Atelier Build 1

## Exercice 1 :

1. Aller sur l’onglet « build » et importer la build defintion (JSON) qui se trouve dans …/passion\sample\Build.
   1. Renommer la build avec votre %TRIGRAMME%
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 »**