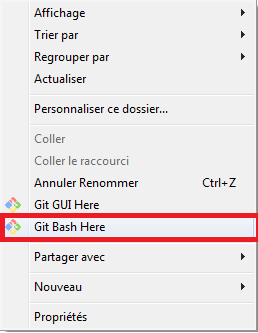
Atelier Git 1

## Exercice 1 :

1. Se connecter au GitHub public avec son compte (login, password) et vérifier que vous avez bien accès au projet « **passion** » :

* <https://github.com/kmahmou/passion>

1. Créer un répertoire sur votre disque local : c:\github\
2. Clic-droit et ouvrir la fenêtre de commande « **git bash for windows** » :



1. Configurer le proxy en copiant/collant la commande suivante :
   1. *git config --global http.proxy* [*http://PRAFAPXYVIPADM.siege.axa-fr.intraxa:8080*](http://PRAFAPXYVIPADM.siege.axa-fr.intraxa:8080)
2. Cloner le repository « passion » dans ce même répertoire (c:\github\)
   1. Indice : git clone…

## Exercice 2 :

1. Créer une branche qui se nomme « **feature/%VOTRE\_TRIGRAMME%** » basé sur la branche **develop**
   1. Indice : git checkout -b…
2. Modifier un des fichiers du contenu du répertoire c:\github\passion\src\application\source\
3. Ajouter/Commiter/pousser vos modification dans votre branche « **feature/%VOTRE\_TRIGRAMME%** »
   1. Indice : git add ., git commit -m « MyComment », git push,…
4. Vérifier que vous voyez bien vos modifications : **https://github.com/kmahmou/passion/tree/feature/%VOTRE\_TRIGRAMME%**

***Bonus :***

* Créer une branche dans le git à partir de la branche master
  + Exemple : « myfeat »
  + Indice : git branch, git checkout…
* Faire une modification sur un des template JSON
  + Modifier dans le fichier « **webapp-parameters.json** » la variable « WEBAPP\_NAME » en « WEBAPP\_NAME1 »
* Merger la branche « myfeat » avec la branche master
  + Indice : git checkout, git merge, …

## Exercice 2 VSTS (Build) :

1. Aller sur l’onglet « build » et créer une « build definition » - Empty
   1. Vous pouvez appeler la build definition comme vous le souhaitez (ex : « MyFirstBuildDefinition »)
2. Faire pointer votre build sur la branche master du repo github que vous avez créé à l’exercice 1
3. Ajouter les tâches pour pousser le code préalablement poussé dans un artifact (packages)
   1. Indice : Artifacts
4. Modifier les paramètres de la tâche pour pousser votre code dans un artifact. Renseigner :
   1. Copy root : le répertoire d’entrée
   2. Contents : \*\*\\*.\* pour copier tous les fichiers ou appliquer un filtre sur \*. json
   3. Artifact Name : ce que vous voulez, ex : MyFirstArtifact
   4. Artifact Type : Server
5. Sauvegarder, Lancer une build et vérifier que celle-ci est en succès
   1. Agent queue : **AzureWindowsBuild**
   2. Queue…

Pause & Explication exercice 3

## Exercice 3 VSTS (Release) :

1. Dans votre GitHub, pousser les fichiers JSON qui permettent de créer le service plan et la WebApp :



1. Aller sur l’onglet « Release » et créer une « Release Definition » - Empty
   1. Vous pouvez appeler la Release definition comme vous le souhaitez (ex : « MyFirstReleaseDefinition »)
   2. Artifact : Choose Later
2. Rattacher votre build (de l’exercice 2) à votre release :
   1. Indice : Artifacts
3. Ajouter les tâches qui permettront de :
   1. Remplacer les variables qui sont dans « **webapp-parameters.json** » et « **sp-parameters.json** »
   2. Déployer le ServicePlan dans Azure
   3. Déployer la WebApp dans AZURE

**** l’ordre de création des ressources est important

1. Configurer l’agent VSTS qui va permettre de déployer
   1. Deployment queue : Default
   2. Demands :
      1. Ajouter le nom « azure\_env » equals « pp »

** Provisionner vos ServicePlan et WebApp dans la région « West Europe »**