****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **MOP SentinelAsCode** | | |  |
|  | |  |  | |

**MOP SentinelAsCode**

- Version 1 -

|  |  |  |  |
| --- | --- | --- | --- |
| **Auteur(s)** |  |  | **Contributeur(s)** |
| **Thomas COUILLEAUX**  Architecte  thomas.couilleaux@metsys.fr |  |  | **-**  -  - |

|  |
| --- |
|  |
| |  |  |  | | --- | --- | --- | | **METSYS**  121 Rue d'Aguesseau  92100 Boulogne-Billancourt |  | Tél : 0181891970  www.metsys.fr | |
|  |

## Historique des modifications

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Auteur | Version | Description du changement |
| 16/11/2020 | Thomas COUILLEAUX | 1.0 | Version initiale |
|  |  |  |  |
|  |  |  |  |

## Diffusion

La diffusion de ce document devrait être limitée aux administrateurs Metsys.

## Document(s) de référence

|  |  |  |
| --- | --- | --- |
| Référence | Date / Version | Titre du document |
|  |  |  |

## Sommaire

[Historique des modifications 2](#_Toc59029533)

[Diffusion 2](#_Toc59029534)

[Document(s) de référence 2](#_Toc59029535)

[Sommaire 3](#_Toc59029536)

[1 Introduction 4](#_Toc59029537)

[1|1 Objectifs 4](#_Toc59029538)

[1|2 Contenu du document 4](#_Toc59029539)

[1|3 Applicabilité 4](#_Toc59029540)

[1|4 Guide de lecture 4](#_Toc59029541)

[2 Périmètre 5](#_Toc59029542)

[3 Commissionnement 6](#_Toc59029543)

[3|1 Commande d’un nouveau KEYVAULT. 6](#_Toc59029544)

[3|2 Intégration du nouvel KEYVAULT au cluster. 7](#_Toc59029545)

[3|3 Intégration des disques du nouvel KEYVAULT au datastore du cluster. 9](#_Toc59029546)

[4 Décommissionnement 11](#_Toc59029547)

[4|1 Suppression d’un KEYVAULT au sein d’un cluster. 11](#_Toc59029548)

# Introduction

## Objectifs

L’objet de ce document est de décrire le mode opératoire pour configurer un tenant sentinel avec l’outil SentinelAsCode.

## Contenu du document

Outre la présente introduction, le document comporte les chapitres suivants :

* Définition du périmètre
* Mode Opératoire

## Applicabilité

En cas d’impossibilité d’appliquer l’une des fonctions prescrites dans ce document, une demande de dérogation doit être adressée au chef de projet.

La demande de dérogation « Effectuer et traiter une demande de dérogation » doit être adressé au référent, qui en assurera le traitement et le suivi.

## Guide de lecture

Indications dans la marge

Les éléments importants et les remarques sont indiqués par une main dans la marge. (style Remarque)

* Les questions importantes sont indiquées par un point d’interrogation dans la marge.

Codes couleur

Les codes couleur suivants sont utilisés dans ce document :

* Texte surligné jaune = texte ajouté par rapport à la version précédente
* ~~Texte barré~~ = texte supprimé

# Périmètre

À la suite d’une demande d’intégration d’un client, ce mode opératoire couvre la configuration d’un tenant sentinel en utilisant l’outil SentinelAsCode.

# Intégration Script

## Synchroniser le repository SentinelAsCode

-Télécharger et installer git pour windows :

**Exemple d’Intégration :**

https://git-scm.com/download/win

-Lancer une invite de commande powershell et synchroniser le repository en utilisant la branche dédiée au client :

**Exemple d’Intégration :**

#git clone --branch client1 <https://github.com/gaelor/SentinelAsCode.git> SentinelAsCode

## Créer le fichier de onboarding

-Lister les workspace au sein du tenant :

|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **TenantID** | Identifiant du tenant Microsoft Azure. | <String> |
| **SubscriptionID** | Identifiant de la soucription Microsoft Azure. | <String> |

**Exemple d’Intégration :**

.\LocalScripts\GenOnboarding.ps1 -OnboardingFolder Onboard -TenantID <String> -SubscriptionID <String>

Le fichier « *onboarding\_ddMMyyyy.json* » est généré dans le répertoire « *Onboard* » à la racine du répertoire de synchronisation.

-Modifier le fichier pour qu’il contienne les workspaces devant être intégrés, un fichier exemple est disponible dans « *Onboard/onboarding.json.sample* ».

-Renommer le fichier en « *onboarding.json »* une fois les modifications effectuées.

## Activer Sentinel

-Installer sentinel sur les workspaces définis dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\InstallSentinel.ps1 -OnboardingFile Onboard\onboarding.json

## Déployer les connecteurs

-La documentation des connecteurs est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/Connectors/*README.md*

-Supprimer et déployer les connecteurs par défauts sur les workspaces définis dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreateConnectors.ps1 -OnboardingFile Onboard\onboarding.json -DeleteAll

-Ajouter manuellement les autres connecteurs nécessaires (industrialisation à venir) :

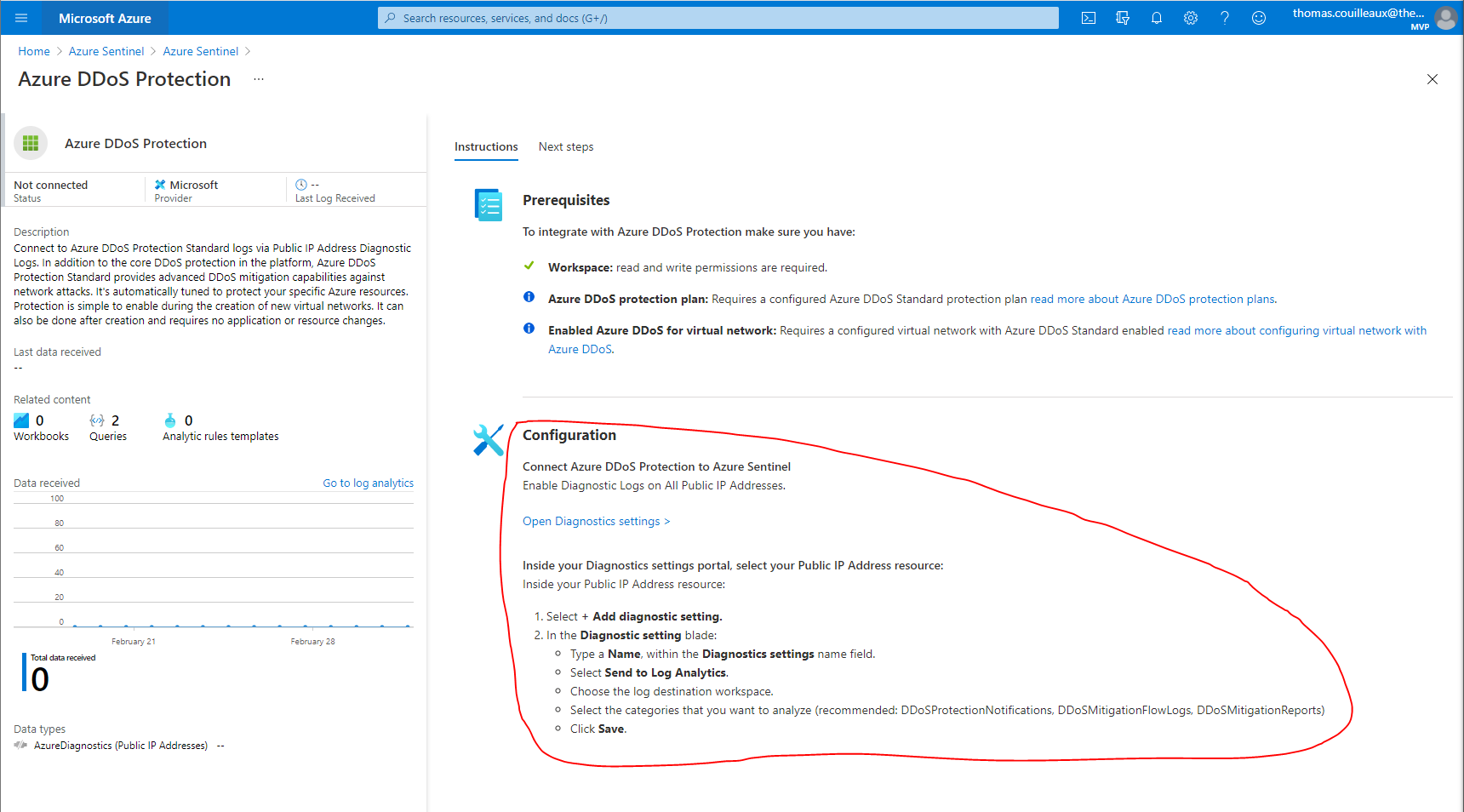
* Se connecter au portail de gestion des workspaces dans Sentinel :

<https://portal.azure.com/#blade/HubsExtension/BrowseResource/resourceType/microsoft.securityinsightsarg%2Fsentinel>

* Sélectionner le workspace, puis sélectionner le connecteur souhaité :



* Suivre les étapes de configuration du connecteur présentées :



## Générer le fichier de analytics rules Microsoft

-Lister toutes les analytics rules fournis par microsoft au sein d’un workspace :

|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **TenantID** | Identifiant du tenant Microsoft Azure. | <String> |
| **SubscriptionID** | Identifiant de la soucription Microsoft Azure. | <String> |
| **Workspace** | Identifiant du workspace Microsoft Azure. | <String> |

**Exemple d’Intégration :**

.\LocalScripts\GenAnalyticsRules.ps1 -AnalyticsRulesFolder AnalyticsRules -TenantID <String> -SubscriptionID <String> -Workspace <String>

Le fichier « *analytics-rules\_ddMMyyyy.json »* est généré dans le répertoire « *AnalyticsRules* » à la racine du répertoire de synchronisation (ddMMyyyy correspond au jour).

-La documentation du fichier généré est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/AnalyticsRules/*analytics-rules\_ddMMyyyy.json.md*(ddMMyyyy correspond au jour)

-Modifier le fichier « *analytics-rules\_ddMMyyyy.json »* pour qu’il contienne les analytics rules devant être intégrés.

-Renommer le fichier en « *analytics-rules.json »* une fois les modifications effectuées.

## Déployer les analytics rules

-La documentation du fichier analytics-rules.json est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/AnalyticsRules/*README.md*

-Déployer les analytics rules définis dans le fichier *analytics-rules.json* sur les workspaces renseigné dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreateAnalyticsRules.ps1 -OnboardingFile Onboard\onboarding.json -RulesFile AnalyticsRules\analytics-rules.json

## Générer le fichier de hunting rules Microsoft

-Récupérer toutes les hunting rules fournis par microsoft sur github :

**Exemple d’Intégration :**

.\LocalScripts\GenMSHuntingRules.ps1 -HuntingRulesFolder HuntingRules

Le fichier « *MS\_hunting-rules\_ddMMyyyy.json »* est généré dans le répertoire « *HuntingRules* » à la racine du répertoire de synchronisation.

-La documentation du fichier généré est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/HuntingRules/*MS\_hunting-rules\_ddMMyyyy.json.md*(ddMMyyyy correspond au jour)

-Modifier le fichier « *MS\_hunting-rules\_ddMMyyyy.json »* pour qu’il contienne les hunting rules devant être intégrés.

-Renommer le fichier en « *hunting-rules.json »* une fois les modifications effectuées.

## Générer le fichier de hunting rules Wortell

-Récupérer toutes les hunting rules fournis par le SOC Wortell sur github :

**Exemple d’Intégration :**

.\LocalScripts\GenWortellHuntingRules.ps1 -HuntingRulesFolder HuntingRules

Le fichier « *Wortell\_hunting-rules\_ddMMyyyy.json »* est généré dans le répertoire « *HuntingRules* » à la racine du répertoire de synchronisation.

-La documentation du fichier généré est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/HuntingRules/*Wortell\_hunting-rules\_ddMMyyyy.json.md*(ddMMyyyy correspond au jour)

-Modifier le fichier « *Wortell\_hunting-rules\_ddMMyyyy.json »* pour qu’il contienne les hunting rules devant être intégrés.

-Concaténer le contenu du fichier au fichier « *hunting-rules.json »* (précédemment créé) une fois les modifications effectuées.

## Générer le fichier de hunting rules RodTrent

-Récupérer toutes les hunting rules fournis par le SOC RodTrent sur github :

**Exemple d’Intégration :**

.\LocalScripts\GenRodTrentHuntingRules.ps1 -HuntingRulesFolder HuntingRules

Le fichier « *RodTrent\_hunting-rules\_ddMMyyyy.json »* est généré dans le répertoire « *HuntingRules* » à la racine du répertoire de synchronisation.

-La documentation du fichier généré est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/HuntingRules/*RodTrent\_hunting-rules\_ddMMyyyy.json.md*(ddMMyyyy correspond au jour)

-Modifier le fichier « *RodTrent\_hunting-rules\_ddMMyyyy.json »* pour qu’il contienne les hunting rules devant être intégrés.

-Concaténer le contenu du fichier au fichier « *hunting-rules.json »* (précédemment créé) une fois les modifications effectuées.

## Générer le fichier de hunting rules BlueTeam

-Récupérer toutes les analytics rules fournis par le SOC BlueTeam sur github :

**Exemple d’Intégration :**

.\LocalScripts\GenBlueTeamHuntingRules.ps1 -HuntingRulesFolder HuntingRules

Le fichier « *BlueTeam\_hunting-rules\_ddMMyyyy.json »* est généré dans le répertoire « *HuntingRules* » à la racine du répertoire de synchronisation.

-La documentation du fichier généré est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/HuntingRules/*BlueTeam\_hunting-rules\_ddMMyyyy.json.md*(ddMMyyyy correspond au jour)

-Modifier le fichier « *BlueTeam\_hunting-rules\_ddMMyyyy.json »* pour qu’il contienne les hunting rules devant être intégrés.

-Concaténer le contenu du fichier au fichier « *hunting-rules.json »* (précédemment créé) une fois les modifications effectuées.

## Déployer les hunting rules

-La documentation du fichier hunting-rules.json est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/HuntingRules/*README.md*

-Déployer les hunting rules définis dans le fichier *hunting-rules.json* sur les workspaces renseigné dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreateHuntingRulesAPI.ps1 -OnboardingFile Onboard\onboarding.json -RulesFile HuntingRules\hunting-rules.json

## Déployer les workbooks

-La documentation des workbooks qui seront déployés est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/Workbooks/*Readme.md*

-Déployer les workbooks présents dans le répertoire « Workbooks » sur les workspaces renseigné dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreateWorkbooks.ps1 -OnboardingFile Onboard\onboarding.json -WorkbooksFolder Workbooks

## Déployer les playbooks

-La documentation des playbooks qui seront déployés est disponible à cette adresse :

https://github.com/gaelor/SentinelAsCode/blob/master/Playbooks/*Readme.md*

### Déploiement des playbooks MDATP(sauf Restrict-MDATPIPAddress)

-Editer le fichier *« Playbooks\Playbooks.params »* pour qu’il contienne ces informations :

* <azure\_serviceaccount> : Le nom du compte de service au sein de Azure AD qui exécutera les playbooks.

**Exemple d’Intégration :**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"Azure\_ServiceAccount": {

"value": "<azure\_serviceaccount>"

}

},

"variables": {

}

}

-Déployer les playbooks se nommant *« \*MDATP\*.json »* sur les workspaces renseignés dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreatePlaybooks.ps1 -OnboardingFile .\Onboard\onboarding.json -PlaybooksFolder ".\Playbooks\\*MDATP\*.json" -PlaybooksParamsFile .\Playbooks\Playbooks.params

### Déploiement du playbook Restrict-MDATPIPAddress

-Editer le fichier *« Playbooks\Playbooks.params »* pour qu’il contienne ces informations :

* <azure\_serviceaccount> : Le nom du compte de service au sein de Azure AD qui exécutera les playbooks.
* <integrationAccount> : Le nom du compte d’intégration au sein de Azure AD qui exécutera les playbooks.
* <integrationAccount\_resourceGroup> : Le nom du ressource groupe contenant le compte d’intégration au sein de Azure AD qui exécutera les playbooks.

**Exemple d’Intégration :**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"Azure\_ServiceAccount": {

"value": "<azure\_serviceaccount>"

},

"integrationAccount": {

"value": "<integrationAccount>"

},

"integrationAccount\_resourceGroup": {

"value": "<integrationAccount\_resourceGroup>"

}

},

"variables": {

}

}

-Déployer le playbook *Restrict-MDATPIPAddress* sur les workspaces renseignés dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreatePlaybooks.ps1 -OnboardingFile .\Onboard\onboarding.json -PlaybooksFolder ".\Playbooks\Restrict-MDATPIPAddress.json" -PlaybooksParamsFile .\Playbooks\Playbooks.params

### Déploiement des playbooks AAD

-Editer le fichier *« Playbooks\Playbooks.params »* pour qu’il contienne ces informations :

* <azure\_serviceaccount> : Le nom du compte de service au sein de Azure AD qui exécutera les playbooks.

**Exemple d’Intégration :**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"Azure\_ServiceAccount": {

"value": "<azure\_serviceaccount>"

}

},

"variables": {

}

}

-Déployer les playbooks se nommant *« \*AAD\*.json »* sur les workspaces renseignés dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreatePlaybooks.ps1 -OnboardingFile .\Onboard\onboarding.json -PlaybooksFolder ".\Playbooks\\*AAD\*.json" -PlaybooksParamsFile .\Playbooks\Playbooks.params

### Déploiement des playbooks VirusTotal

-Editer le fichier *« Playbooks\Playbooks.params »* pour qu’il contienne ces informations :

* <azure\_serviceaccount> : Le nom du compte de service au sein de Azure AD qui exécutera les playbooks.
* <virustotal\_key> : La clef d’accès à l’API VirusTotal.

**Exemple d’Intégration :**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"Azure\_ServiceAccount": {

"value": "<azure\_serviceaccount>"

} ,

"Virustotal\_Key": {

"value": "<virustotal\_key>"

}

},

"variables": {

}

}

-Déployer les playbooks se nommant *« \*Reputation\*.json »* sur les workspaces renseignés dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreatePlaybooks.ps1 -OnboardingFile .\Onboard\onboarding.json -PlaybooksFolder ".\Playbooks\\*Reputation\*.json" -PlaybooksParamsFile .\Playbooks\Playbooks.params

### Déploiement du playbook Isolate-NSGMachine

-Editer le fichier *« Playbooks\Playbooks.params »* pour qu’il contienne ces informations :

* <azure\_serviceaccount> : Le nom du compte de service au sein de Azure AD qui exécutera les playbooks.
* <EmailApprovalContact> : L’adresse email utilisé pour approuver ou non l’isolement du système.

**Exemple d’Intégration :**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"Azure\_ServiceAccount": {

"value": "<azure\_serviceaccount>"

},

"EmailApprovalContact": {

"value": "<user>@<domain>"

}

},

"variables": {

}

}

-Déployer le playbook *Isolate-NSGMachine* sur les workspaces renseignés dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreatePlaybooks.ps1 -OnboardingFile .\Onboard\onboarding.json -PlaybooksFolder ".\Playbooks\Isolate-NSGMachine.json" -PlaybooksParamsFile .\Playbooks\Playbooks.params

### Déploiement du playbook Prompt-User

-Editer le fichier *« Playbooks\Playbooks.params »* pour qu’il contienne ces informations :

* <azure\_serviceaccount> : Le nom du compte de service au sein de Azure AD qui exécutera les playbooks.

**Exemple d’Intégration :**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"Azure\_ServiceAccount": {

"value": "<azure\_serviceaccount>"

}

},

"variables": {

}

}

-Déployer le playbook *Prompt-User* sur les workspaces renseignés dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreatePlaybooks.ps1 -OnboardingFile .\Onboard\onboarding.json -PlaybooksFolder ".\Playbooks\Prompt-User.json" -PlaybooksParamsFile .\Playbooks\Playbooks.params

### Déploiement du playbook Restrict-CAIPAddress

-Editer le fichier *« Playbooks\Playbooks.params »* pour qu’il contienne ces informations :

* <azure\_serviceaccount> : Le nom du compte de service au sein de Azure AD qui exécutera les playbooks.
* <integrationAccount> : Le nom du compte d’intégration au sein de Azure AD qui exécutera les playbooks.
* <integrationAccount\_resourceGroup> : Le nom du ressource groupe contenant le compte d’intégration au sein de Azure AD qui exécutera les playbooks.
* <ca\_locationname> : Le nom de la location au sein des accès conditionnels utilisé pour bloquer les ip’s.

**Exemple d’Intégration :**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"Azure\_ServiceAccount": {

"value": "<azure\_serviceaccount>"

},

"integrationAccount": {

"value": "<integrationAccount>"

},

"integrationAccount\_resourceGroup": {

"value": "<integrationAccount\_resourceGroup>"

},

"CA\_LocationName": {

"value": "<ca\_locationname>"

}

},

"variables": {

}

}

-Déployer le playbook « *Restrict-CAIPAddress »* sur les workspaces renseignés dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreatePlaybooks.ps1 -OnboardingFile .\Onboard\onboarding.json -PlaybooksFolder ".\Playbooks\Restrict-CAIPAddress.json" -PlaybooksParamsFile .\Playbooks\Playbooks.params

### Déploiement du playbook Restrict-NSGIPAddress

-Editer le fichier *« Playbooks\Playbooks.params »* pour qu’il contienne ces informations :

* <azure\_serviceaccount> : Le nom du compte de service au sein de Azure AD qui exécutera les playbooks.
* <integrationAccount> : Le nom du compte d’intégration au sein de Azure AD qui exécutera les playbooks.
* <integrationAccount\_resourceGroup> : Le nom du ressource groupe contenant le compte d’intégration au sein de Azure AD qui exécutera les playbooks.
* <nsg\_group> : Le nom du « network security group » utilisé.
* <nsg\_resourcegroup> : Le nom de la ressource de groupe comprenant le « network security group » utilisé.
* <tri\_client> : Trigramme du client (permet de nommer en fonction).

**Exemple d’Intégration :**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"Azure\_ServiceAccount": {

"value": "<azure\_serviceaccount>"

},

"integrationAccount": {

"value": "<integrationAccount>"

},

"integrationAccount\_resourceGroup": {

"value": "<integrationAccount\_resourceGroup>"

},

"NSG\_Group": {

"value": "<nsg\_group>"

},

"NSG\_ResourceGroup": {

"value": "<nsg\_resourcegroup>"

},

"TriClient": {

"value": "<triclient>"

}

},

"variables": {

}

}

-Déployer le playbook « *Restrict-NSGIPAddress »* sur les workspaces renseignés dans le fichier de onboarding :

**Exemple d’Intégration :**

.\LocalScripts\CreatePlaybooks.ps1 -OnboardingFile .\Onboard\onboarding.json -PlaybooksFolder ".\Playbooks\Restrict-NSGIPAddress.json" -PlaybooksParamsFile .\Playbooks\Playbooks.params

### Déploiement du playbook Open-jira-Ticket

-Editer le fichier *« Playbooks\Playbooks.params »* pour qu’il contienne ces informations :

* <azure\_serviceaccount> : Le nom du compte de service au sein de Azure AD qui exécutera les playbooks.
* <integrationAccount> : Le nom du compte d’intégration au sein de Azure AD qui exécutera les playbooks.
* <integrationAccount\_resourceGroup> : Le nom du ressource groupe contenant le compte d’intégration au sein de Azure AD qui exécutera les playbooks.
* <jira\_url> : L’URL de l’instance jira utilisé.
* <jira\_user> : Le compte de service au sein de l’instance jira utilisé.
* <jira\_pwd> : Le mot de passe du compte de service au sein de l’instance jira utilisé.

**Exemple d’Intégration :**

{

"$schema": "https://schema.management.azure.com/schemas/2015-01-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"Azure\_ServiceAccount": {

"value": "<azure\_serviceaccount>"

},

"integrationAccount": {

"value": "<integrationAccount>"

},

"integrationAccount\_resourceGroup": {

"value": "<integrationAccount\_resourceGroup>"

},

"Jira\_URL": {

"value": "<jira\_url>"

},

"Jira\_Organization": {

"value": "<jira\_organization>"

},

"Jira\_User": {

"value": "<jira\_user>"

},

"Jira\_Pwd": {

"value": "<jira\_pwd>"

}

},

"variables": {

}

}

-Déployer le playbook « *Open-jira-Ticket »* sur les workspaces renseignés dans le fichier de onboarding :

**Exemple d’Intégration :**

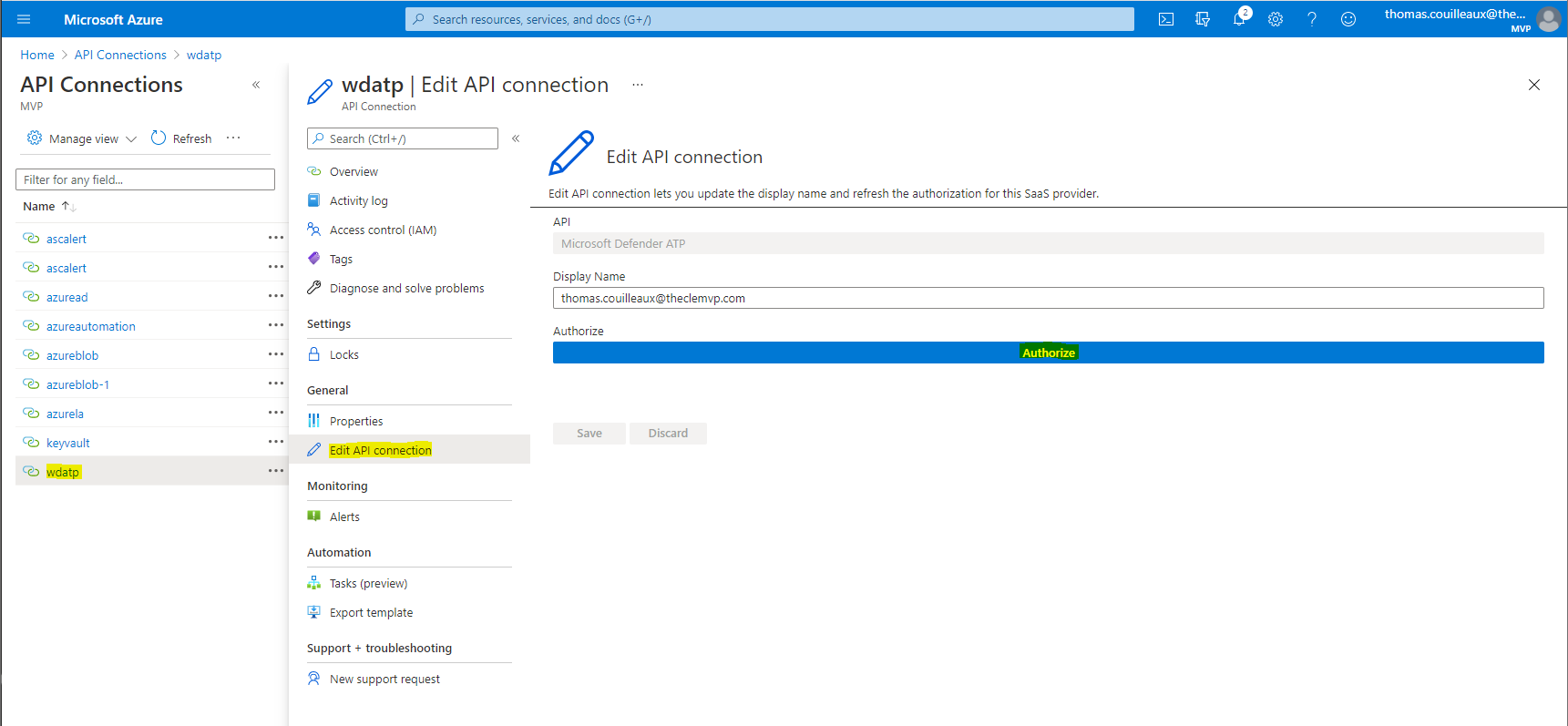
.\LocalScripts\CreatePlaybooks.ps1 -OnboardingFile .\Onboard\onboarding.json -PlaybooksFolder ".\Playbooks\Open-jira-Ticket.json" -PlaybooksParamsFile .\Playbooks\Playbooks.params

### Déploiement des autorisations des playbooks

-Se rendre sur le portail de gestion des connexions aux API et autoriser les connections créées pour les playbooks :

<https://portal.azure.com/#blade/HubsExtension/BrowseResource/resourceType/Microsoft.Web%2Fconnections>

* azuresentinel : utilisé pour la connection à l’API Sentinel.
* wdatp : utilisé pour la connection à l’API Defender.
* office365 : utilisé pour la connection à l’API des ressources Office 365.
* office365users : utilisé pour la connection à l’API des utilisateurs Office 365.
* azuread : utilisé pour la connection à l’API Azure Active Directory.



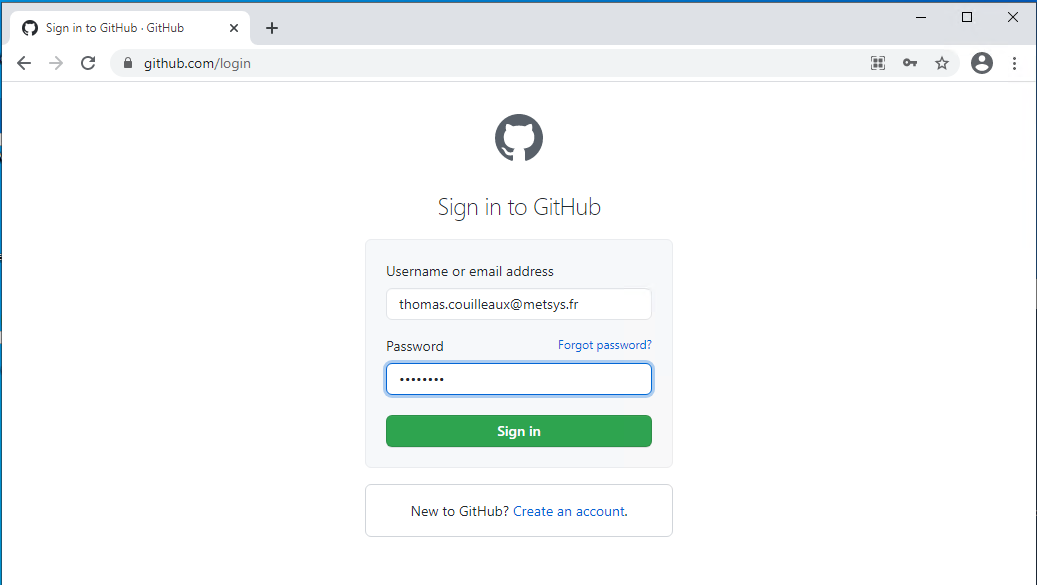
# Intégration Industrialisée

## Synchroniser le repository SentinelAsCode

-Avoir un compte ayant les droits d’exécuter des workflows sur github :

**Exemple d’Intégration :**

https://github.com/login



-Télécharger et installer git pour windows :

**Exemple d’Intégration :**

https://git-scm.com/download/win

-Télécharger et installer chrome pour windows :

**Exemple d’Intégration :**

https://www.google.com/intl/en/chrome/

-Lancer une invite de commande powershell et synchroniser le repository en utilisant la branche dédiée au client :

**Exemple d’Intégration :**

#git clone –branch client1 <https://github.com/gaelor/SentinelAsCode.git> SentinelAsCode

|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **ClientID** | Identifiant du client. | client1 |

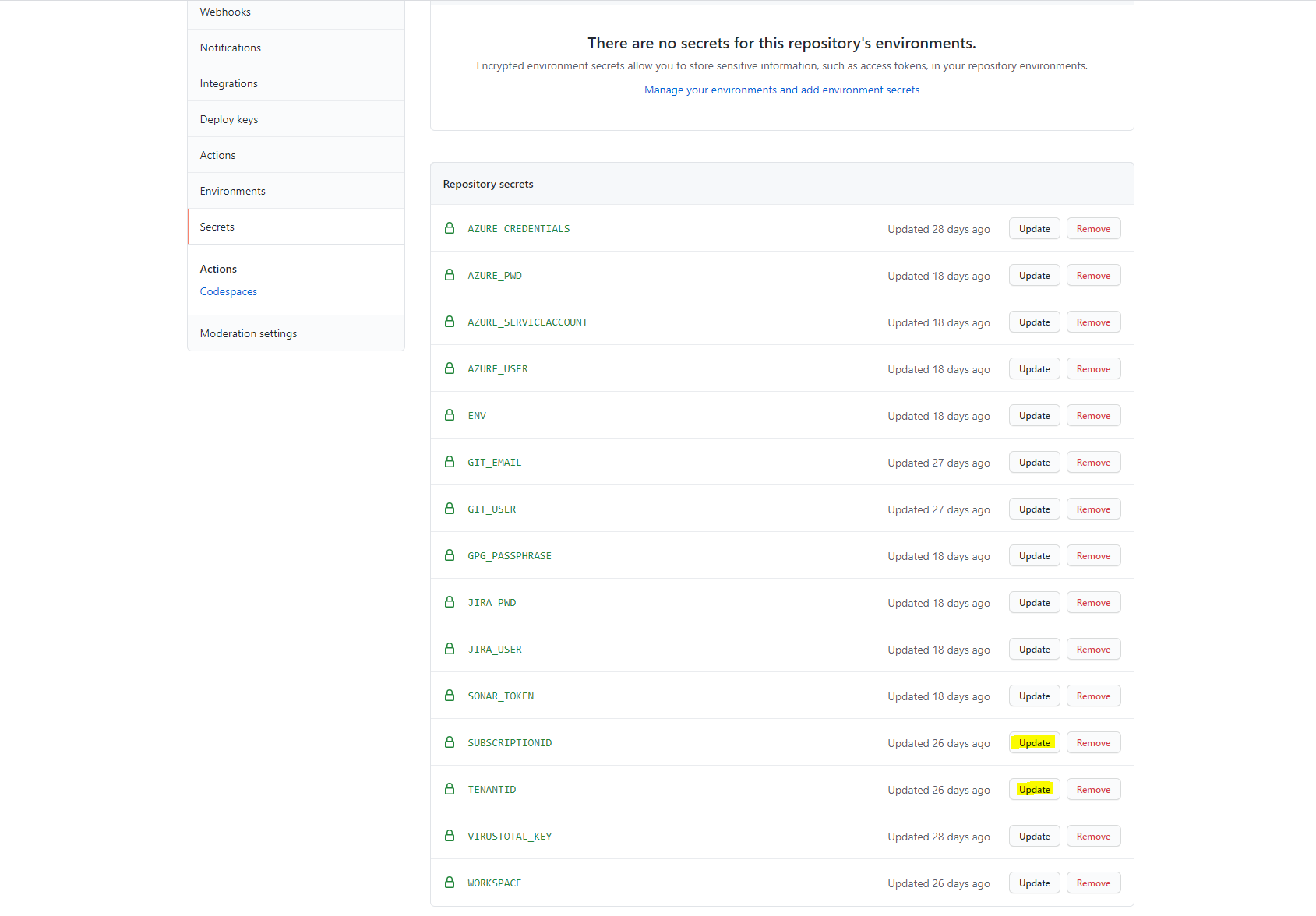
## Créer le fichier de onboarding

-Configurer le tenant et la souscription à utiliser en renseignant les différent id’s :

|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **TenantID** | Identifiant du tenant Microsoft Azure. | <String> |
| **SubscriptionID** | Identifiant de la soucription Microsoft Azure. | <String> |

**Exemple d’Intégration :**

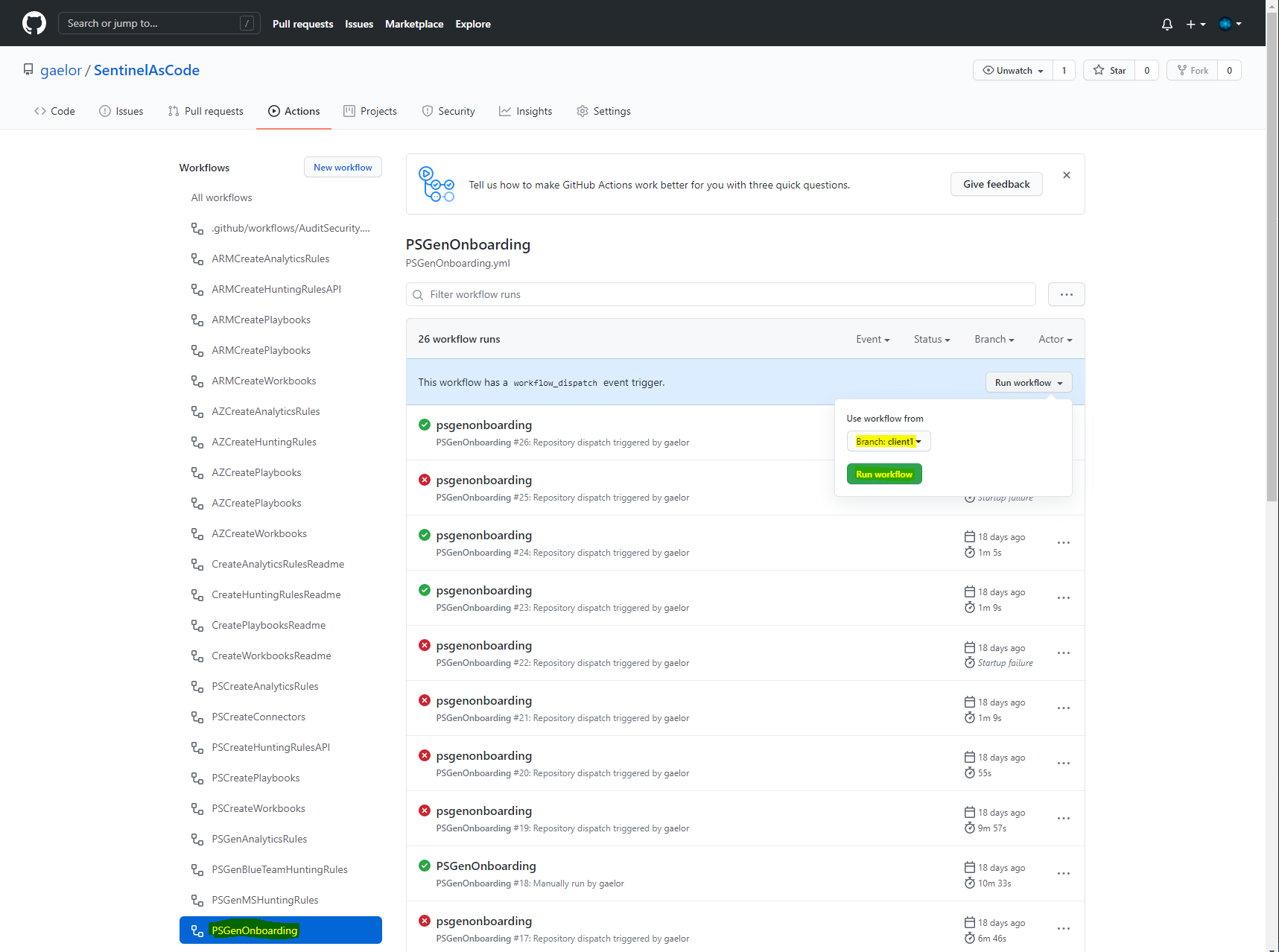
https://github.com/gaelor/SentinelAsCode/settings/secrets/actions



-Lister les workspace au sein du tenant (utiliser la branche dédiée au client) :

**Exemple d’Intégration :**

https://github.com/gaelor/SentinelAsCode/actions/workflows/PSGenOnboarding.yml



-Le fichier onboarding\_ddMMyyyy est généré dans le répertoire « Onboard » (celui-ci est chiffré par gpg).

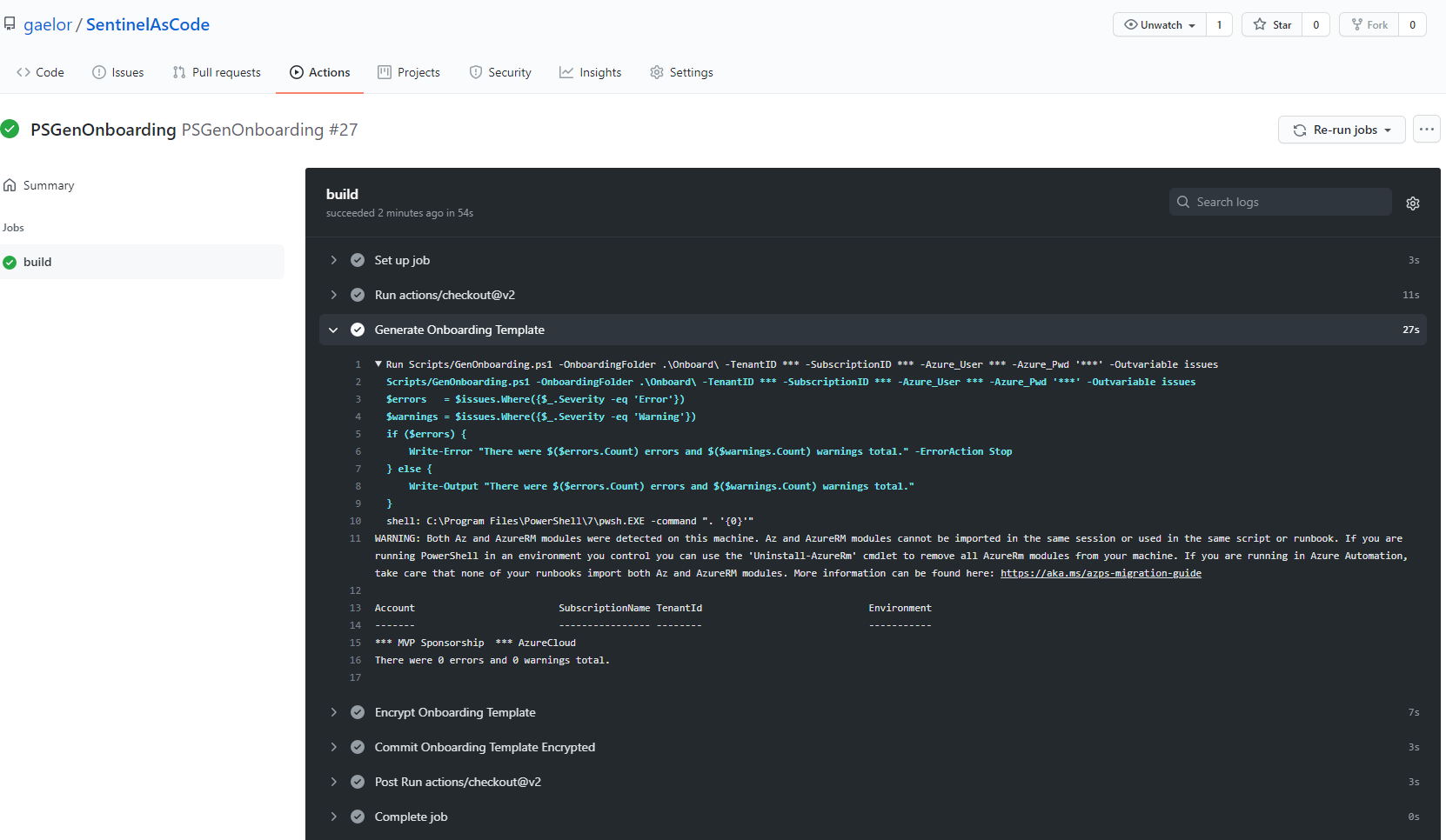
**Exemple d’Intégration :**

https://github.com/gaelor/SentinelAsCode/Onboard/onboarding\_08032021.json.gpg

-L’on peut suivre l’évolution de la tâche en la sélectionnant :

**Exemple d’Intégration :**

https://github.com/gaelor/SentinelAsCode/runs/2066832742?check\_suite\_focus=true



## Adapter le fichier de onboarding

-Télécharger et installer GPG pour windows :

**Exemple d’Intégration :**

https://gnupg.org/download/

-Déchiffrer le fichier généré :

|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **passphrase** | GPG passphrase pour déchiffrer le contenu (voir keypass). | <String> |

**Exemple d’Intégration :**

Get-ChildItem .\Onboard\\*.gpg | ForEach-Object {gpg --quiet --batch --yes --decrypt --passphrase='<passphrase>' --output $\_.FullName.replace('.gpg','') $\_.FullName}

-Adapter le fichier pour contenir uniquement les workspaces à intégrer dans sentinel.

-Chiffrer le fichier modifié :

|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **passphrase** | GPG passphrase pour déchiffrer le contenu (voir keypass). | <String> |

**Exemple d’Intégration :**

gpg --quiet --batch --yes --passphrase='<passphrase>' –symmetric .\Onboard\onboarding.json

1. Adapter le fichier des paramètres pour les playbooks et chiffrer le fichier.

L’ensemble des définitions des paramètres sont disponibles à cette adresse :

<https://github.com/gaelor/SentinelAsCode/tree/master/Playbooks>

-Chiffrer le fichier de paramètre :

|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **passphrase** | GPG passphrase pour déchiffrer le contenu (voir keypass). | <String> |

**Exemple d’Intégration :**

gpg --quiet --batch --yes --passphrase='<passphrase>' --symmetric .\Playbooks\Playbooks.params

1. Configurer les paramètres sensibles au sein du keyvault de github à cette adresse :

<https://github.com/gaelor/SentinelAsCode/settings/secrets/actions>

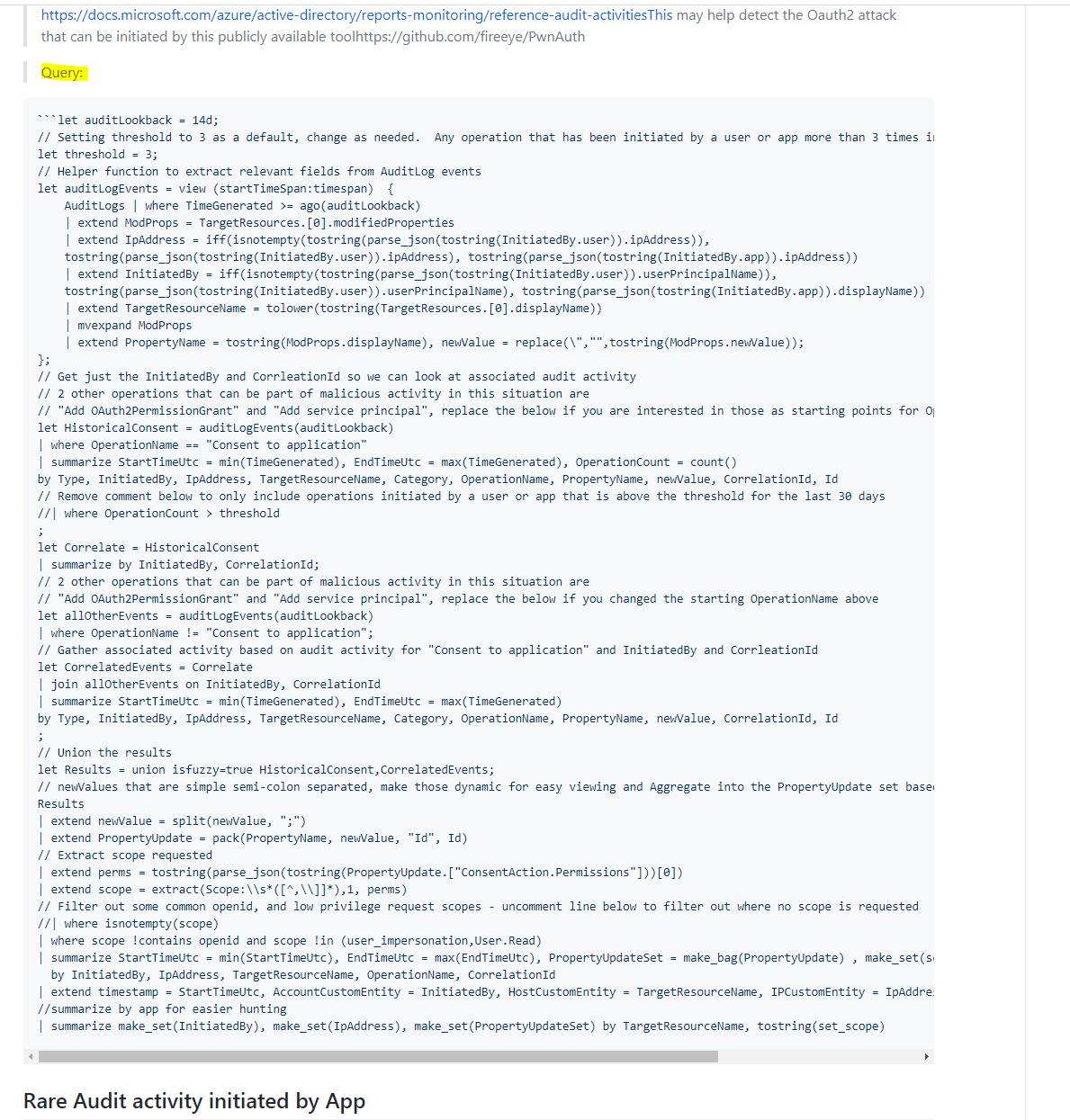
# Intégration Manuelle

## Déploiement des hunting rules

* Se rendre à cette adresse :

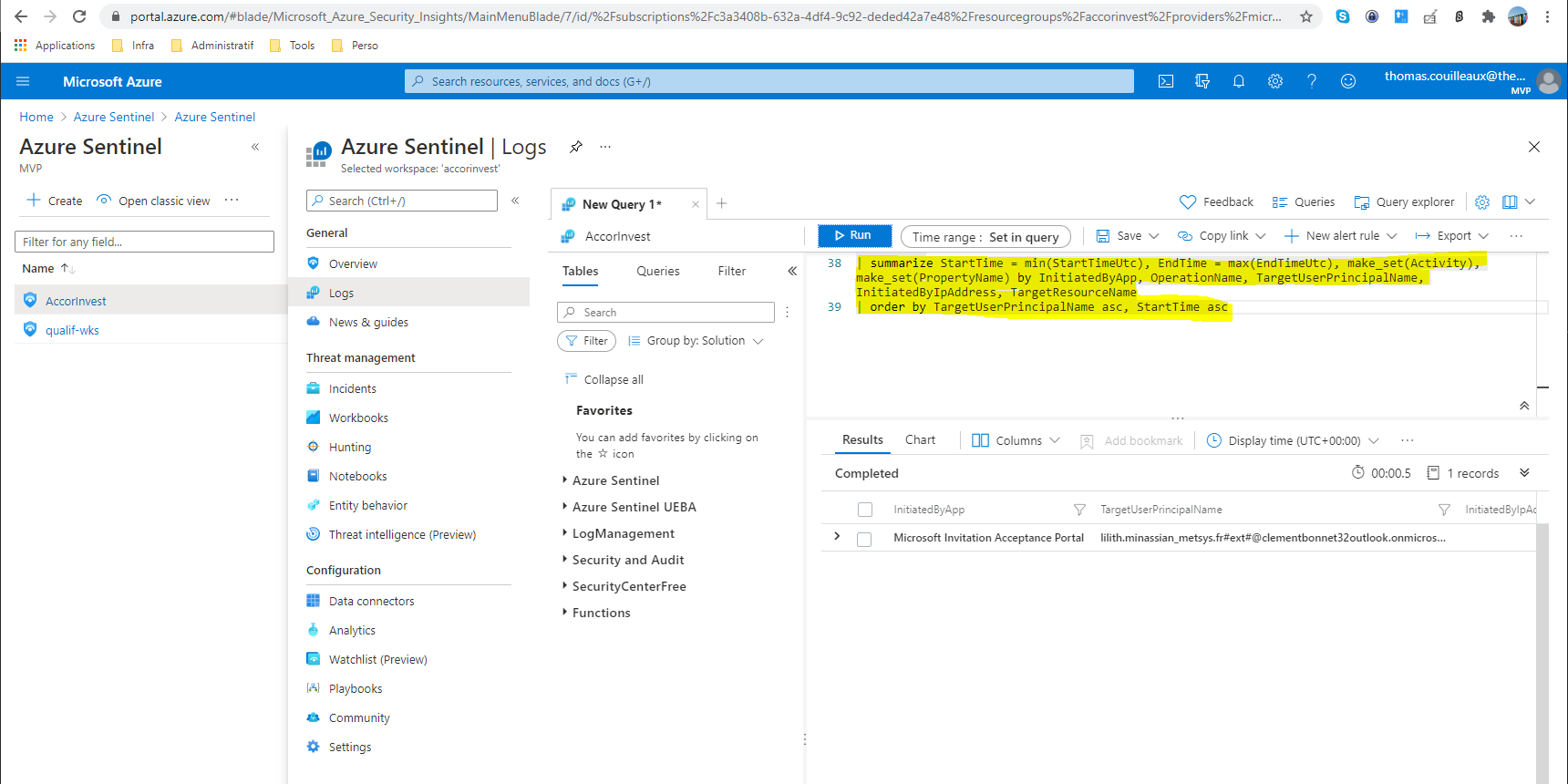
<https://github.com/gaelor/SentinelAsCode/tree/master/HuntingRules>

* Choisir la hunting rules à utiliser et copier la requête dans le presse-papier :



* Se rendre à cette adresse et coller la requête :

https://portal.azure.com/#blade/HubsExtension/BrowseResource/resourceType/microsoft.securityinsightsarg%2Fsentinel

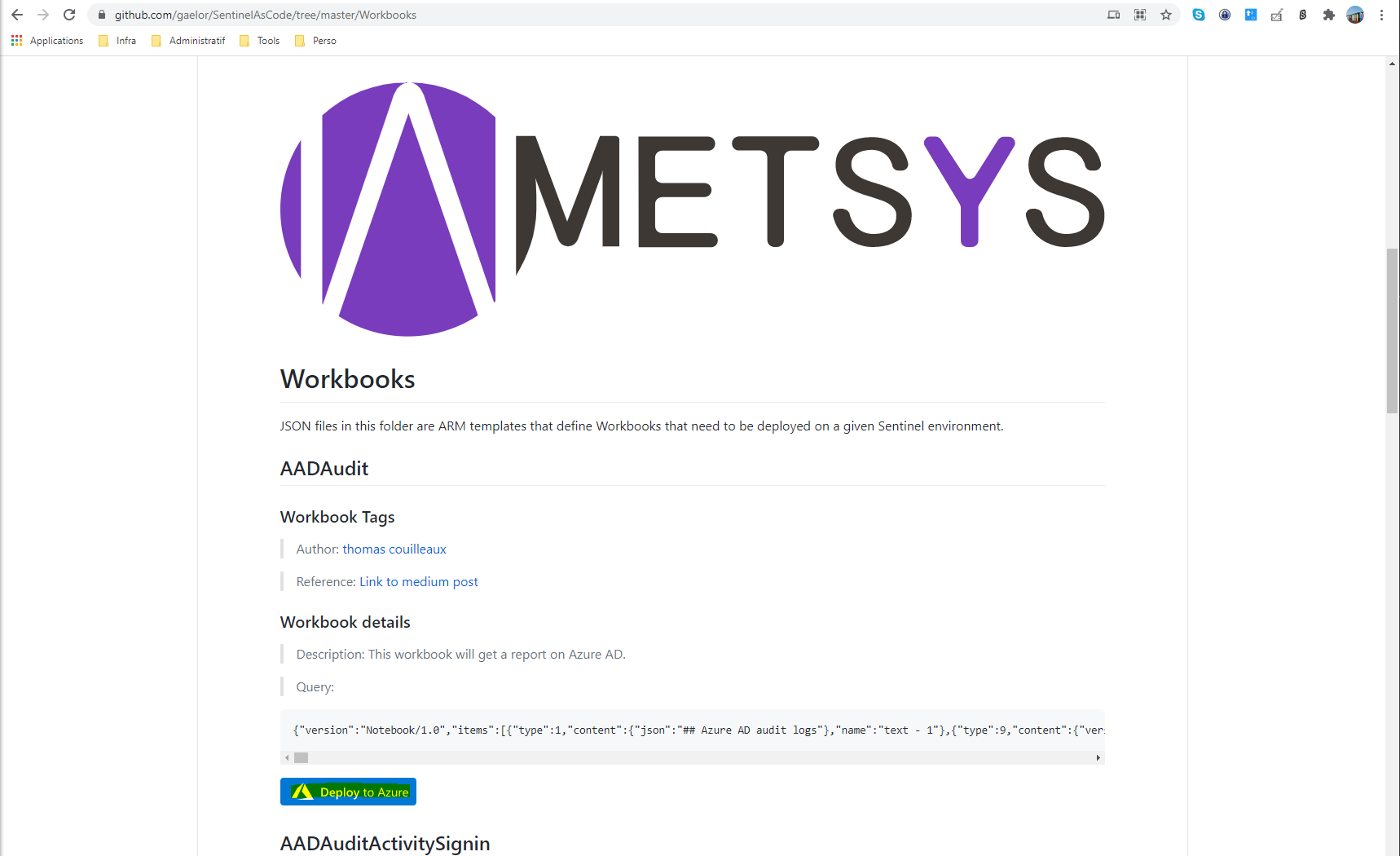


## Déploiement des workbooks

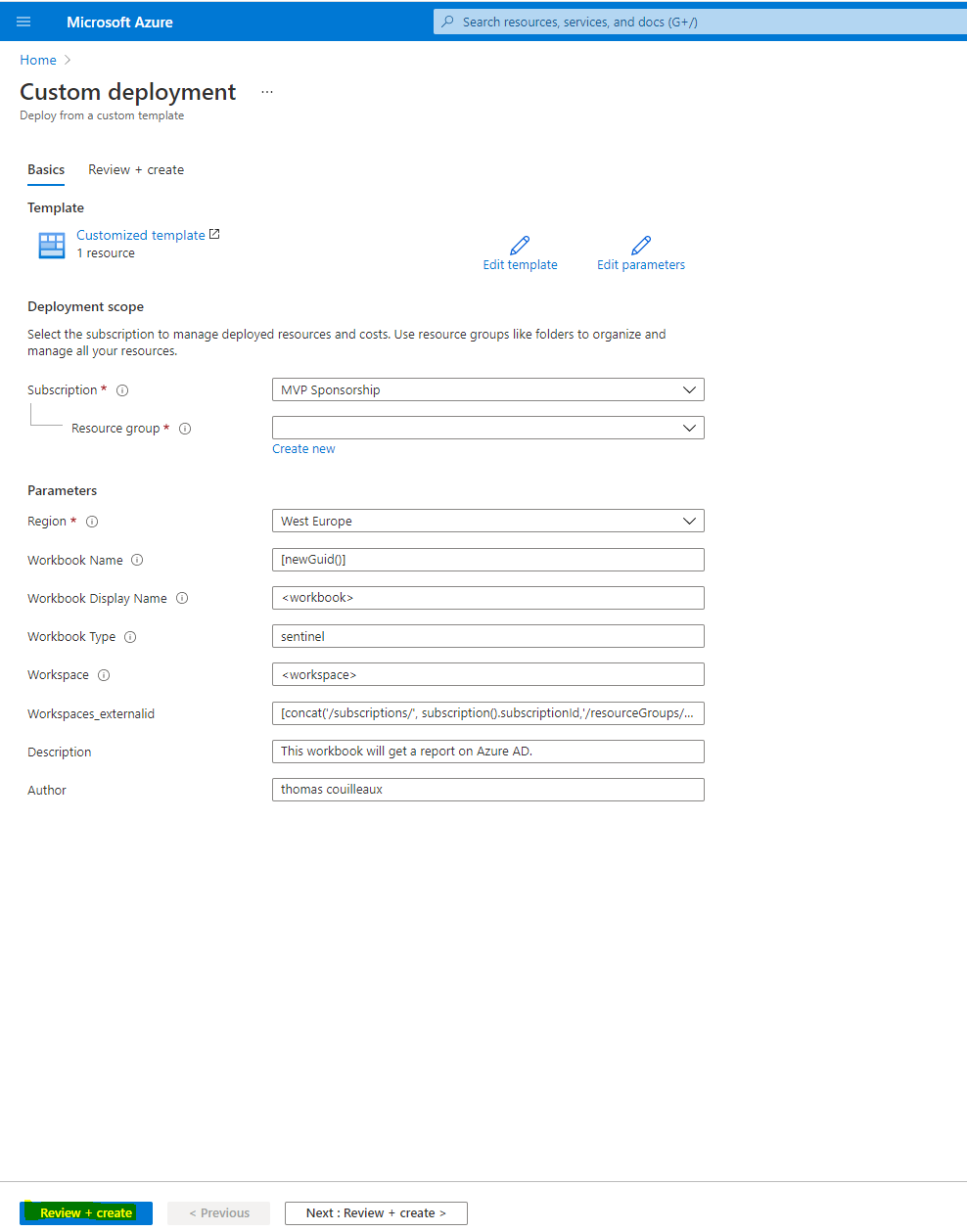
* Se rendre à cette adresse :

<https://github.com/gaelor/SentinelAsCode/tree/master/Workbooks>

* Choisir le workbook à déployer et cliquer sur « Deploy to Azure » :



* Configurer les paramètres et déployer :

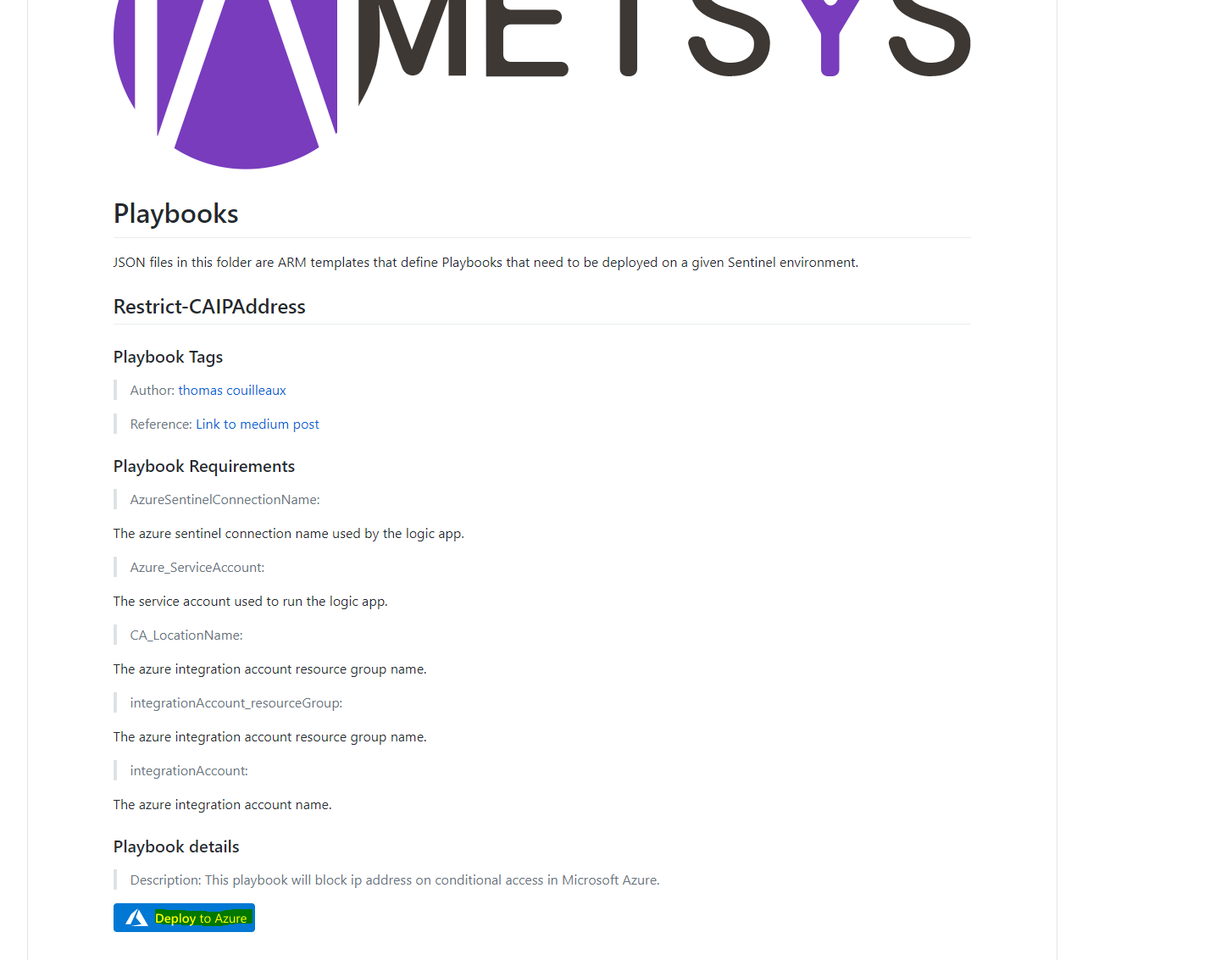


## Déploiement des playbooks

* Se rendre à cette adresse :

<https://github.com/gaelor/SentinelAsCode/tree/master/Playbooks>

* Choisir le playbook à déployer et cliquer sur « Deploy to Azure » :



* Configurer les paramètres et déployer :

