**NB : en Sous-Ligné Jaune : le texte du speaker.**

**SLIDE1 –** [**Lien**](https://github.com/ylecleach/labs/raw/master/cloudco-chatbot-demo.pptx) **vers les slides**

Je vais vous presenter une demo qui a été produite pour l’évènement World of Watson, et qui met en oeuvre une compagnie d’assurance fictive “CloudCo” du 21 ème siècle.

Cette demo vise à démontrer comment une expérience de support aux utilisateurs peut être radicalement amélioré par l’utilisation d’un ChatBot base sur des services cognitifs de Watson. En plus d’améliorer la satisfaction client, il s’agit de libérer les conseillers clientèles ou agent de call-center de tâches répétitives.

Cette demo est base sur une architecture de micro services s’éxécutant sur la plateforme Bluemix, et illustre comment une application web peut intéragir avec les services Conversation et Tone Analyser de Watson.

Nous introduirons également ToolChain, le nouvel outil de DevOps de Bluemix, et qui permet d’intégrer différents plugins (Slack, Pager Duty), etc…

Je vous propose de commencer par vous montrer comment installer chez vous cette demo via ToolChain.

**DEPLOIEMENT**

<<Pre-req avant demo >>:

* Penser à bien supprimer le space cloudco et les ToolChain existantes>>
* Idéalement, pensé à avoir un backup : un environnement déjà déployé **mais non configuré** **et jamais utilisé** sur Dallas, space cloudco. (en cas de lenteur de provisioning)
* Idéalement, avoir un environnement déjà configuré – prêt à l’emploi sur Sydney, space cloudco, pour parer à toute éventualité en cas de soucis de provisionning sur les autres env.

1. Créer un space « **cloudco »** sur le datacenter London
2. ToolChain : montrer comment y accéder : **Depuis DALLAS (only),** aller dans DevOps via Toolbar Console
3. Aller sur <https://github.com/IBM-Bluemix/insurance-toolchain>
4. Cliquer sur Bouton ToolChain
5. mettre un nom unique pour la toolchain : **« cloudco42 »** (sera utilisé comme préfixe)
6. dans **GIT**/ Repository Type : choisir **« Existing »** car déjà créer les projets GitHub. (Clone par défaut)
7. dans **Delivery Pipeline** : mettre si besoin des noms uniques
8. dans **Delivery Pipeline**: choisir **London** / space **cloudco**
9. cliquer sur **Create**
10. Explication courte sur ce qui se passe : création de 4 projets sur GitHub, création de 4 pipelines, puis aller dans le **Dashboard London** dans le space **CloudCo** sur London, pour montrer le résultat de création des instances.

**=> SLIDE : 2,3,4,5,6 en 7 min.**

**S2** : Imaginons que CloudCo soit un assureur ou une Mutuelle Francaise. Vous avez acheté récemment des lunettes de vues. Aujourd’hui comment la plupart des assurés font –ils ?

**S3** : ils téléchargent la plupart du temps sur le web des documentations pour connaître leur couverture, et pour faire une demande de prise en charge (CLAIM en anglais), il doivent télécharger, le plus souvent des pré-imprimés, accompagnés de notice explicatives. Dans le meilleur des cas, il y a un bon site web, mais au final l’assuré devra envoyer sa demande de remboursement par papier, augmentant donc le temps de sa prise en charge, et devra souvent appelé ou faire un mail à son agent pour connaître le montant de remboursement qu’il lui reste. Bref, il est temps d’utiliser les technologies d’aujourd’hui pour améliorer l’expérience des clients.

**S4** : et pourquoi pas utiliser un Chatbot Intelligent pour faire l’ensemble de ces actions : questions sur mes droits, demande de remboursement, solde restant. C’est justement l’objectif de ce démonstrateur. Voyons comment est construit ce ChatBot

**S5** : Voici la première partie de l’architecture de ce démonstrateur : une application mobile ou web en responsive utilise un BOT Microservice (Node.js),, ainsi qu’une base de données Cloudant et un service Watson CONVERSATION

**CONFIGURATION**

**S6** : LETS CHAT with Watson Conversation. Mais avant nous devons terminer la configuration de notre démo par :

1. configurer le service Conversation
2. configurer le Workspace ID de ce service Conversation dans notre ChatBot.

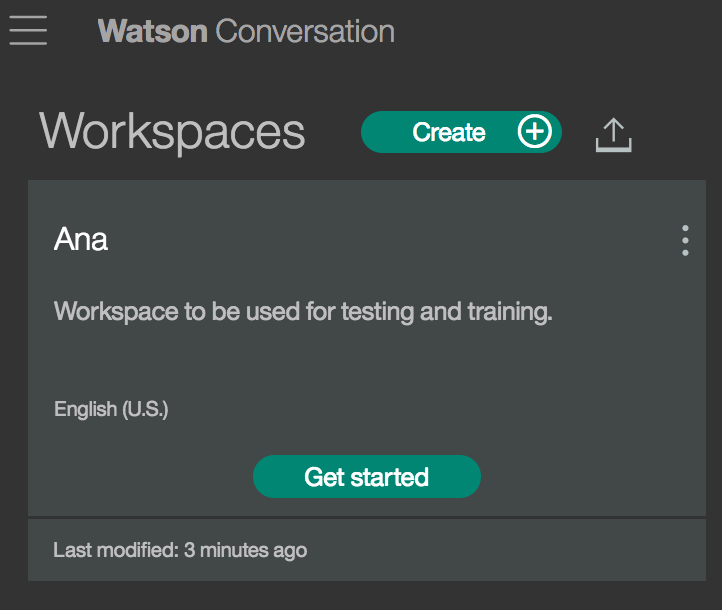
Ouvrir le Dashboard et ouvrir le detail de l’instance Watson Conversation. Expliquer brièvement ce service : « Le Service Watson Conversation permet d’ajouter une interface en langage naturel à vos application afin d’automatiserles interactions avec vos utilisateurs finaux. Les applications les plus courantes utilisant cette API proposent des assistants virtuels et des chatbots s’intégrant à n’importe quel canal ou périphériques, voir même à un robot”

Ensuite on IMPORTER le Workspace Watson Conversation utilisé pour le ChatBot appelé Ana.

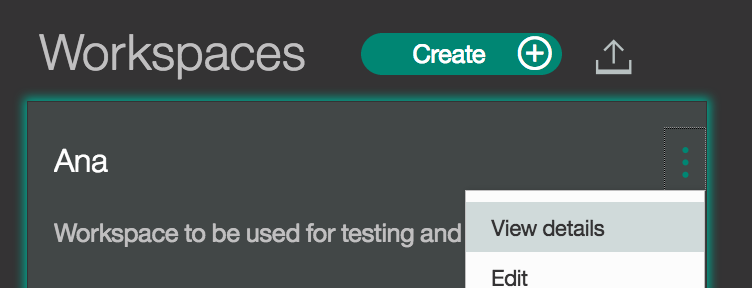
=> [Lien vers le Fichier de Configuration du ChatBot Ana](https://raw.githubusercontent.com/ylecleach/labs/master/Ana.json).

Pour copier le WorkSpace ID pour configurer l’app :

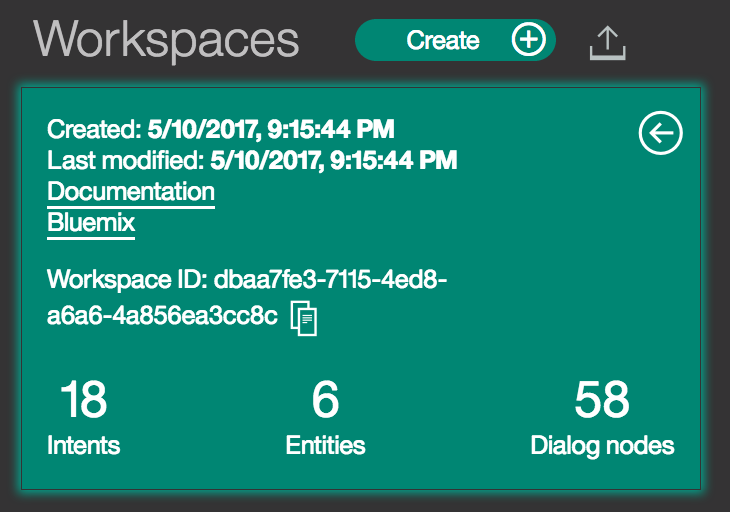
1. retourner sur le niveau Workspaces.



1. Cliquer sur View Details

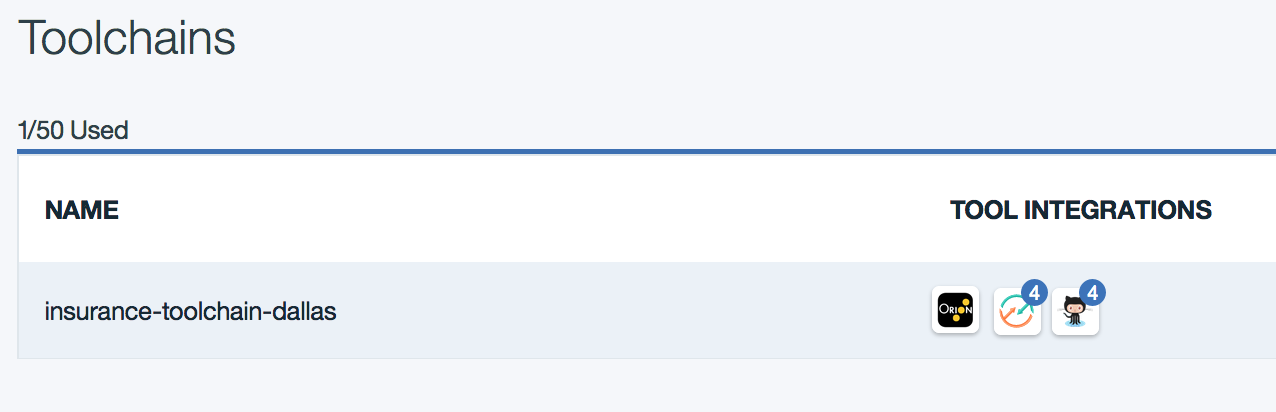


1. Cliquer sur l’icône de fichiers à droite du Workspace ID

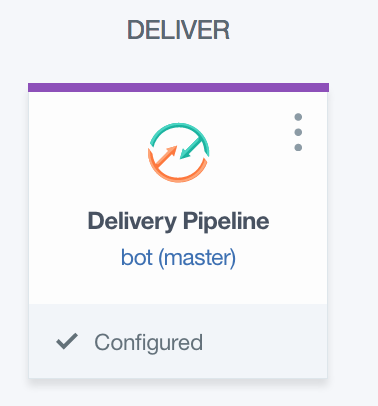


Cela à pour effet de mettre le Workspace ID dans le presse papier.

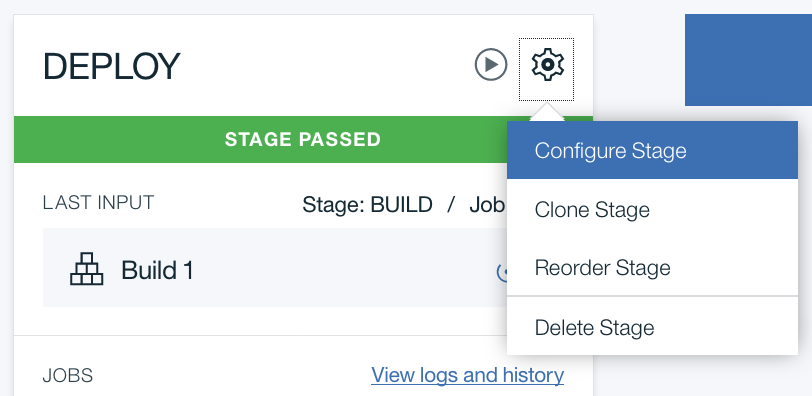
Taper : <https://console.ng.bluemix.net/devops> pour revenir au dashboard de Toolchain.



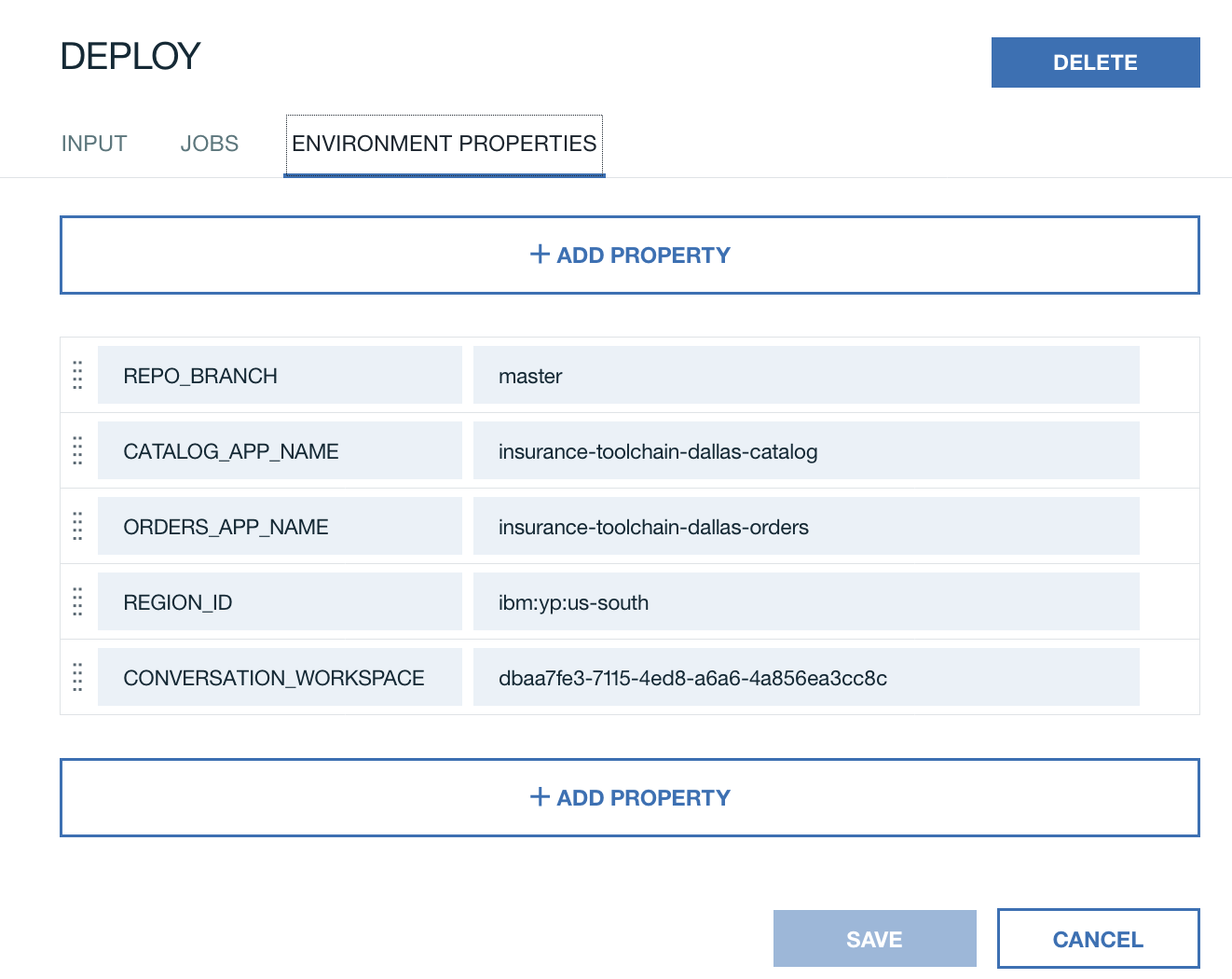
Puis cliquer sur votre Toolchain, puis cliquer sur le Delivery Pipeline « bot (master) »



Puis cliquer sur l’icone d’engrenage du step DEPLOY et sélectionner Configure Stage.

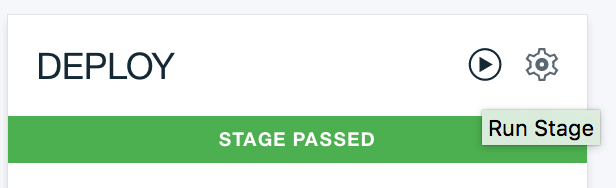


Dans l’onglet Environment, ajouter la Text Property CONVERSATION\_WORKSPACE avec comme valeur la valeur contenu dans le presse papier.



Terminer en cliquant sur le bouton SAVE

Dans l’écran précédent, lancer manuellement un redeploiement de l’application pour prendre en compte cette nouvelle variable. (icone de triangle noir)



Attendre 2-3 minutes, le temps que le Blue/Green redeploy se fasse. (permet de conserver en ligne la version précédente, jusqu’à la mise à disposition de la nouvelle version de l’application). => ZERO DOWNTIME.

* **repasser dans Conversation pour montrer un peu le modèle en expliquant**

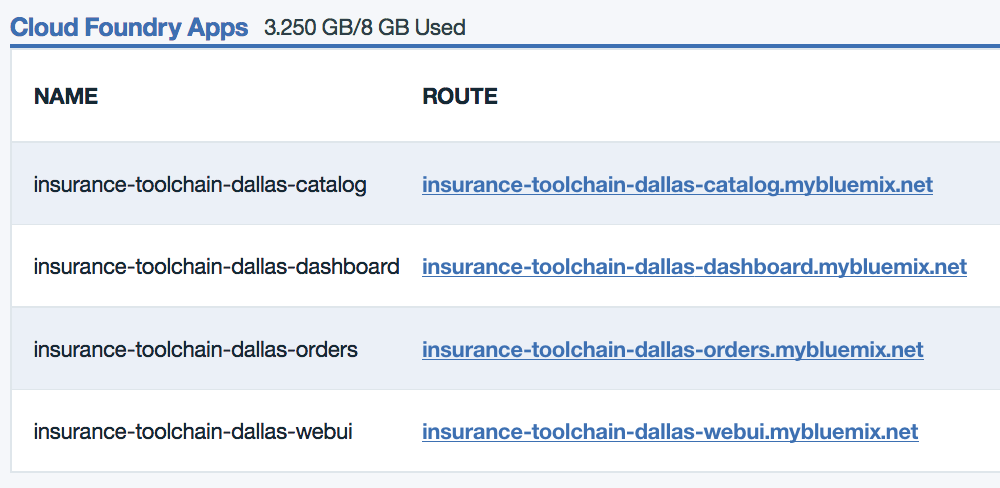
**ouvrir doc via le Catalog**

[**https://watson.stage1.mybluemix.net/watson/developercloud/conversation.html**](https://watson.stage1.mybluemix.net/watson/developercloud/conversation.html)

**CONFIGURATION TERMINE**

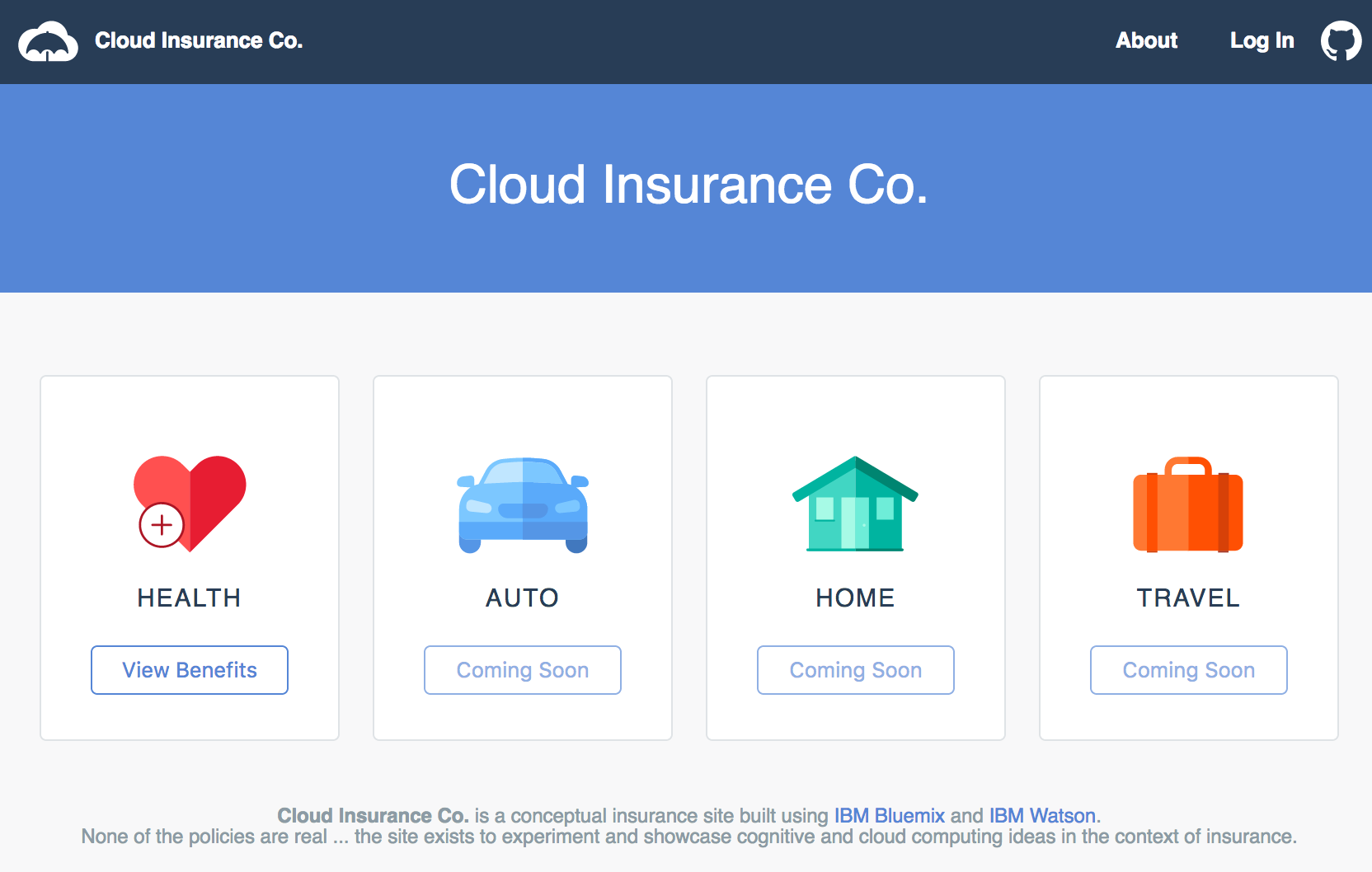
**Nous allons passer à la première démo**

**Lancer l’application suffixé en webui. :**

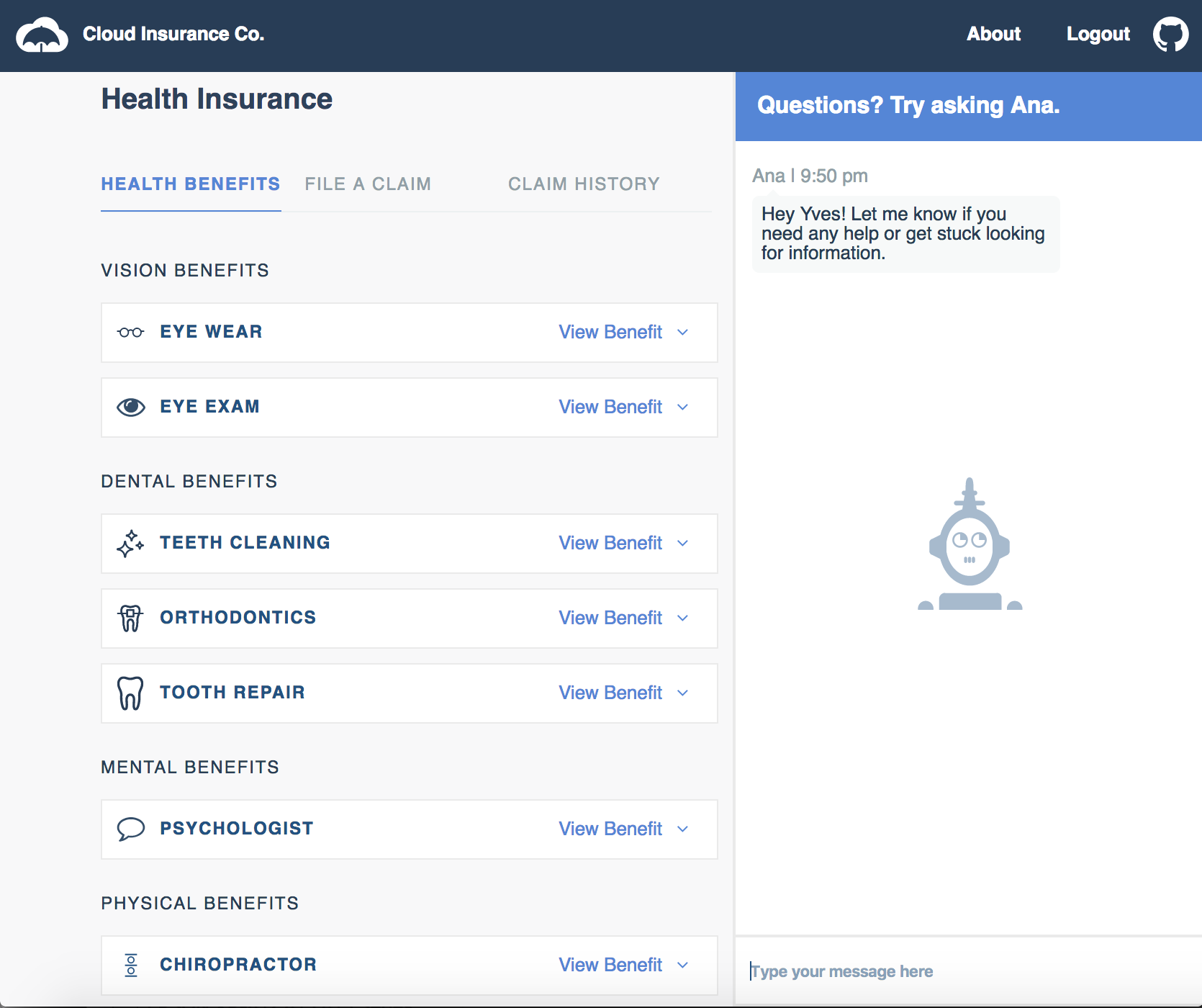
****

**Vous devez arriver sur la page d’accueil de CloudCo.**

**Cliquer sur Log In et cliquer sur Create new account pour vous créer un accès. (self service).**

****

**Une fois créé, vous pouvez enfin vous connecter sur votre compte fictif chez CloudCo.**

****

**Vous pouvez dialoguer avec le ChatBotAna avec la zone de texte en bas à droite.**

**DEMO 1 : LETS CHAT** Ouvrir **AVEC CHROME l’app ChatBot (prod)**

1) Créer un compte e et faire un Log In => Ana devrait dire bonjour.

2) Montrer le contenu de l’application en particulier la couverture Lunettes offrant 300 $ de remboursement annuel (Eyewear), puis suivre le script 1

**SCRIPT1 : Filing a claim can be a painful process**

Ana: Hey Yves! Let me know if you need any help or get stuck looking for information.

You: **I'd like to know my coverage for glasses**

Ana: Your coverage for eye wear is 80% and you have claimed $0 of $300

You: **Can I file a claim now?**

Ana: Let me take care of that paperwork for you. What did you have done?

You: **I bought new glasses**

Ana: Got it. Now what was the date of your appointment?

You: **Yesterday (ou Today si le soir) (attentien au bug de timezone)**

Ana: I have a date of 2016-11-30. Is that right?

You: **yep**

Ana: Fantastic. What is the name of the provider or physician you went to?

You: **Optics Avenue**

Ana: Alright, last thing. What is the amount you want to claim?

You: **$235**

Ana: Your eye wear claim for 235.00 was successfully filed! You have reached max coverage. Remaining $100 of policy limit applied.

You: **thank you!**

Ana: Happy to help!

**Que se passe-t-il si le Chatbot n’a pas été entrainé ?**

=> Logout

**SCRIPT #2:**

Ana: Happy to help!

You**: I bought new spectacles**

Ana: I'm not confident enough to answer that one. Maybe try something else.

**Heureusement, nous pouvons via le dashboard du service Watson Conversation directement modifier/entrainer le ChatBot.**

Go into the Conversation workspace, add "**spectacles**" as synomym for "glasses" :

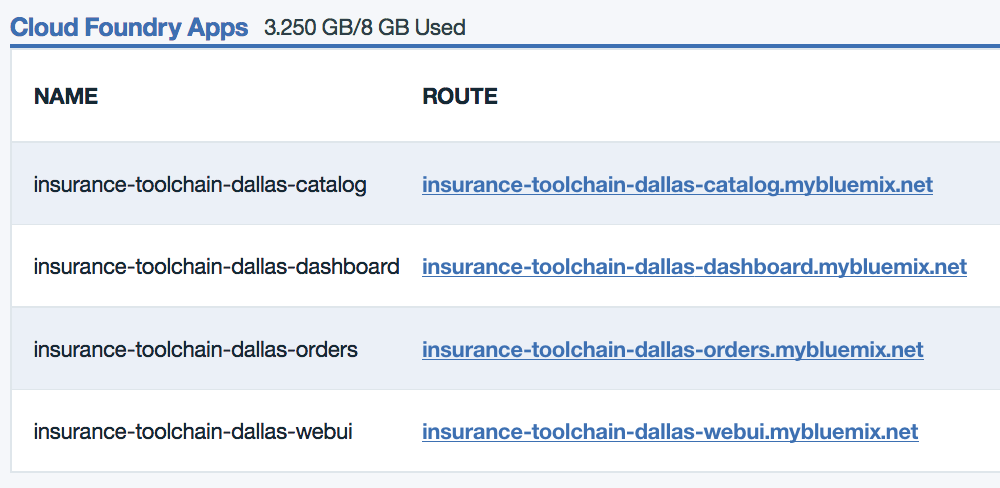
1. **Open the Workspace of the Conversation service instance.**
2. **Clicks on Get Started**
3. **Select the ENTITIES tab**
4. **Click on @procedures**
5. **Click on eye wear and add the “spectacles” synonyms.**
6. **(Waiting some time. (> 1 min) : montrer le modèle**
7. **refaire le SCRIPT #2 après avoir fait un Logout.**

**DASHBOARD avec Tone Analyser**

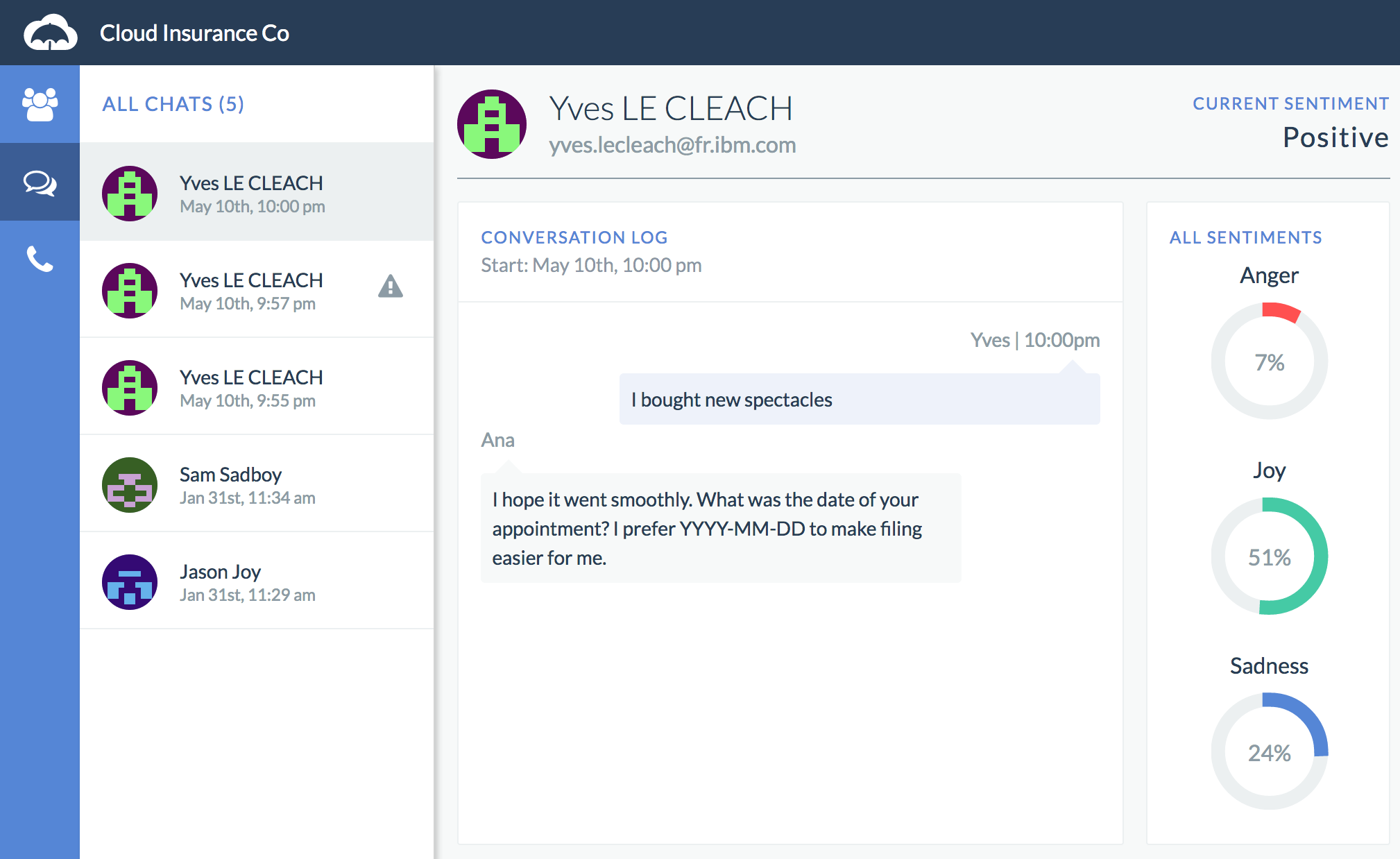
**SLIDE 8 :** maintenant que nous avons vu comment configurer et modifier l’instance de service Watson Conversation, comment à l’échelle d’une entreprise fait-on pour améliorer le ChatBot ? Les Business Users de CloudCo disposent d’un Dashboard qui affiche l’ensemble des conversations, et pour lequelle on va utiliser de l’analyse de sentiment grace au service Watson Tone Analyser disponible sous Bluemix.

**SLIDE 9** : Architecture (complète)

Demo : ouvrir l’application suffixée avec “dashboard”

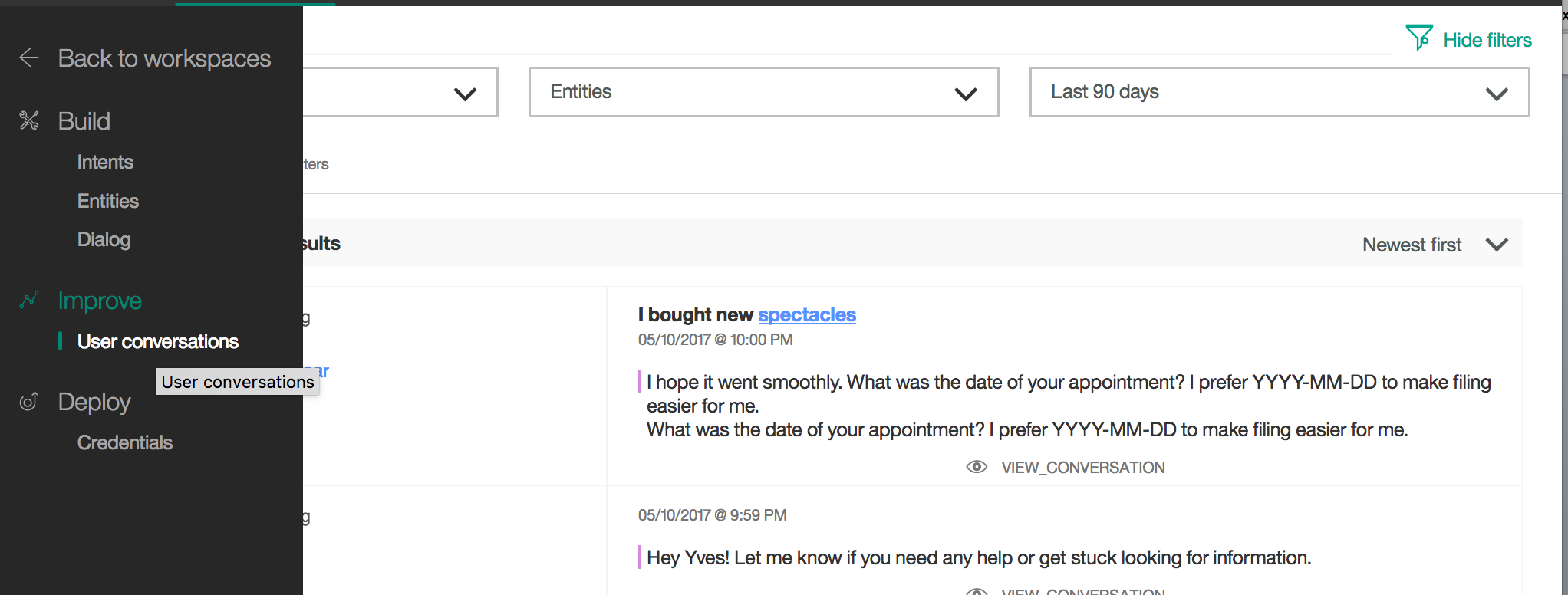
****

Vous devez arriver sur



Expliquer que les conversations sont enregistrer en backend, et analysé avec Watson Tone Analyser. (analyse de sentiment).

Montrer également dans le Dashboard de Watson Conversation la fonction Improve User Conversation, qui sert aux Utilisateurs Métier (spécialiste ChatBot) d’améliorer la pertinence de cette IA.



**SLIDE 11** : what next ? Watson Catalog.

**SLIDE 12** : conclusion. Il est très facile de créer un ChatBot puissant avec Watson Conversation, et d’exploiter les Sentiments de la Conversation. Il est même possible de coupler les deux. (Voir la démo OK Watson sur GitHub Node-RED Watson. Open Tool Chain est le standard open source qui est utilisé pour l’implémentation des services de Continuous Delivery, et permet de créer des templates d’initialisation de projets cloud pour vos équipes projets. (1 click et une personne peut démarrer à travailler en mode DevOps Continuous Delivery selon les patterns Open Tool Chain mis en place).

**SLIDE 13** : n’hésitez pas à tester par vous même ce code de démo disponible sur GitHub.

**LIEN UTILES**

<https://github.com/IBM-Bluemix/cloudco-insurance>

instructions (IBMer) : <https://github.ibm.com/Bluemix/developer-productivity/blob/master/demos/cloudco-insurance/chatbot/cloudco-chatbot-script.md>

Démo in 3 min et Script pour Ana : <https://github.ibm.com/Bluemix/developer-productivity/blob/master/demos/cloudco-insurance/chatbot/cloudco-chatbot-3min-script.md>

toolchain : <https://github.com/IBM-Bluemix/insurance-toolchain>

Lien DevOps : <https://console.ng.bluemix.net/devops> (à montrer)

DEBUG [**https://dev-demo42-insurance-dashboard.eu-gb.mybluemix.net/chatlogs**](https://dev-demo42-insurance-dashboard.eu-gb.mybluemix.net/chatlogs)