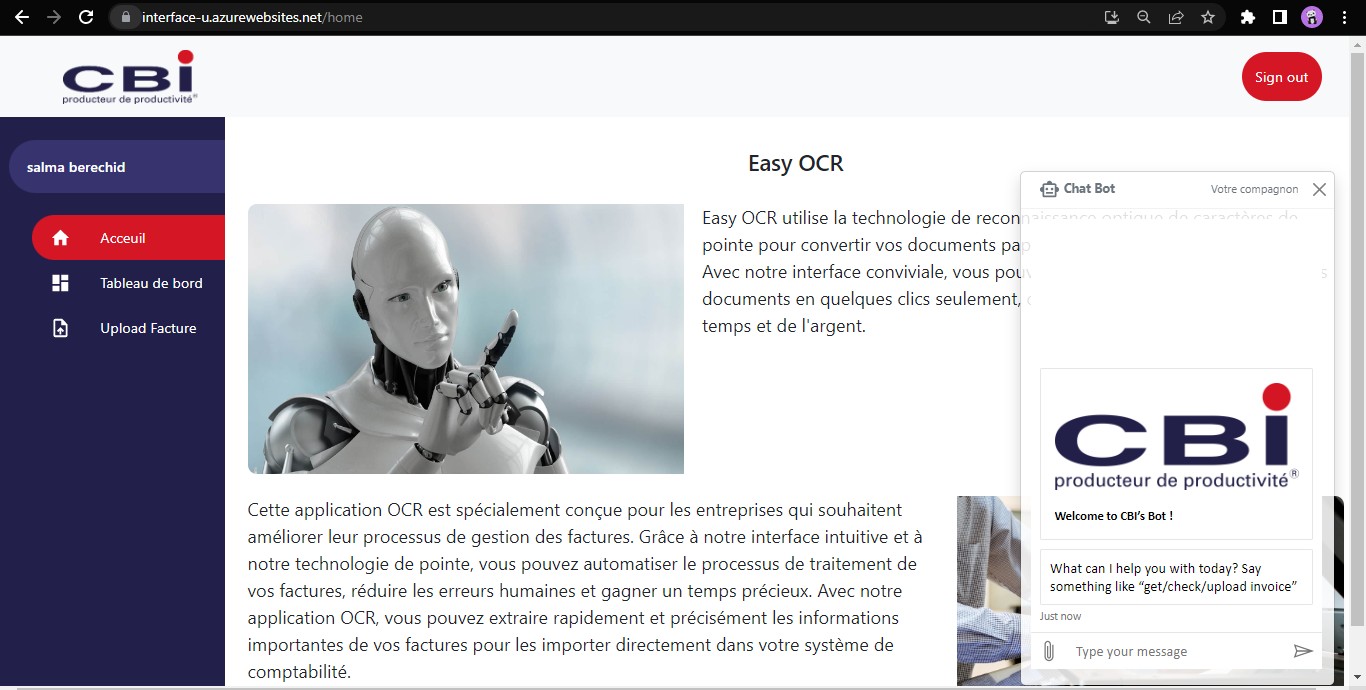
L’interface suivante présente le tableau de bord de l’application.



**Figure 44 : Interface Tableau de bord**

##### Interface chatbot :

Cette interface présente le chatbot qui a comme rôle décrire l’application a l’utilisateur. Il aide à la circulation dans l’application



**Figure 45 : Interface Chatbot**

#### Conclusion :

Dans ce chapitre on a vu les solutions proposées pour moderniser des applications à l’échelle du Cloud Azure et on-premise.

### Conclusion Générale

Notre mission au sein de la société CBI était de déployer une application vers le cloud Azure et On-premise.

Ce stage a été extrêmement bénéfique pour compléter notre formation en Master. Il nous a permis de mettre en pratique les connaissances et les méthodes de travail que nous avons acquises tout au long de nos études, en les confrontant à la réalité du monde professionnel. En effet, ce stage nous a offert l'opportunité de développer des compétences relationnelles essentielles pour notre future carrière. Nous avons dû travailler sur notre expression orale afin de communiquer de manière claire et concise, et nous avons appris à hiérarchiser les différents modes de communication utilisés en entreprise.

Nous nous sommes sentis très bien accueillis par les personnes de l’organisme parce que l'environnement de travail est à la fois très professionnel et très humain. Nous avons compris l'importance d'être à l'aise dans son environnement de travail et d'apprécier les collègues en dehors des contextes strictement professionnels.. Ce stage nous a permis de nous épanouir aussi bien dans le plan professionnel que personnel.

En termes de perspectives, on envisage de travailler avec le service d’Azure nommé Azure Cognitives Services pour traiter toutes les genres de factures et les océriser. Ainsi, on va travailler avec terraform pour automatiser la création des services sur Azure et les configurer.

### Webographie & Bibliographie

*Accueil*. <https://www.cbi.ma/>.

*Argo CD - Declarative GitOps CD for Kubernetes*. <https://argo-cd.readthedocs.io/en/stable/>.

asudbring. *Create an Ingress Controller in Azure Kubernetes Service (AKS) - Azure Kubernetes Service*. 9 mars 2023, [https://learn.microsoft.com/en-us/azure/aks/ingress-basic.](https://learn.microsoft.com/en-us/azure/aks/ingress-basic)

*Azure DevOps Services | Microsoft Azure*. [https://azure.microsoft.com/en-us/products/devops.](https://azure.microsoft.com/en-us/products/devops)

dlepow. *Authorize Test Console of API Management Developer Portal Using OAuth 2.0 User Authorization - Azure API Management*. 9 janvier 2023, [https://learn.microsoft.com/en-](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-oauth2)

[us/azure/api-management/api-management-howto-oauth2.](https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-oauth2)

*Documentation for Visual Studio Code*. <https://code.visualstudio.com/docs>.

*How to Deploy React Application to Microsoft Azure WebApp (App Service)? | CodingDeft.Com*. [https://www.codingdeft.com/posts/react-deploy-microsoft-azure/.](https://www.codingdeft.com/posts/react-deploy-microsoft-azure/)

JonathanFingold. *Create a Basic Bot - Bot Service*. 1 novembre 2022, [https://learn.microsoft.com/en-us/azure/bot-service/bot-service-quickstart-create-bot.](https://learn.microsoft.com/en-us/azure/bot-service/bot-service-quickstart-create-bot)

*Create an Azure Bot Resource in the Azure Portal - Bot Service*. 6 août 2022, [https://learn.microsoft.com/en-us/azure/bot-service/abs-quickstart.](https://learn.microsoft.com/en-us/azure/bot-service/abs-quickstart)

*Provision and Publish a Bot in Azure - Bot Service*. 8 mars 2023, [https://learn.microsoft.com/en-](https://learn.microsoft.com/en-us/azure/bot-service/provision-and-publish-a-bot)

[us/azure/bot-service/provision-and-publish-a-bot.](https://learn.microsoft.com/en-us/azure/bot-service/provision-and-publish-a-bot)

« Kubernetes Documentation ». *Kubernetes*, [https://kubernetes.io/docs/home/.](https://kubernetes.io/docs/home/)

*MicroK8s - Zero-ops Kubernetes for developers, edge and IoT*. <https://microk8s.io/>.

*MySQL :: MySQL 8.0 Reference Manual :: 21 InnoDB Cluster*. <https://dev.mysql.com/doc/refman/8.0/en/mysql-innodb-cluster-introduction.html>.

*Portail Microsoft Azure | Microsoft Azure*. [https://azure.microsoft.com/fr-fr/get-started/azure-](https://azure.microsoft.com/fr-fr/get-started/azure-portal)

[portal.](https://azure.microsoft.com/fr-fr/get-started/azure-portal)

*Prometheus Kubernetes | Kubernetes Monitoring | What is Grafana*. <https://k21academy.com/docker-kubernetes/prometheus-grafana-monitoring/>.

« PyCharm : l’IDE Python pour développeurs professionnels par JetBrains ». *JetBrains*, [https://www.jetbrains.com/fr-fr/pycharm/.](https://www.jetbrains.com/fr-fr/pycharm/)

*Securing Backend APIs Using Azure API Management Policies (Part 2) – Implementing Tae of*

*<T>*. [https://taerimhan.com/securing-backend-apis-using-azure-api-management-policies-](https://taerimhan.com/securing-backend-apis-using-azure-api-management-policies-part-2/)

[part-2/.](https://taerimhan.com/securing-backend-apis-using-azure-api-management-policies-part-2/)