Sommaire

[Introduction 2](#_Toc359847389)

[1. Terminologies cloud 2](#_Toc359847390)

[a. Modèles de cloud 2](#_Toc359847391)

[b. Caractéristiques d’une offre cloud 2](#_Toc359847392)

[2. Couches principales du cloud 3](#_Toc359847393)

[a. IaaS 3](#_Toc359847394)

[b. PaaS 3](#_Toc359847395)

[i. APaaS 3](#_Toc359847396)

[c. SaaS 3](#_Toc359847397)

[d. XaaS 3](#_Toc359847398)

[3. Présentation de certains leader PaaS du marché 4](#_Toc359847399)

[a. Heroku de SalesForces 4](#_Toc359847400)

[b. CloudFoundry 4](#_Toc359847401)

[c. CloudBees 4](#_Toc359847402)

[d. OpenShift Online de Red Had 4](#_Toc359847403)

[e. Windows Azure de Microsoft 4](#_Toc359847404)

[f. Amazon Web Services EC2 4](#_Toc359847405)

[g. Engine Yard Cloud 4](#_Toc359847406)

[h. CloudControl 4](#_Toc359847407)

[i. AppHarbor 4](#_Toc359847408)

[j. Cloudify de Gigaspace 4](#_Toc359847409)

[k. Static 4](#_Toc359847410)

[4. Comparaison des PaaS 4](#_Toc359847411)

[a. Caractéristiques générales 5](#_Toc359847412)

[b. Comparaison par les Framework supportés 8](#_Toc359847413)

[c. Comparaison par type de modules complémentaire disponible en extension 9](#_Toc359847414)

[d. Comparaison par les cas d’usage réalisés. 10](#_Toc359847415)

[User Stories 11](#_Toc359847416)

[1 11](#_Toc359847417)

[2 11](#_Toc359847418)

[3 11](#_Toc359847419)

[4 11](#_Toc359847420)

[Afin de vérifier qu’il n’y a pas de régression de la qualité d’une application à chaque modification du code source. 11](#_Toc359847421)

[5 11](#_Toc359847422)

[6 11](#_Toc359847423)

[7 11](#_Toc359847424)

[8 11](#_Toc359847425)

[9 11](#_Toc359847426)

[10 11](#_Toc359847427)

[11 11](#_Toc359847428)

[12 11](#_Toc359847429)

[13 11](#_Toc359847430)

[14 12](#_Toc359847431)

[15 12](#_Toc359847432)

[16 12](#_Toc359847433)

[17 12](#_Toc359847434)

[18 12](#_Toc359847435)

[19 12](#_Toc359847436)

# Introduction

# Terminologies cloud

## Modèles de cloud

Il existe quatre modèles de déploiement cloud.

* Le cloud privé
* Le cloud publique
* Le cloud hybride
* Le cloud communautaire

## Caractéristiques d’une offre cloud

Une offre cloud présente certaines caractéristiques essentielles.

* Libre-service à la demande
* L’offre cloud doit être accessible sur l’ensemble d’un réseau
* Les ressources sont mutualisées
* Le cloud doit s’adapter rapidement aux différentes variations des besoins lors de l’utilisation des ressources
* Le service doit être mesurable

# Couches principales du cloud

## IaaS

La totalité de l'infrastructure (ressources matérielles) est externe. Par exemple, capacité de stockage et capacité de calcul à la demande sur un réseau. Le client accède à une partie à des ressources pour créer et utiliser sa propre infrastructure dont il a besoin, quand il veut et comme il le veut. L'IaaS est le premier modèle de cloud, où :

* **Le fournisseur Cloud** maintient : la virtualisation, le matériel serveur, capacité de stockage et capacité de calcul à la demande sur un réseau;
* **L'entreprise cliente** maintient : les applications, les environnements de développement et d'exécution d'applications (incluant systèmes d’exploitation), l'intégration SOA, les bases de données, le logiciel serveur.

L’IaaS est le modèle adapté aux entreprises souhaitant utiliser leur propre cloud sans avoir à supporter des investissements en infrastructure. Le client IaaS peut être un fournisseur PaaS.

## PaaS

Le client utilise la plateforme PaaS pour porter ses applications dans le cloud ou en développer d'autres à partir des outils de développement fournis par la plate-forme de Cloud Computing. La plateforme PaaS est le second modèle de cloud, où :

* **Le fournisseur Cloud** maintient : les applications, les environnements de développement et d'exécution d'applications (incluant systèmes d’exploitation), l'intégration SOA, les bases de données, le logiciel serveur, la virtualisation, le matériel serveur, le stockage, les réseaux. Exemple de fournisseur : OpenShift qui propose une suite de logicielle Linux+Apache+MySQL+Java;
* **L'entreprise cliente** maintient uniquement les applications qu’il déploie au travers des services, librairies, langages fournis. Exemple de services fournis: base de données MySQL, serveur JBoss.

Le PaaS est le modèle adapté aux entreprises souhaitant contrôler le déploiement de ses applications ou qui veulent développer leurs propres applications. Le client ici fournit l’application finale.

### APaaS

**APaaS** pour application Platform-as-a-Service : environnement de développement plus facile à apprendre et utiliser que le PaaS, délivrant des applications paramétrables. L'APaaS fournit des services intégrés de développement, déploiement et gestion, optimisés pour un environnement de cloud computing et prenant en charge la fourniture de l'application finale qui est distribué comme un service (SaaS).

## SaaS

SaaS pour « Services-as-a-Software » est l'ultime modèle de cloud, où :

* **Le fournisseur Cloud** maintient : les applications qu’il déploie, les environnements de développement et d'exécution d'applications (incluant systèmes d’exploitation), l'intégration SOA, les bases de données, le logiciel serveur, la virtualisation, le matériel serveur, le stockage, les réseaux ;
* **Le client** accède aux applications et services via un navigateur internet. Par exemple, ce peut être une banque qui loue un logiciel de comptabilité en ligne, à la demande.

## XaaS

qsdfq

# Présentation de certains leader PaaS du marché

## Heroku de SalesForces

### Présentation

Heroku a été fondée en  2007 rachetée en 2010 par Salesforces. Heroku étend alors son offre  au développement Java. C’est une plateforme de d’application cloud initialement prévue pour les langages Ruby, Python, Scala et Clojure. La zone Europe est disponible depuis avril 2013 sur le Datacenter d’Amazon situé en Irlande (Heroku dépend donc des pannes Amazon !); ce qui permet un gain de 100ms par requête par rapport à la zone US.

La philosophie de Heroku, c’est d’augmenter la productivité du développeur en minimisant le nombre d’étapes, en le permettant de passer moins de temps dans la gestion des serveurs d’applications, le déploiement ou la configuration de la scalabilité.

### Fonctionnement

Heroku est une plate-forme PaaS pour construire, déployer et exécuter des applications sur cloud. L’architecture de la plate-forme inclut des outils pour le déploiement et la gestion (plugin et SDK pour l'exécution en ligne de commande), un environnement d'exécution ainsi que des modules complémentaires.

Les différentes parties qui décrivent le fonctionnement sont

* Déploiement des fichiers binaires ;
* Connexion des modules complémentaires ;
* L’interface de contrôle composée d’une CLI, d’une console Web, d’une API et d’un plugin IDE pour permettre la gestion des applications, des ressources, des utilisateurs, noms de domaines, journalisation etc. ;
* Observaé(tion

## CloudFoundry

## CloudBees

## OpenShift Online de Red Had

## Windows Azure de Microsoft

## Amazon Web Services EC2

## Engine Yard Cloud

## CloudControl

## AppHarbor

## Cloudify de Gigaspace

## Static

# Comparaison des PaaS

## Caractéristiques générales

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Critères de comparaison | Fournisseur Cloud PaaS | | | | | | | | | | |
|  | Heroku | CloudFoundry | CloudBees | OpenShift | Azure | AWS EC2 | Engine Yard Cloud | CloudControl | AppHarbor | Cloudify | Static |
| Fournisseur | SalesForces | Pivotal | CloudBees | Red Hat | Microsoft | Amazon | Engine Yard | CloudControl |  | Gigaspace | Static |
| Type de logiciel | Open Source | Open Source |  | Open Source | Propriétaire |  | Open Source | Open et Propriétaire |  | Open Source dans la majorité des composants | Open Source |
| Système d’exploitation | Ubuntu 10.04 | Ubuntu |  | Red Hat Entreprise Linux | Windows server | RHEL, Windows Server, Oracle EL, Ubuntu, SUSE LE, AMI Amazon Linux, Debian, Fedora, Gentoo Linux | Gento Linux | Ubuntu [\*](https://www.cloudcontrol.com/dev-center/Platform%20Documentation#available-stacks) | Windows Server 2008 R2 (x64) |  | Ubuntu, Debian, FreeBSD, CentOS |
| Datacenter | AWS US, AWS EU Irlande |  | Par défaut : AWS US.  Et localisation selon le souhait avec AnyCloud, AWS US & UE, OpenStack, HP Cloud Services, Entreprise Datacenter |  | Azure US centre-nord, centre-sud, ouest, est. Asie de l’est, sud-est. Europe de l’ouest, nord. [\*](http://en.wikipedia.org/wiki/Windows_Azure#Datacenters) | AWS US est, ouest, AWS UE, AWS Asie-Pacifique, AWS Amérique du sud, AWS US GovCloud  9 régions au total | Tous les Datacenter AWS EC2 US et UE. | AWS Irlande, ou alternativement dans le Datacenter souhaité | AWS à l’est des US | AWS, OpenStack, Rackspace, Windows Azure, HP Cloud VMware, Citrix Cloud Stack1, Terramark, XenServe. Peut être installé sur tous cloud supportant JClouds API | AWS, Rackspace : US, Hong Kong, Royayme Unis et Australie, DigitalOcean : NY et Amsterdam, Static Datacenter |
| Types de cloud | Publique | Publique, privé, hybride | Publique, privé | Publique, privé | Publique | Privé, publique | Publique, travaille pour un future PaaS privé et hybride | Publique, privé, hybride |  | Publique, privé, hybride | Publique, privé |
| Fourniture Self-service | Oui, on peut choisir 512Mo ou 1024Mo, avec stockage personnalisable | Oui | Oui | Oui | Oui | Oui | Oui | Oui, la mémoire et tout autre service complémentaire | Oui | Oui | Oui |
| Disponibilité du système / Service Level Agreement SLA | 99,22%  à 99,99%, 99,97% en moyenne | N/A | 99,9% |  | 99,95% [\*](http://www.windowsazure.com/fr-fr/support/legal/sla/) | 99,95% [\*](http://aws.amazon.com/fr/ec2-sla/) | 99,9% (à travers le support Premium) | 100% [\*](http://checks.paaswatch.com/600030/history) | 99,95% | Haute disponibilité et auto-recouvrement des applications | 24/7/365 |
| Sécurité des données | Les applications s’exécutent dans des environnements séparés. Les données sont stockées dans des BDs d’accès contrôlées. [\*](https://policy.heroku.com/security) |  |  |  |  |  | https://www.engineyard.com/products/cloud/cloud-security | https://www.cloudcontrol.com/privacy-policy |  | Sécurité par authentification, autorisation et dans le transport des données | Conformité PCI de niveau 3 et réalisation d’audits de sécurité |
| Application privées | Oui | Non, applications accessible par tous | Oui |  | Oui | Oui | Non, mais application derrière un pare-feu, ou doit être implémenté. | Non, mais doit être implémenté au niveau de l’application dans le cas du PaaS publique |  |  | Oui |
| Environnements | Un seul confondu | Développement, test, et autres environnements que l’on peut ajouter | Développement, local et production [\*](http://wiki.cloudbees.com/bin/view/RUN/CloudBeesWebXml#HApplicationEnvironments) | Un seul |  | Développement, production… | Développement, production, staging, et d’autre à ajouter | Développement, staging et production |  |  | Production, test et développement, possibilité d’ajouter d’autres environnements |
| Interopérabilité | Oui | Oui | Oui |  |  | Oui, avec plusieurs autres services cloud Amazon | Oui | Oui |  | Oui | Oui, avec d’autre fournisseur Cloud comme Amazon, Rackspace et DigitalOcean |
| Applications portables / No Lock-in | Oui, no-lock in pour la plateforme | Oui | Oui | Oui | Non, l’architecture des applications est propriétaire | Oui, grâce au large choix de OS,  Framework, serveur d’applications | Oui | Oui, les applications peuvent être exécutées sur des PaaS compatible buildpack-API (Heroku, Stackato…) ou autre serveur régulier |  | Oui, no-lock in concernant les PaaS, OS et code. | Oui |
| Polyglotte ou non | Oui.  Ruby est le plus utilisé, puis Python (l’architecte en chef Ruby est le créateur du langage) | Oui | Oui | Oui | Oui | Oui | Oui. Ruby est le plus utilisé. PHP est encore nouveau pour cette plateforme. | Oui, tout langage utilisé sur un environnement Linux. Les plus utilisés dans la plateforme sont Java, Ruby, Python et PHP | Que le Framework .NET | Oui | Oui |
| Mode de scalabilité | Automatique | Automatique |  | Automatique, manuelle, ou pas de scalabilité | Automatique ou manuelle | Automatique, avec frais supplémentaires (Amazon CloudWatch) | Automatique avec paramétrage | Automatique avec paramétrage | Automatique | Automatique avec règles à définir. | Automatique |
| Load Balancing | Oui [\*](https://devcenter.heroku.com/articles/http-routing#routing) | Oui | Oui, automatique | Oui | Oui | Oui avec frais supplémentaires (Elastic Load Balancing) | Oui. A activer manuellement |  | Oui |  |  |
| Politique en cas de panne |  |  |  |  |  |  | Des volumes EBS (Elastic Block Storage) sont montés pour l’application et les serveurs de base de données. Possibilité d’avoir une base de données en master et slave | Le système est déployé à travers plusieurs zones disponibles, donc une panne peu probable. La plateforme peut résoudre les problèmes avant que les utilisateurs finaux en soient affectés |  | Auto recouvrement |  |
| Plateforme mobile avec support d’application | Oui | Non | Oui, Android, iOs |  | Oui, Windows 8, iOs, Android | Oui | Oui avec quelques modules complémentaires [\*](https://www.engineyard.com/docs/Mobile_Solutions_Brief.pdf) | Oui | Non |  |  |
| Outils d’interaction | Console web, CLI Heroku Toolbelt, plugin Heroku | Console web, CLI CF, plugin CloudFoundry, API | Console web Grand Central, CLI bees, plugin CloudBees | Console web, CLI RHC, plugin OpenShift via JBoss | Console Web, plugin Azure pour IDE, API, Windows Azure PowerShell | Console Web, AWS Toolkit pour Eclipse, AWS Toolkit pour Visual Studio | Console Web et CLI EY | CLI CCTRL, Console Web, API | Console web, API | CLI Cloudify Shell, Console web | Console Web, CLI STATIC |
| Services complémentaires | Oui, en grand nombre | Oui, mais encore très peut |  | Oui | Oui | Oui | Oui | Oui | Oui | Oui |  |
| Méthode de reporting |  | Pas de reporting |  |  |  |  | Alertes mails à activer. Alertes sur les ressources utilisées |  |  | Envoi d’alertes et évènements |  |
| Méthodes de Gestion des ressources | Via la console web seule | Via le plugin |  | Console web, CLI |  | Via la console web et de façon automatique | Via la console web et CLI |  | Via la console web | Via la console web et CLI |  |
| Prix |  |  |  | <https://www.openshift.com/products/pricing> |  |  | Le prix dépend de plusieurs facteurs : nombre et taille des serveurs, support technique, bande passante, nombre de backups. [\*](https://www.engineyard.com/products/cloud/pricing) |  | https://appharbor.com/pricing | 0.03$ /Go/hr | http://www.static.com/pricing/cloud-hosting/ |
| Compte gratuit |  | Version d’essai de 60 jours. |  | OpenShift Online : 1,5Go de mémoire et 3Go de stockage | 150€ gratuits, mais carte de crédit demandée à l’inscription. | Niveau gratuit de 750h/mois d’utilisation d’instances | Version d’essai de 500h |  | Oui, 1 unité de travail |  | Oui, $10 de crédit gratuit |
| Disponibilité du support technique |  |  |  | Support de la communauté et professionnel |  |  | Support standard en live, support premium. |  | Support sur le site AppHarbor et Stackoverflow.com | Support par téléphone 24/7, support de la communauté |  |

## Comparaison par les Framework supportés

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Frameworks | Fournisseur Cloud PaaS | | | | | | | | | | |
|  | Heroku | CloudFoundry | CloudBees | OpenShift | Azure | AWS EC2 | Engine Yard | CloudControl | AppHarbor | Cloudify | Static |
| Spring |  |  |  |  |  |  |  |  |  |  |  |
| Java EE 6 Web Profile |  |  |  |  |  |  |  |  |  |  |  |
| Glass Fish App Server |  |  |  |  |  |  |  |  |  |  |  |
| Jetty |  |  |  |  |  |  |  |  |  |  |  |
| Zend PHP |  |  |  |  |  |  |  |  |  |  |  |
| Symphony |  |  |  |  |  |  |  |  |  |  |  |
| .NET |  |  |  |  |  |  |  |  |  |  |  |
| ASP. |  |  |  |  |  |  |  |  |  |  |  |
| Java |  |  |  |  |  |  |  |  |  |  |  |
| Play 2.0 |  |  |  |  |  |  |  |  |  |  |  |
| Lift |  |  |  |  |  |  |  |  |  |  |  |
| Grails |  |  |  |  |  |  |  |  |  |  |  |
| Python |  |  |  |  |  |  |  |  |  |  |  |
| Rails |  |  |  |  |  |  |  |  |  |  |  |
| Grape |  |  |  |  |  |  |  |  |  |  |  |
| Rack |  |  |  |  |  |  |  |  |  |  |  |
| Sinatra |  |  |  |  |  |  |  |  |  |  |  |
| Node.js |  |  |  |  |  |  |  |  |  |  |  |
| Liferay |  |  |  |  |  |  |  |  |  |  |  |
| JBoss AS |  |  |  |  |  |  |  |  |  |  |  |
| JBoss EAP |  |  |  |  |  |  |  |  |  |  |  |
| Tomcat |  |  |  |  |  |  |  |  |  |  |  |
| Django |  |  |  |  |  |  |  |  |  |  |  |
| Flask |  |  |  |  |  |  |  |  |  |  |  |
| Express |  |  |  |  |  |  |  |  |  |  |  |
| Clojure |  |  |  |  |  |  |  |  |  |  |  |
| Erlang |  |  |  |  |  |  |  |  |  |  |  |
| Google Go Webapp |  |  |  |  |  |  |  |  |  |  |  |
| Facebook App |  |  |  |  |  |  |  |  |  |  |  |
| iUI web Apps |  |  |  |  |  |  |  |  |  |  |  |
| Dart App |  |  |  |  |  |  |  |  |  |  |  |
| Compojure |  |  |  |  |  |  |  |  |  |  |  |
| Metor.js |  |  |  |  |  |  |  |  |  |  |  |
| Backbone.js |  |  |  |  |  |  |  |  |  |  |  |
| Angular JS |  |  |  |  |  |  |  |  |  |  |  |
| Railo (CFML) |  |  |  |  |  |  |  |  |  |  |  |
| Procfile |  |  |  |  |  |  |  |  |  |  |  |
| Perl |  |  |  |  |  |  |  |  |  |  |  |
| CakePHP |  |  |  |  |  |  |  |  |  |  |  |
| Reveal.js |  |  |  |  |  |  |  |  |  |  |  |
| WordPress |  |  |  |  |  |  |  |  |  |  |  |
| Drupal |  |  |  |  |  |  |  |  |  |  |  |
| Joomla |  |  |  |  |  |  |  |  |  |  |  |
| Switchyard |  |  |  |  |  |  |  |  |  |  |  |
| Cron |  |  |  |  |  |  |  |  |  |  |  |
| Bootstrap |  |  |  |  |  |  |  |  |  |  |  |
| FuelPHP |  |  |  |  |  |  |  |  |  |  |  |
| Lavarel |  |  |  |  |  |  |  |  |  |  |  |
| Kohana |  |  |  |  |  |  |  |  |  |  |  |
| WSGI |  |  |  |  |  |  |  |  |  |  |  |
| Rack |  |  |  |  |  |  |  |  |  |  |  |

## Comparaison par type de modules complémentaire disponible en extension

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Services complémentaires | Fournisseurs | | | | | | | | | | |
|  | Heroku [\*](https://addons.heroku.com/) | CloudFoundry | CloudBees [\*](http://www.cloudbees.com/platform/ecosystem/tech-partners.cb) | OpenShift | Azure | AWS EC2 | Engine Yard | CloudControl | AppHarbor | Cloudify | Static |
| Analyse de performance |  |  |  |  | AppDynamics |  |  |  |  |  |  |
| Test | Blitz |  |  |  |  |  |  |  |  |  |  |
| SCM | Seulement hébergé |  | Git, SVN | Git hébergé ou personnel |  | Subversion, GitLab … 18 au total | Propose github, mais peut être changé |  | Bitbucket, CodePlex, et GitHub |  |  |
| ALM |  |  |  |  |  |  |  |  |  |  |  |
| Intégration continue | Tddium, Codeship, Semaphore |  | Jenkins | Jenkins | Non mais un plugin Windows Azure Storage peut être installé sur Jenkins |  |  |  |  |  |  |
| Gestion de projet |  |  |  |  |  |  |  |  |  |  |  |
| Bases de données SQL | PostgreSQL |  |  |  | ClearDB |  |  |  |  |  |  |
| Bases de données NoSQL | MongoDB |  |  |  | Table Service |  |  |  |  |  |  |
| Logging | Logentries |  |  |  |  |  |  |  |  |  |  |
| Email | Mailgun |  |  |  | SendGrid |  |  |  |  |  |  |
| SMS | Blower.io |  |  |  | Twilio |  |  | + MMS |  |  |  |
| Queuing | IronMQ |  |  |  | Service Bus Queues, Windows Azure Queue Service |  |  |  |  |  |  |
| Cache | MemCachier |  |  |  | Windows Azure Caching |  |  |  |  |  |  |
| Monitoring | New Relic |  |  |  | New Relic |  |  |  |  |  |  |
| Paiement | Spreedly |  |  |  |  |  |  |  |  |  |  |
| Collaboration | Wercker |  | xWiki | Wercker | VS Anywhere |  |  |  |  |  |  |
| Analyse de données Big Data | Treasure Data |  |  |  |  |  |  |  |  |  |  |
| Recherche | ElasticSearch |  |  |  | Bing Search API |  |  |  |  |  |  |
| Analyse de codes sources |  |  |  |  |  |  |  |  |  |  |  |
| Continious delivery | Wercker |  |  |  |  |  |  |  |  |  |  |

## Comparaison par les cas d’usage réalisés.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Cas d’usages | Fournisseurs | | | | | | | | | | |
|  | Heroku | CloudFoundry | CloudBees | OpenShift | Azure | AWS EC2 | Engine Yard | CloudControl | AppHarbor | Cloudify | Static |
| 1 |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  | Avec Jenkins | Avec Jenkins |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Seulement avec le gestionnaire Git hébergé |  | Possibilité d’utiliser son propre SCM | SCM au choix |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |
| 11 |  |  |  |  |  |  |  |  |  |  |  |
| 12 | Contrôle de la taille mémoire CPU seulement, les DNS | Contrôle du nombre d’instances et de la taille de la mémoire, les domaines | Contrôle du nombre d’instances, domaines | Contrôle du nombre de Gears, de la taille de stockage et de la scalabilité, les alias |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  |
| 15 |  |  |  |  |  |  |  |  |  |  |  |
| 16 | Interaction par un SDK, et IDE |  | Interaction avec API, SDK et IDE |  |  |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |  |  |  |  |  |
| 18 | Système de backup intégré |  |  |  |  |  |  |  |  |  |  |
| 19 | CloudVertical Beta |  | CloudBees Metering and Billing |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
|  | User Stories |
| 1 | En tant que développeur,  Je souhaite mettre en place un environnement de développement déterminé (serveur ou conteneur d’application, base de données, etc.) sur le cloud,  Afin de développer une application. |
| 2 | En tant que développeur,  Je souhaite utiliser environnement de développement existant sur le cloud,  Afin de participer au développement d’une application. |
| 3 | En tant que développeur,  Je souhaite mettre en place un gestionnaire de dépendances  Afin de disposer d’un dépôt de librairies partagées et de proximités |
| 4 | En tant que développeur,  Je souhaite mettre en place un environnement d’intégration continue sur le cloud, Afin de vérifier qu’il n’y a pas de régression de la qualité d’une application à chaque modification du code source. |
| 5 | En tant que développeur,  Je souhaite analyser le code source de mon application et en obtenir les métriques de qualités,  Afin d’évaluer rapidement la dette technique de mon application et recenser les corrections à apporter |
| 6 | En tant que développeur,  Je souhaite stocker et gérer les versions de mes fichiers,  Afin de partager mes fichiers et conserver l’historique des modifications. |
| 7 | En tant que chef de projet,  Je souhaite planifier, piloter, suivre l’avancement et gérer les ressources d’un projet,  Afin de m’assurer de la conformité par rapport aux plans initiaux. |
| 8 | En tant que membre du projet,  Je souhaite mémoriser et organiser mes connaissances,  Dans le but de capitaliser et les partager avec tous mes collaborateurs. |
| 9 | En tant que testeur ou développeur,  Je souhaite mettre en place un système de suivis d’anomalies  Afin de corriger des erreurs fonctionnelles ou techniques d’une application. |
| 10 | En tant que testeur,  Je souhaite mettre en place un environnement de test (serveur ou conteneur d’application, base de données, etc.) sur le cloud,  Afin de tester une application dans un environnement déterminé (homologation, recette, pré-production). |
| 11 | En tant que développeur,  Je souhaite identifier, partager et centraliser les versions de mon logiciel binaire de ses dépendances  Afin d’éviter les conflits et les erreurs liées aux versions des logiciels et des dépendances |
| 12 | En tant CP,  Je souhaite configurer les ressources allouées sur le cloud  Afin de mieux en contrôler le coût |
| 13 | En tant qu’administrateur,  Je souhaite choisir des fournisseurs d’infrastructures différents pour mon environnement d'exécution  Afin de contrôler les coûts de consommation. |
| 14 | En tant qu’expert,  Je souhaite surveiller, diagnostiquer une application  Afin de contrôler ses performances |
| 15 | En tant qu’architecte,  Je souhaite modéliser mon système par des vues logiques, d'implémentations, de déploiements et de processus,  Afin de mieux faire comprendre et avoir une meilleure visibilité du système. |
| 16 | En tant que développeur,  Je souhaite interagir avec la plateforme cloud avec mon code source (via une API), avec un SDK (exemple de RHC) ou un IDE,  Afin d’ajouter à mon application un nouveau service. |
| 17 | En tant que décideur,  Je souhaite collecter et analyser mes données Big Data  Afin d’avoir une vue d’ensemble sur une activité et de pouvoir prendre des décisions. |
| 18 | En tant qu’administrateur,  Je souhaite sauvegarder les images de mes fichiers, base de données et applications  Afin de pouvoir les restaurer en cas de perte de données ou de défaillance du système. |
| 19 | En tant CP,  Je souhaite consulter un rapport de la consommation en ressources / être prévenu d’un seuil...  Afin de maîtriser leurs coûts |