Cahier de recette – MongoCloud v 0.2

Projet informatique

European
Secured
Database

Semestre 5
Systèmes et réseaux

E.S.D vous propose d'héberger vos données de consentement en Europe.

intech
ECOLE SUPERIEURE
D'INGENIERIE INFORMATIQUE

Date de diffusion	Edité par	Intitulé
17/12/2018	Vincent DELTEL	Document initial

Table des matières

1. Objet 3

2. Introduction 3

3. Opération de recette 3

3.1 Vérification de la documentation : 4

3.2 Vérification du produit 5

3.2.1 Interface utilisateur 5

3.2.2 Fonctionnalités du produit 6

3.2.3 Essais complémentaires : 7

4. Décision du client 8

1. Objet

Ce document constitue le Cahier de Recette Client du POC MongoCloud version 0.2 issu du projet European Secured Database.

Domaine d'application : POC à destination de la société Agilitation pour valider la faisabilité d'une API permettant le stockage des données liées au Règlement Général sur la Protection des Données.

Document de référence :

- Etudes de faisabilité
- Spécifications fonctionnelles
- Charte de projet
- Compte-rendu de réunion
- Compte-rendu hebdomadaire
- Présentation d'avant-projet

2. Introduction

Ce document a pour but de présenter au client les résultats du projet MongoCloud. Il se décompose en 2 parties distinctes :

- Vérification de la présence de toute la documentation projet demandée (examen du contenu sur demande)
- Vérification des fonctionnalités du produit séparées en user stories prévues dans le backlog produit.

3. Opération de recette

La recette finale du produit s'est déroulée à partir de ……, dans les locaux d'INTECH Sud Agen le 20 décembre 2018, en présence de :

- M. Vincent DELTEL	Etudiant en systèmes et réseaux	Scrum master
- M. Florian PITANCE	Enseignant systèmes et réseaux	Suiveur
- M. Romain BESSUGES-MEUSY	Fondateur, CEO, CTO Agilitation	Product Owner

Le matériel minimum requis pour le bon déroulement de la recette est un ordinateur, un vidéoprojecteur et le serveur hébergeant le projet sur un même réseau local.

3.1 Vérification de la documentation :

Vérifier l'existence des documents suivants :

DOCUMENTS	RESULTAT (OK-NOK)
Etudes de faisabilité	
Spécifications fonctionnelles	
Charte de projet	
Spécifications techniques	
Procédure Git Kraken	
Manuel d'utilisation	
Lien Github du projet	
Compte-rendu de réunions/ hebdomadaires	

Observations :

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

3.2 Vérification du produit

Vérifier la présence des éléments suivants :

ELEMENTS CONSTITUTIFS DU PRODUIT	RESULTAT (OK-NOK)
Dossier assets	
Dossier images – 1 fichier	
Dossier include – 1 fichier	
Dossier script – 5 fichiers	
create_instance.py	
Docker_allinone.sh	
id_rsa	
index.php	
master.sh	
part2.php	
part3.php	
part4.php	
vlan_creation.py	

3.2.1 Interface utilisateur

Le projet doit-être hébergé sur un serveur local.

TÂCHE DE VERIFICATION	RESULTAT (OK-NOK)
Accéder au site web local conformément au manuel utilisateur.	

3.2.2 Fonctionnalités du produit

Entrées

USER STORIES - TÂCHE DE VERIFICATION	RESULTAT (OK-NOK)
Tom peut renseigner ses clés secrètes	
Tom peut sélectionner un projet	
Tom peut choisir un nom pour son client	
Tom peut choisir la région où les données seront hébergées	
Tom peut choisir les ressources des machines virtuelles à créer	
Tom peut sélectionner la clé SSH à utiliser	

Sorties

	Action DE VERIFICATION	RESULTAT (OK-NOK)
Un vlan est créé.	Visible dans l'interface client OVH	
Un sous-réseau est créé.	Accéder à l'URL suivante et renseigner le nom de projet et l'ID de réseau : https://api.ovh.com/console/#/cloud/project/%7BserviceName%7D/network/private/%7BnetworkId%7D#GET	
3 machines virtuelles sont créées.	Visible dans l'interface client OVH	
Chaque machine dispose d'une interface publique et d'une interface privée.	Visible en effectuant la commande « ip a » sur les machines.	
Les machines peuvent se contacter entre elles.	Effectuer un ping entre les machines « exemple : ping 192.168.1.4 »	
Les machines sont accessibles depuis une ip publique.	Effectuer une connexion ssh à l'aide de la commande « ssh –i /id_rsa debian@ippublique	
Docker est déployé sur chaque machine	Effectuer la commande « su docker » puis « docker ps » sur la machine cible.	
MongoDB est déployé sur chaque machine	Effectuer la commande Mongo --host@ippublique	

Un replica set est créé entre les machines	Effectuer mongo --host@ippublique sur la première machine créée
	Use agilitation ;
	Show collection;
	db.CreateCollection(« mabase ») ;
	db.mabase.insert({“Nom”: ”Prenom”});
	exit ;
	
	Se connecter à une machine une machine secondaire :
	mongo --host@ippublique2
	db.getMongo().setSlaveOk(true);
	show dbs;
	use agilitation;
	show collections;
	db.mabase.find();
	
	Cela renvoie le document suivant : ({“Nom”: ”Prenom”})
	La synchronisation entre les machines est active.	

3.2.3 Essais complémentaires :

Le client peut effectuer en séance tous les essais complémentaires qu’il juge nécessaire, après la réalisation des tests de bon fonctionnement ci-dessus.

4. Décision du client

MongoCloud v 0.2
☐ Validé sans remarques ☐ Validé avec remarque(s) mineure(s) ☐ Refusé
JUSTIFICATIONS DU CLIENT EN CAS DE REFUS :
REMARQUES CLIENT :
DATE : **Signature Client :** **Signatures Equipe Projet :**