

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **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 intégrateurs Metsys.

## Document(s) de référence

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

## Sommaire

[Historique des modifications 2](#_Toc66266831)

[Diffusion 2](#_Toc66266832)

[Document(s) de référence 2](#_Toc66266833)

[Sommaire 3](#_Toc66266834)

[1 Introduction 5](#_Toc66266835)

[1|1 Objectifs 5](#_Toc66266836)

[1|2 Contenu du document 5](#_Toc66266837)

[1|3 Applicabilité 5](#_Toc66266838)

[1|4 Guide de lecture 5](#_Toc66266839)

[2 Périmètre 6](#_Toc66266840)

[3 Intégration Script 7](#_Toc66266841)

[3|1 Synchroniser le repository SentinelAsCode 7](#_Toc66266842)

[3|2 Créer le fichier de onboarding 7](#_Toc66266843)

[3|3 Activer Sentinel 7](#_Toc66266844)

[3|4 Déployer les connecteurs 8](#_Toc66266845)

[3|5 Générer le fichier de analytics rules Microsoft 9](#_Toc66266846)

[3|6 Déployer les analytics rules 10](#_Toc66266847)

[3|7 Générer le fichier de hunting rules Microsoft 10](#_Toc66266848)

[3|8 Générer le fichier de hunting rules Wortell 10](#_Toc66266849)

[3|9 Générer le fichier de hunting rules RodTrent 11](#_Toc66266850)

[3|10 Générer le fichier de hunting rules BlueTeam 11](#_Toc66266851)

[3|11 Déployer les hunting rules 12](#_Toc66266852)

[3|12 Déployer les workbooks 12](#_Toc66266853)

[3|13 Déployer les playbooks 12](#_Toc66266854)

[4 Intégration Industrialisée 21](#_Toc66266855)

[4|1 Synchroniser le repository SentinelAsCode 21](#_Toc66266856)

[4|2 Créer le fichier de onboarding 22](#_Toc66266857)

[4|3 Adapter le fichier de onboarding 24](#_Toc66266858)

[4|4 Activer Sentinel 26](#_Toc66266859)

[4|5 Déployer les connecteurs 27](#_Toc66266860)

[4|6 Générer le fichier de analytics rules Microsoft 30](#_Toc66266861)

[4|7 Déployer les analytics rules 32](#_Toc66266862)

[4|8 Générer le fichier de hunting rules Microsoft 33](#_Toc66266863)

[4|9 Générer le fichier de hunting rules Wortell 35](#_Toc66266864)

[4|10 Générer le fichier de hunting rules RodTrent 36](#_Toc66266865)

[4|11 Générer le fichier de hunting rules BlueTeam 38](#_Toc66266866)

[4|12 Déployer les hunting rules 39](#_Toc66266867)

[4|13 Déployer les workbooks 40](#_Toc66266868)

[4|14 Déployer les playbooks 41](#_Toc66266869)

[5 Intégration Manuelle 59](#_Toc66266870)

[5|1 Déploiement des hunting rules 59](#_Toc66266871)

[5|2 Déploiement des workbooks 61](#_Toc66266872)

[5|3 Déploiement des playbooks 63](#_Toc66266873)

# 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 de trois manières :

* Intégration par Script Powershell
* Intégration par les github actions
* Intégration manuelle

# Intégration Script

## Synchroniser le repository SentinelAsCode

-Télécharger et installer git pour windows :

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

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

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

**Exemple d’Intégration :**

#git clone --branch <clientid> <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) :

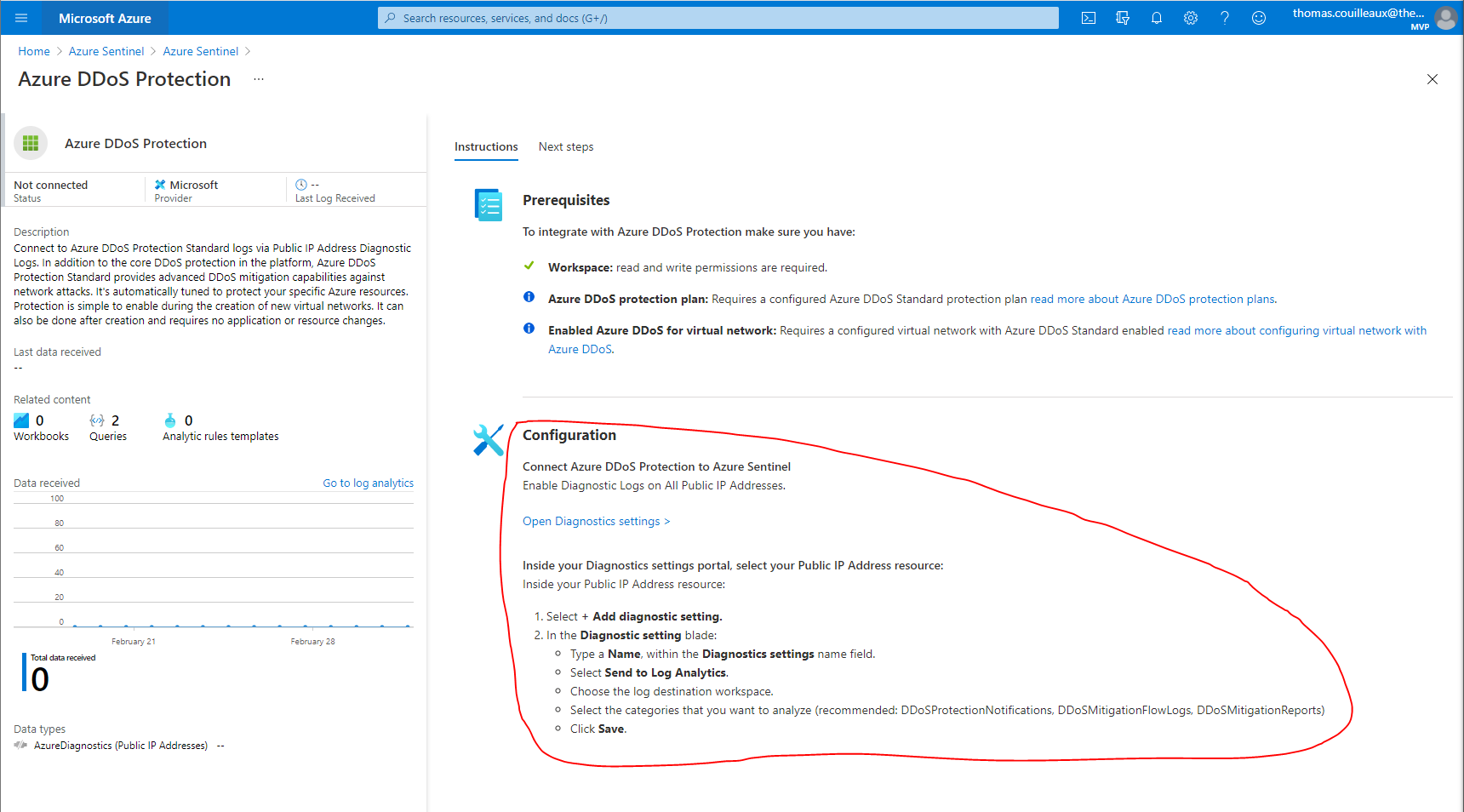
* Se connecter au portail de gestion des workspaces dans Sentinel :

<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

-L’on peut déployer un workbook ou un ensemble de workbook par l’utilisation d’une wildcard sur le paramètre « *WorkbooksFolder* » :

**Exemple d’Intégration :**

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

## 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.

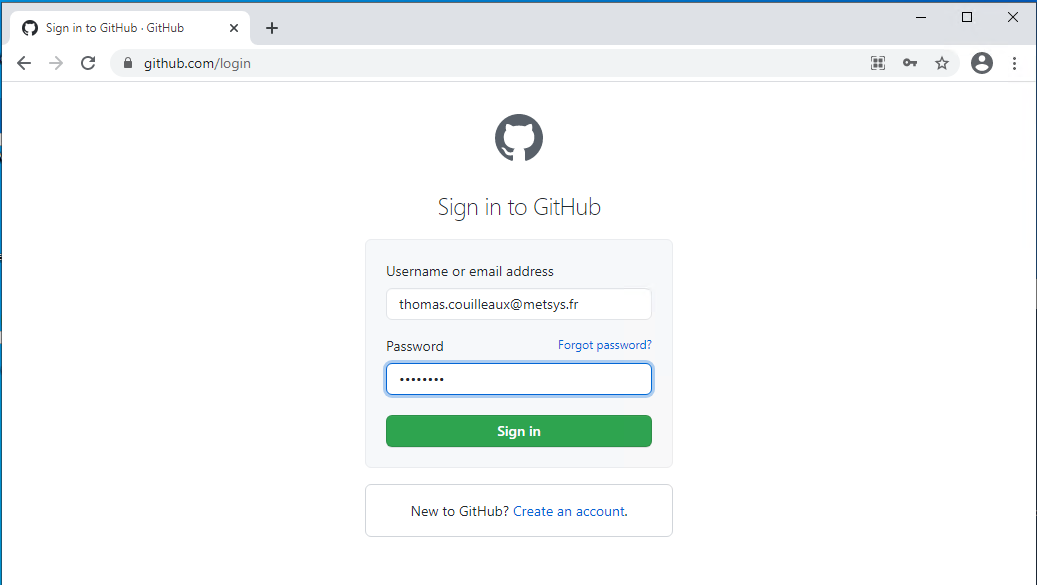


# Intégration Industrialisée

## Synchroniser le repository SentinelAsCode

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

https://github.com/login



-Télécharger et installer git pour windows :

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

-Télécharger et installer chrome pour windows :

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

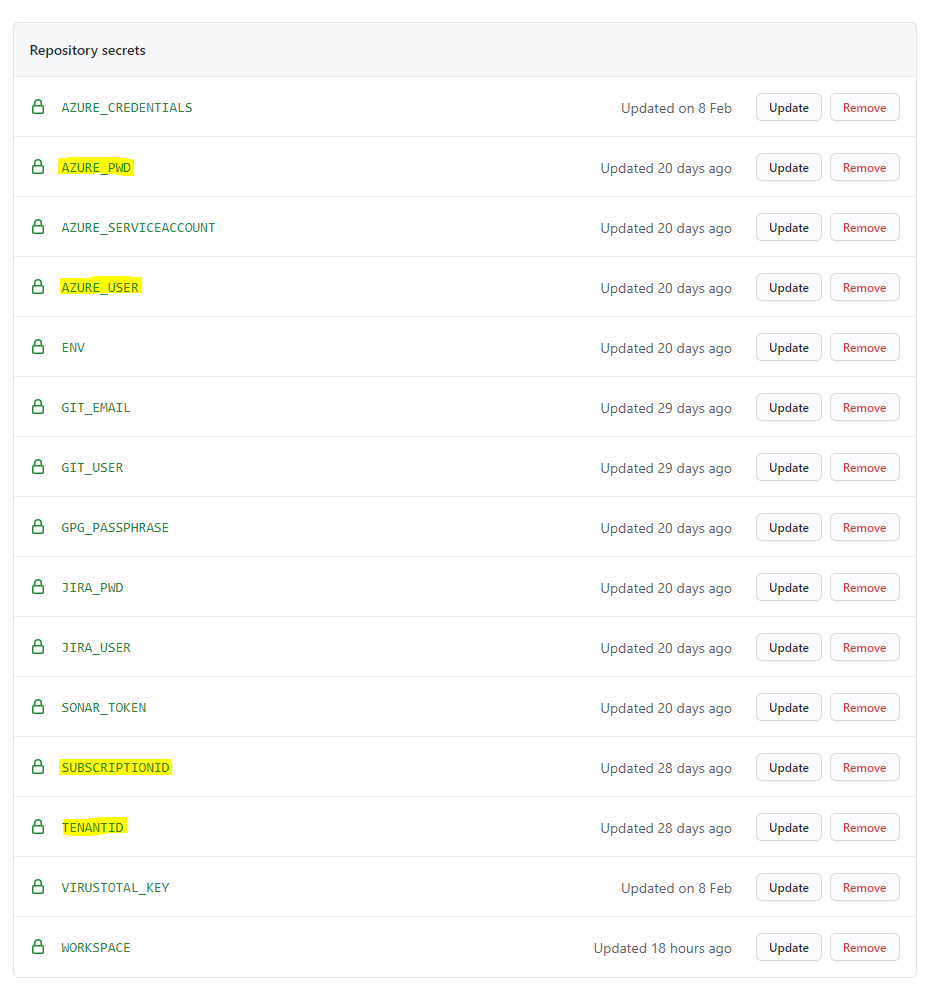
|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **ClientID** | Identifiant du client. | client1 |

## 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> |
| **AzureUser** | Identifiant pour se connecter à Azure. | <String> |
| **AzurePwd** | Mot de passe pour se connecter à Azure. | <String> |

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



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

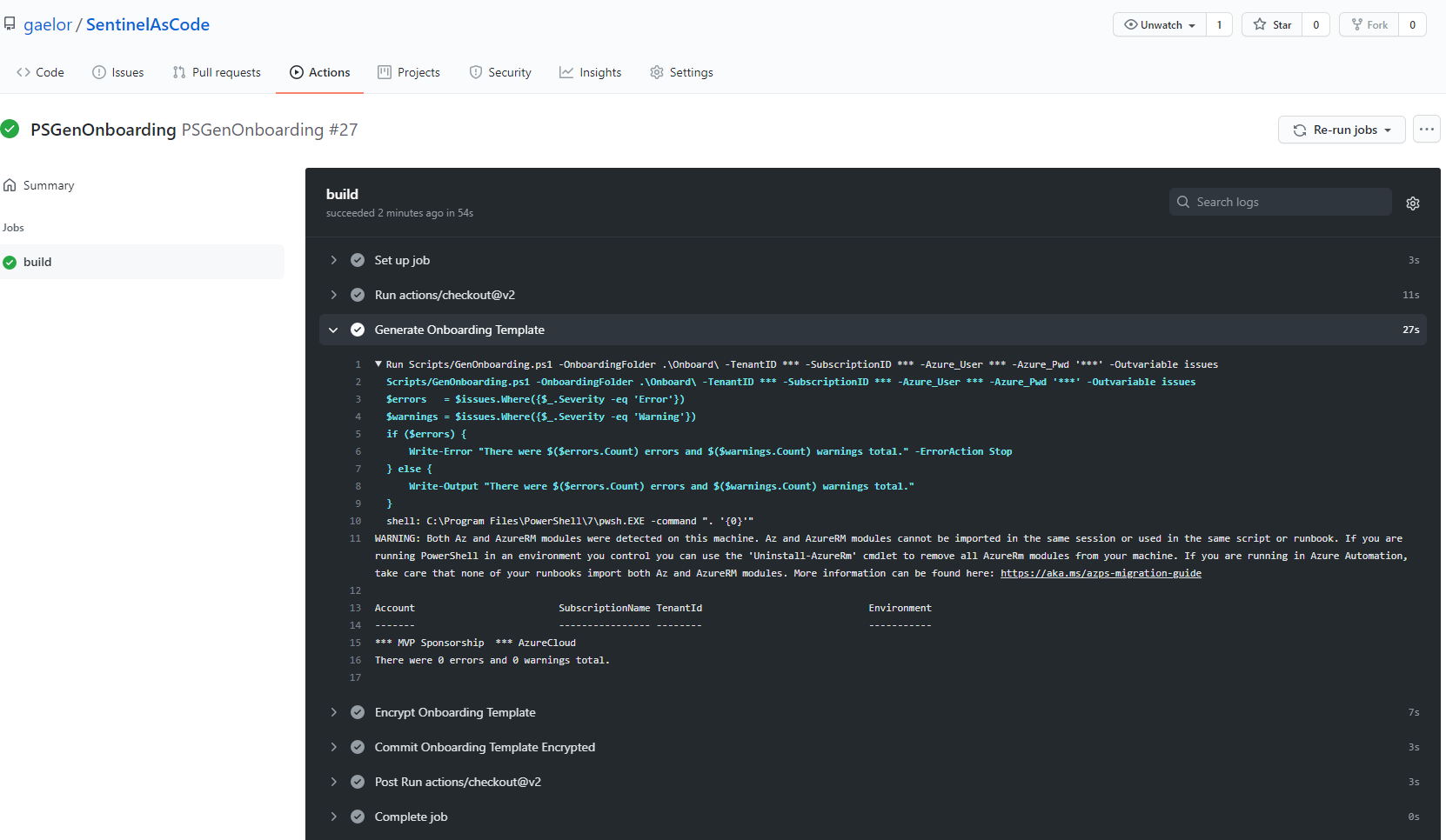
https://github.com/gaelor/SentinelAsCode/actions/workflows/PSGenOnboarding.yml



-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



-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

-Synchroniser le fichier généré.

**Exemple d’Intégration :**

#git.exe pull

remote: Enumerating objects: 13, done.

remote: Counting objects: 100% (13/13), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 7 (delta 4), reused 7 (delta 4), pack-reused 0

Unpacking objects: 100% (7/7), 53.53 KiB | 194.00 KiB/s, done.

From https://github.com/gaelor/SentinelAsCode

c6eaa66..a03ba79 dev -> origin/dev

3da260e..c2becdf client1 -> origin/client1

Updating c6eaa66..a03ba79

Fast-forward

Onboard/onboarding\_09032021.json.gpg | Bin 0 -> 354 bytes

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 Onboard/onboarding\_09032021.json.gpg

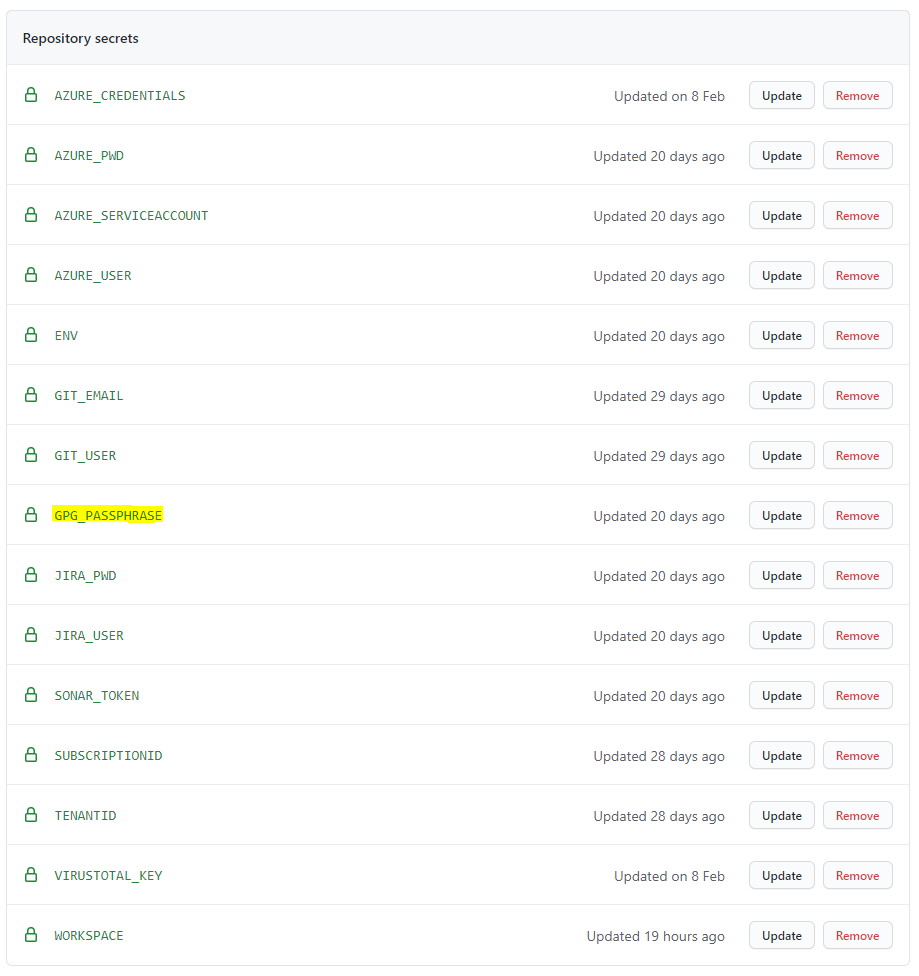
## Adapter le fichier de onboarding

-Télécharger et installer GPG pour windows :

https://gnupg.org/download/

-Récupérer ou mettre à jour la passphrase GPG :

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



-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

-Configurer son compte github et pousser la modification :

|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **email** | Git email address. | <String> |
| **username** | Git username. | <String> |

**Exemple d’Intégration :**

#git config --global user.email "<email>"

#git config --global user.name "<username>"

#git commit -a -m "Add Onboarding File"

[dev acad074] Add Onboarding File

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

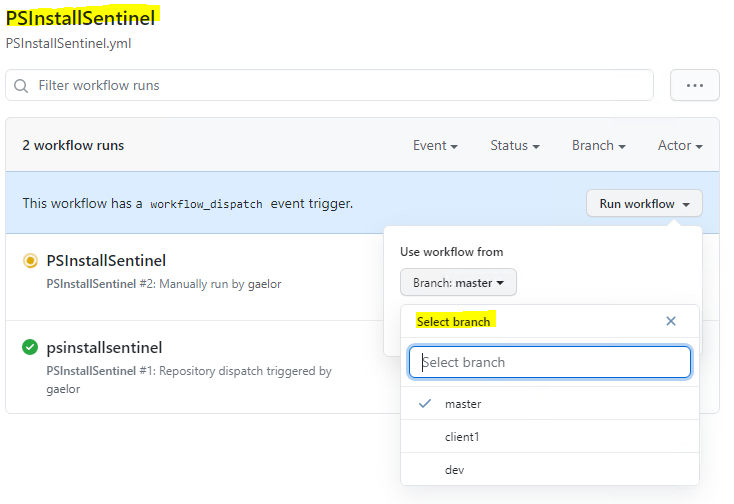
a03ba79..acad074 dev -> dev

## Activer Sentinel

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

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

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



## Déployer les connecteurs

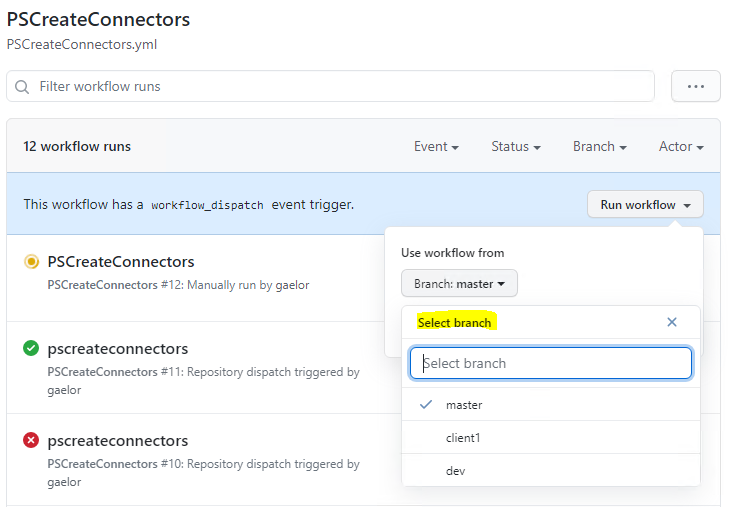
-La documentation des connecteurs est disponible à cette adresse :

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 :

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

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



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

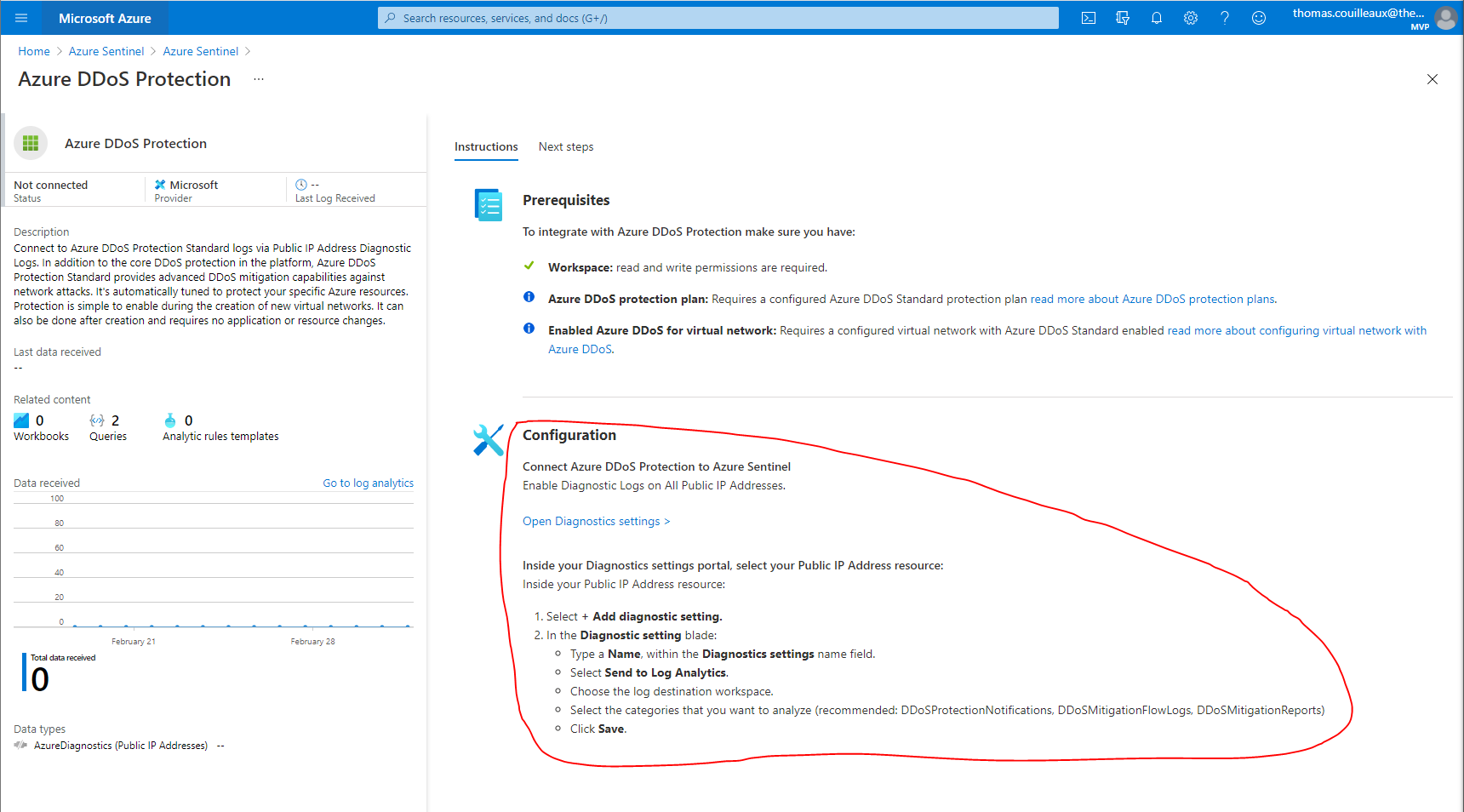
* Se connecter au portail de gestion des workspaces dans Sentinel :

<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 :

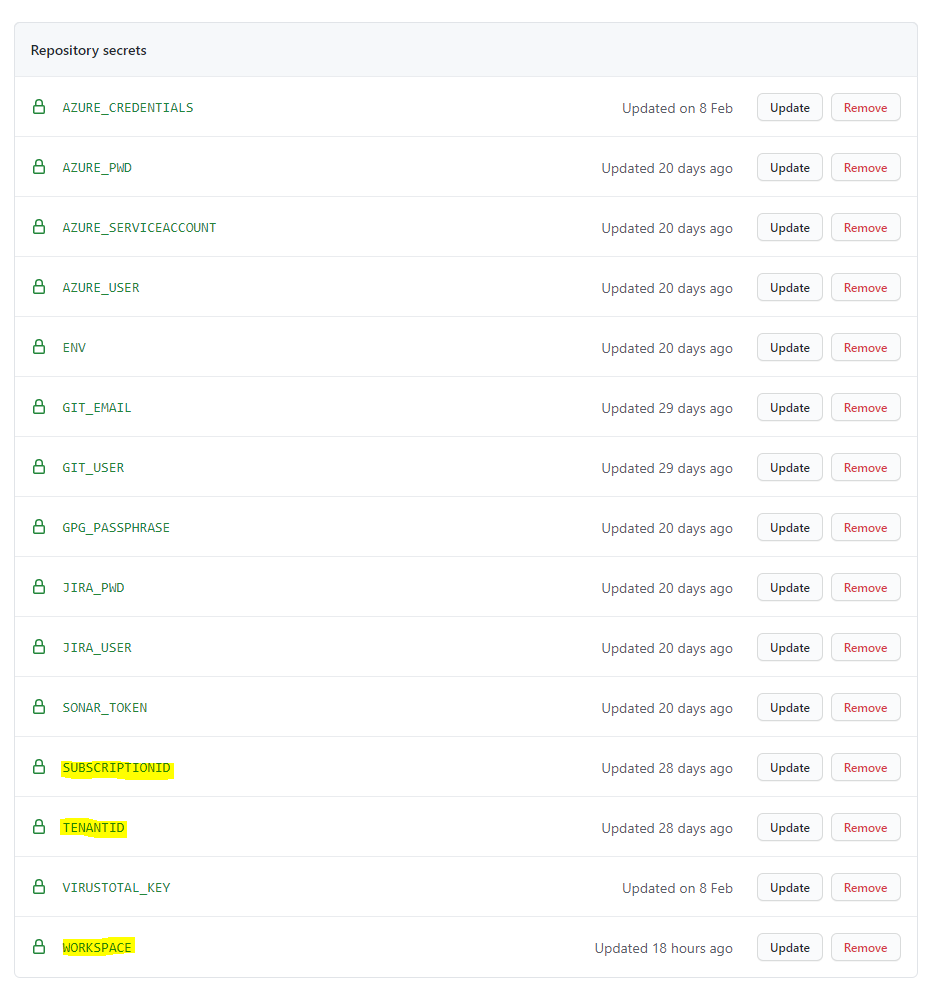


## Générer le fichier de analytics rules Microsoft

-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> |

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



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)

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

**Exemple d’Intégration :**

#git.exe pull

remote: Enumerating objects: 13, done.

remote: Counting objects: 100% (13/13), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 7 (delta 4), reused 7 (delta 4), pack-reused 0

Unpacking objects: 100% (7/7), 53.53 KiB | 194.00 KiB/s, done.

From https://github.com/gaelor/SentinelAsCode

c6eaa66..a03ba79 dev -> origin/dev

3da260e..c2becdf client1 -> origin/client1

Updating c6eaa66..a03ba79

Fast-forward

AnalyticsRules/analytics-rules\_09032021.json | Bin 0 -> 354 bytes

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 AnalyticsRules/analytics-rules\_09032021.json

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

**Exemple d’Intégration :**

#git commit -a -m "Add AnalyticsRules File"

[dev acad074] Add AnalyticsRules File

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

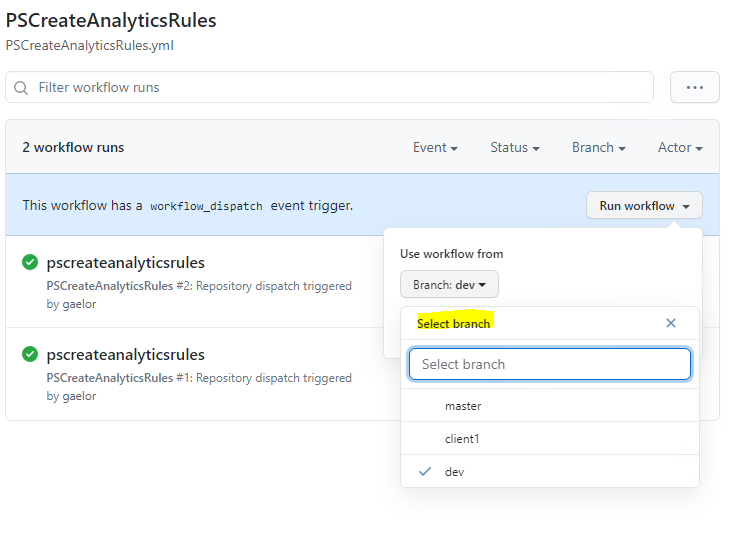
## Déployer les analytics rules

-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 :

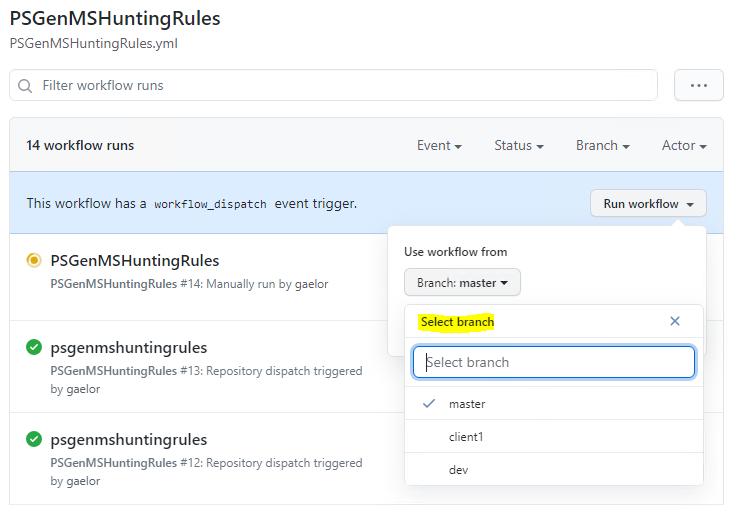
https://github.com/gaelor/SentinelAsCode/actions/workflows/PSCreateAnalyticsRules.yml



## Générer le fichier de hunting rules Microsoft

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

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



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)

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

**Exemple d’Intégration :**

#git.exe pull

remote: Enumerating objects: 13, done.

remote: Counting objects: 100% (13/13), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 7 (delta 4), reused 7 (delta 4), pack-reused 0

Unpacking objects: 100% (7/7), 53.53 KiB | 194.00 KiB/s, done.

From https://github.com/gaelor/SentinelAsCode

c6eaa66..a03ba79 dev -> origin/dev

3da260e..c2becdf client1 -> origin/client1

Updating c6eaa66..a03ba79

Fast-forward

HuntingRules/MS\_hunting-rules\_09032021.json | Bin 0 -> 354 bytes

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 HuntingRules/MS\_hunting-rules\_09032021.json

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

**Exemple d’Intégration :**

#git commit -a -m "Add HuntingRules File"

[dev acad074] Add HuntingRules File

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

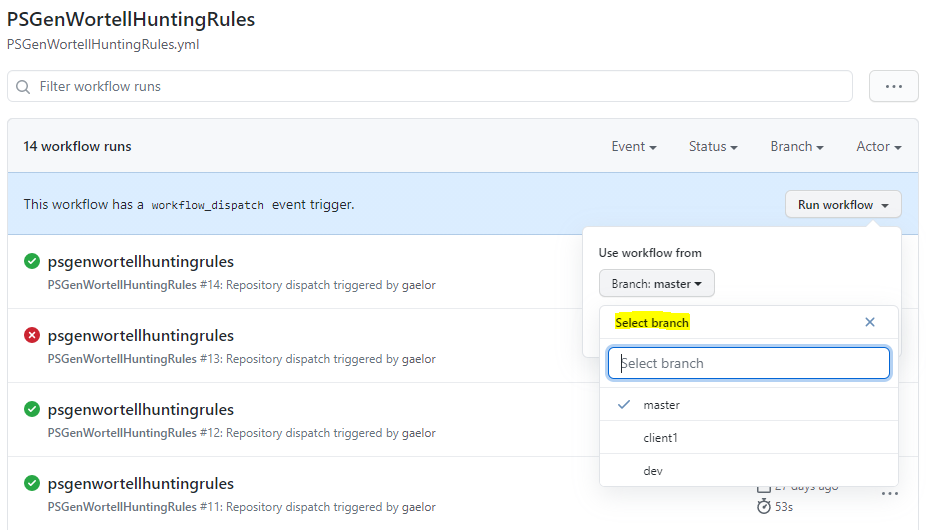
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

## Générer le fichier de hunting rules Wortell

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

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



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)

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

**Exemple d’Intégration :**

#git.exe pull

remote: Enumerating objects: 13, done.

remote: Counting objects: 100% (13/13), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 7 (delta 4), reused 7 (delta 4), pack-reused 0

Unpacking objects: 100% (7/7), 53.53 KiB | 194.00 KiB/s, done.

From https://github.com/gaelor/SentinelAsCode

c6eaa66..a03ba79 dev -> origin/dev

3da260e..c2becdf client1 -> origin/client1

Updating c6eaa66..a03ba79

Fast-forward

HuntingRules/Wortell\_hunting-rules\_09032021.json | Bin 0 -> 354 bytes

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 HuntingRules/Wortell\_hunting-rules\_09032021.json

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

**Exemple d’Intégration :**

#git commit -a -m "Add HuntingRules File"

[dev acad074] Add HuntingRules File

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

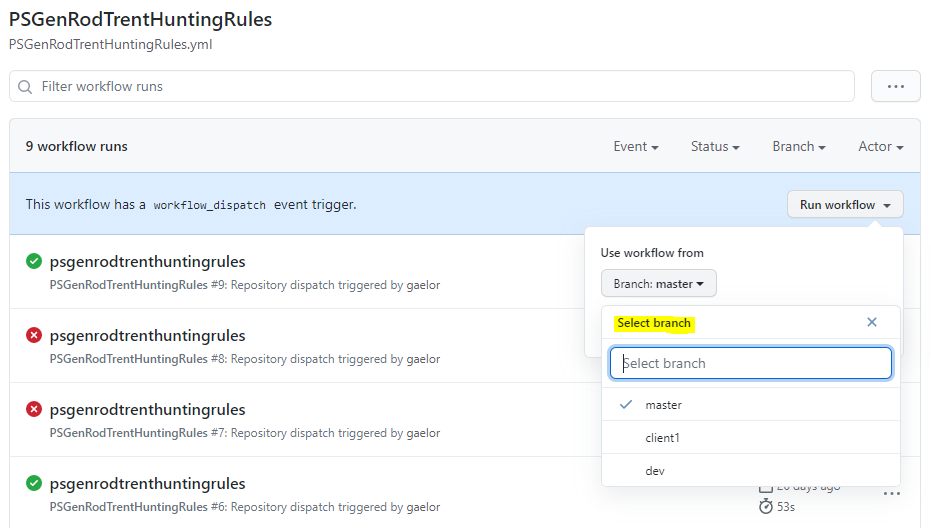
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

## Générer le fichier de hunting rules RodTrent

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

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



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)

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

**Exemple d’Intégration :**

#git.exe pull

remote: Enumerating objects: 13, done.

remote: Counting objects: 100% (13/13), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 7 (delta 4), reused 7 (delta 4), pack-reused 0

Unpacking objects: 100% (7/7), 53.53 KiB | 194.00 KiB/s, done.

From https://github.com/gaelor/SentinelAsCode

c6eaa66..a03ba79 dev -> origin/dev

3da260e..c2becdf client1 -> origin/client1

Updating c6eaa66..a03ba79

Fast-forward

HuntingRules/RodTrent\_hunting-rules\_09032021.json | Bin 0 -> 354 bytes

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 HuntingRules/RodTrent\_hunting-rules\_09032021.json

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

**Exemple d’Intégration :**

#git commit -a -m "Add HuntingRules File"

[dev acad074] Add HuntingRules File

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

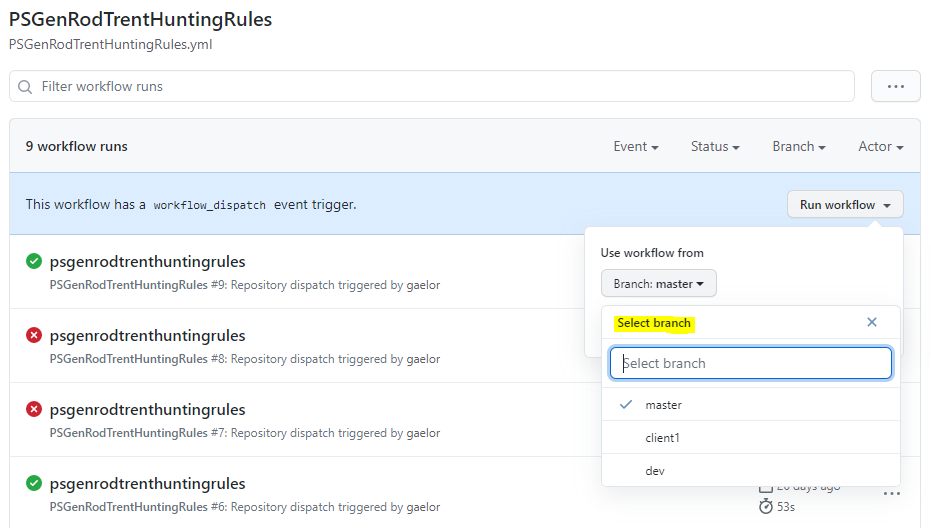
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

## Générer le fichier de hunting rules BlueTeam

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

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



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)

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

**Exemple d’Intégration :**

#git.exe pull

remote: Enumerating objects: 13, done.

remote: Counting objects: 100% (13/13), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 7 (delta 4), reused 7 (delta 4), pack-reused 0

Unpacking objects: 100% (7/7), 53.53 KiB | 194.00 KiB/s, done.

From https://github.com/gaelor/SentinelAsCode

c6eaa66..a03ba79 dev -> origin/dev

3da260e..c2becdf client1 -> origin/client1

Updating c6eaa66..a03ba79

Fast-forward

HuntingRules/BlueTeam\_hunting-rules\_09032021.json | Bin 0 -> 354 bytes

1 file changed, 0 insertions(+), 0 deletions(-)

create mode 100644 HuntingRules/BlueTeam\_hunting-rules\_09032021.json

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

**Exemple d’Intégration :**

#git commit -a -m "Add HuntingRules File"

[dev acad074] Add HuntingRules File

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

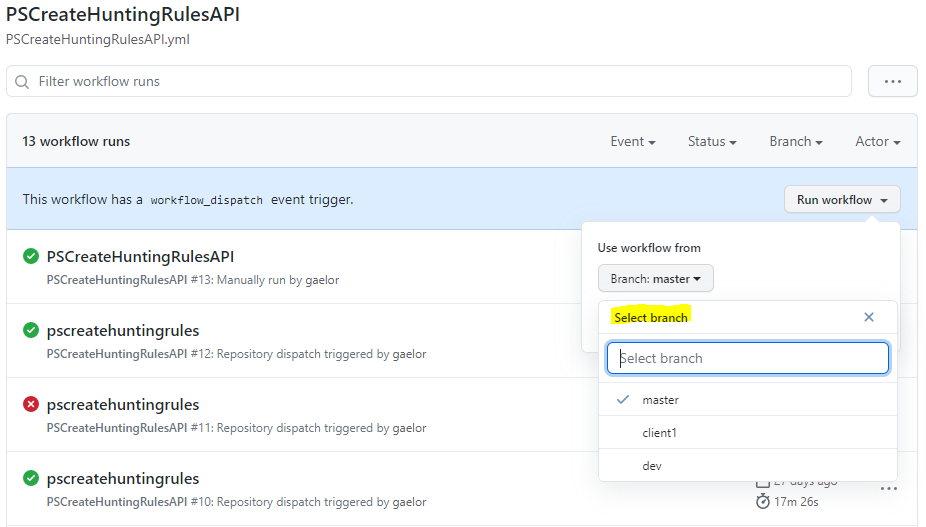
## Déployer les hunting rules

-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 :

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



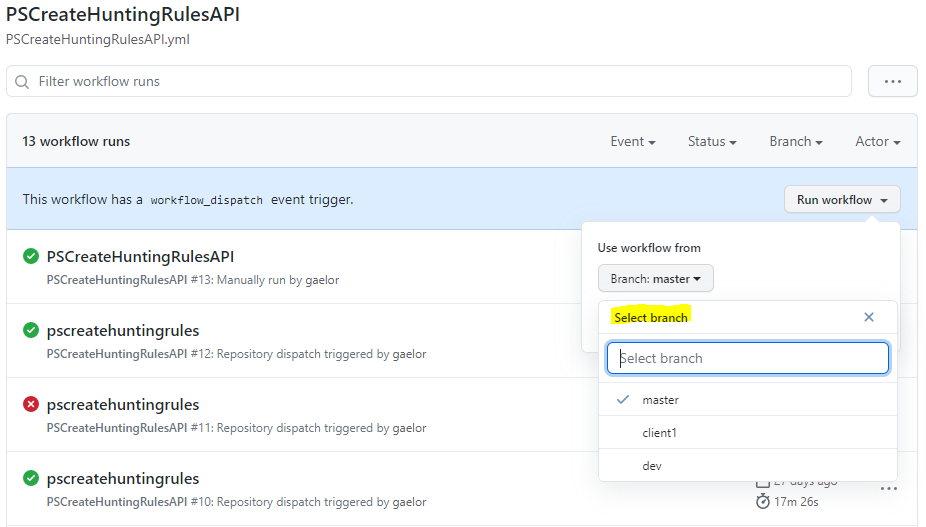
## Déployer les workbooks

-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 :

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



-L’on peut déployer un workbook ou un ensemble de workbook par l’utilisation d’une wildcard sur le paramètre « *WorkbooksFolder* » :

* Editer le fichier « *.github\workflows\PSCreateWorkbooks.yml* » et modifier la section « Create Workbooks on Client Tenant » :

**Exemple d’Intégration :**

Scripts/CreateWorkbooks.ps1 -OnboardingFile Onboard\onboarding.json -WorkbooksFolder Workbooks/AAD\*.json -Azure\_User ${{ secrets.AZURE\_USER }} -Azure\_Pwd '${{ secrets.AZURE\_PWD }}' -Outvariable issues

* Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Workbooks Workflow"

[dev acad074] Modify Workbooks Workflow

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

## 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

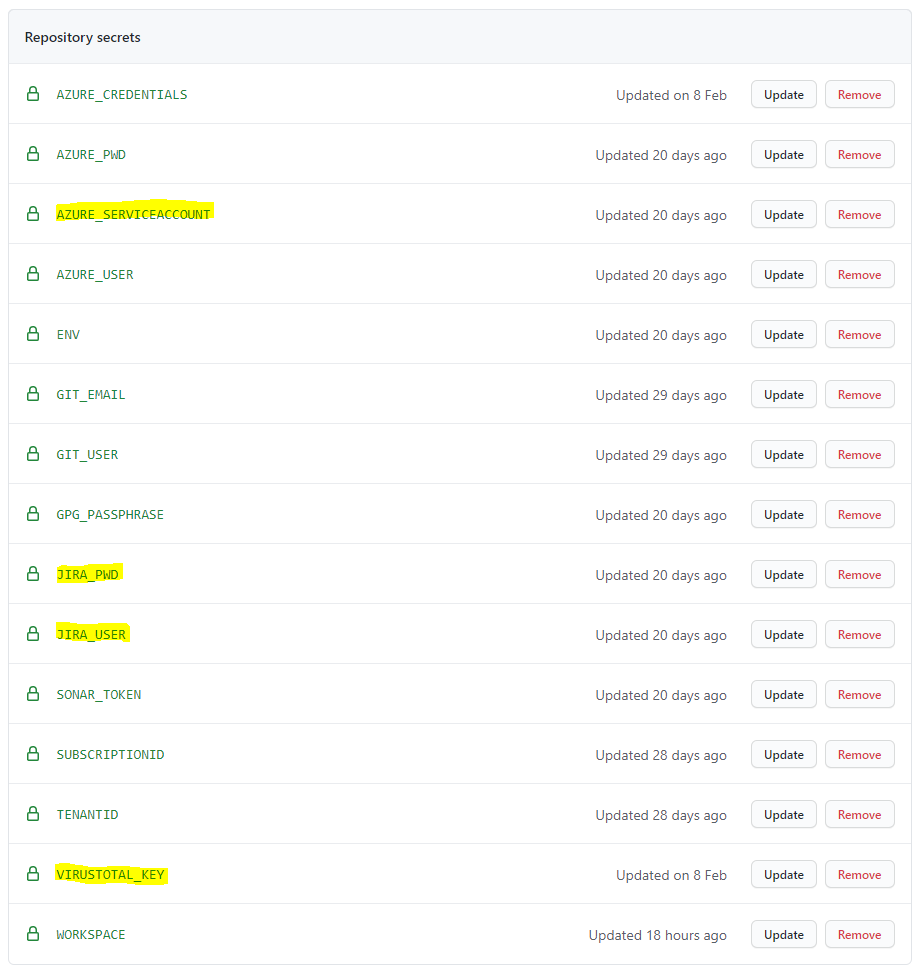
### Adapter le fichier des paramètres pour les playbooks.

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

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

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

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



### 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": {

}

}

-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

-Editer le fichier « *.github\workflows\PSCreatePlaybooks.yml* » et modifier la section « Create Playbooks on Client Tenant » :

**Exemple d’Intégration :**

Scripts/CreatePlaybooks.ps1 -OnboardingFile Onboard\onboarding.json - PlaybooksFolder Playbooks\\*MDATP\*.json -PlaybooksParamsFile Playbooks\Playbooks.params

-Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Playbooks Workflow and Parameters"

[dev acad074] Modify Playbooks Workflow and Parameters

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

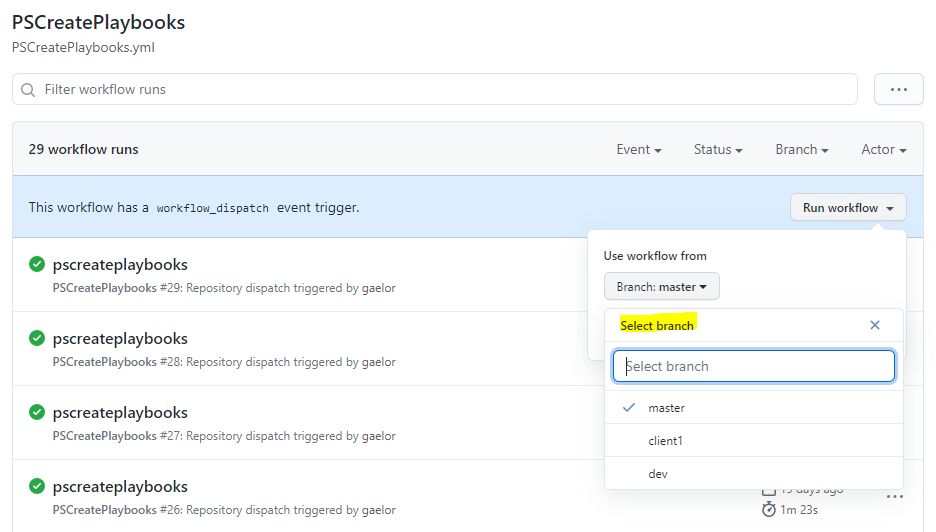
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

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

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



### 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": {

}

}

-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

-Editer le fichier « *.github\workflows\PSCreatePlaybooks.yml* » et modifier la section « Create Playbooks on Client Tenant » :

**Exemple d’Intégration :**

Scripts/CreatePlaybooks.ps1 -OnboardingFile Onboard\onboarding.json - PlaybooksFolder Playbooks\Restrict-MDATPIPAddress.json -PlaybooksParamsFile Playbooks\Playbooks.params

-Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Playbooks Workflow and Parameters"

[dev acad074] Modify Playbooks Workflow and Parameters

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

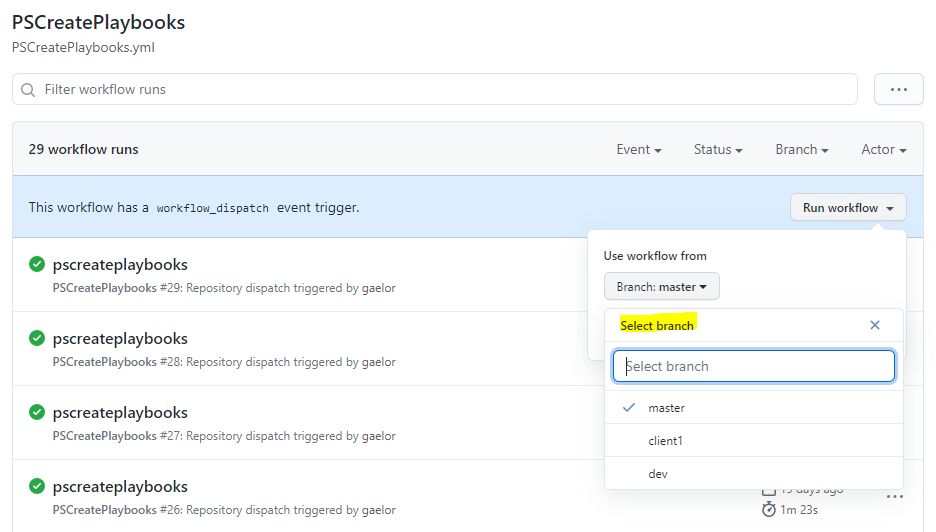
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

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

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



### 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": {

}

}

-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

-Editer le fichier « *.github\workflows\PSCreatePlaybooks.yml* » et modifier la section « Create Playbooks on Client Tenant » :

**Exemple d’Intégration :**

Scripts/CreatePlaybooks.ps1 -OnboardingFile Onboard\onboarding.json - PlaybooksFolder Playbooks\\*AAD\*.json -PlaybooksParamsFile Playbooks\Playbooks.params

-Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Playbooks Workflow and Parameters"

[dev acad074] Modify Playbooks Workflow and Parameters

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

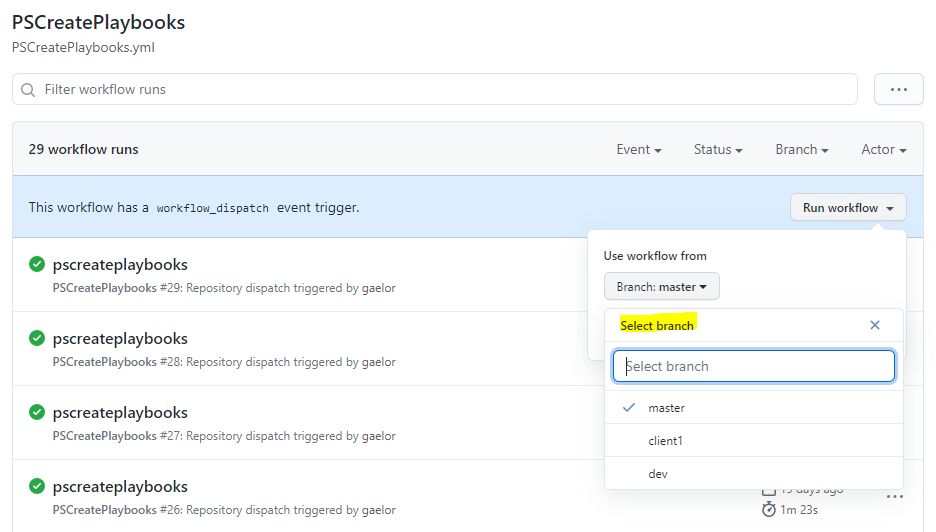
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

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

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



### 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": {

}

}

-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

-Editer le fichier « *.github\workflows\PSCreatePlaybooks.yml* » et modifier la section « Create Playbooks on Client Tenant » :

**Exemple d’Intégration :**

Scripts/CreatePlaybooks.ps1 -OnboardingFile Onboard\onboarding.json - PlaybooksFolder Playbooks\\*Reputation\*.json -PlaybooksParamsFile Playbooks\Playbooks.params

-Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Playbooks Workflow and Parameters"

[dev acad074] Modify Playbooks Workflow and Parameters

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

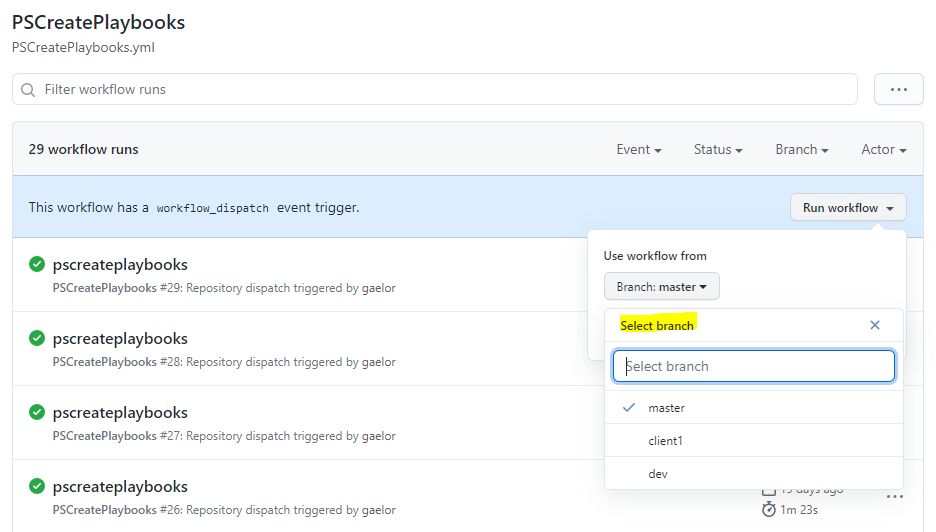
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

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

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



### 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": {

}

}

-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

-Editer le fichier « *.github\workflows\PSCreatePlaybooks.yml* » et modifier la section « Create Playbooks on Client Tenant » :

**Exemple d’Intégration :**

Scripts/CreatePlaybooks.ps1 -OnboardingFile Onboard\onboarding.json - PlaybooksFolder Playbooks\Isolate-NSGMachine.json -PlaybooksParamsFile Playbooks\Playbooks.params

-Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Playbooks Workflow and Parameters"

[dev acad074] Modify Playbooks Workflow and Parameters

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

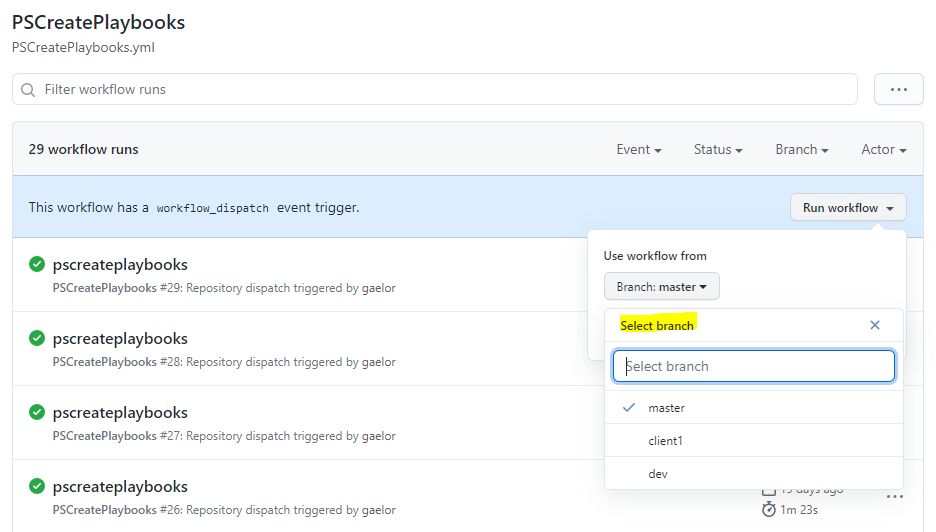
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

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

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



### 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": {

}

}

-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

-Editer le fichier « *.github\workflows\PSCreatePlaybooks.yml* » et modifier la section « Create Playbooks on Client Tenant » :

**Exemple d’Intégration :**

Scripts/CreatePlaybooks.ps1 -OnboardingFile Onboard\onboarding.json - PlaybooksFolder Playbooks\Prompt-User.json -PlaybooksParamsFile Playbooks\Playbooks.params

-Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Playbooks Workflow and Parameters"

[dev acad074] Modify Playbooks Workflow and Parameters

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

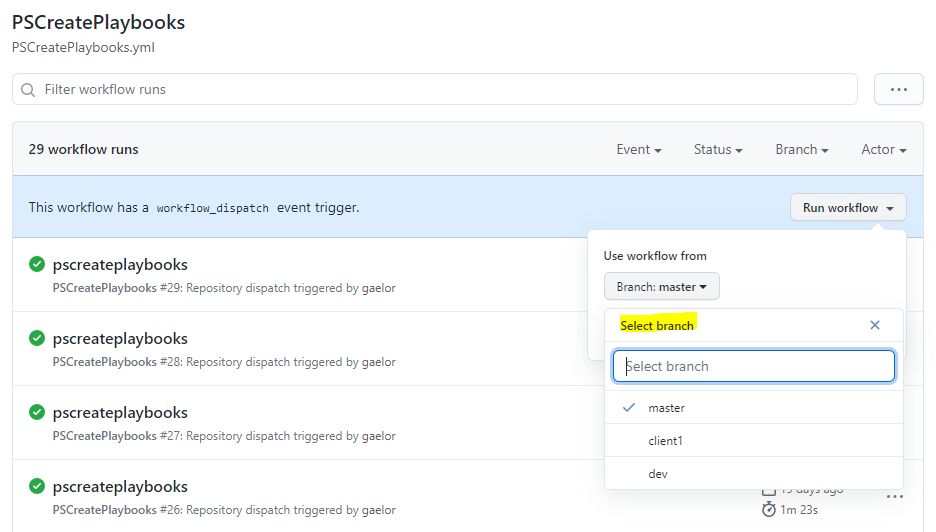
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

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

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



### 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": {

}

}

-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

-Editer le fichier « *.github\workflows\PSCreatePlaybooks.yml* » et modifier la section « Create Playbooks on Client Tenant » :

**Exemple d’Intégration :**

Scripts/CreatePlaybooks.ps1 -OnboardingFile Onboard\onboarding.json - PlaybooksFolder Playbooks\Restrict-CAIPAddress.json -PlaybooksParamsFile Playbooks\Playbooks.params

-Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Playbooks Workflow and Parameters"

[dev acad074] Modify Playbooks Workflow and Parameters

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

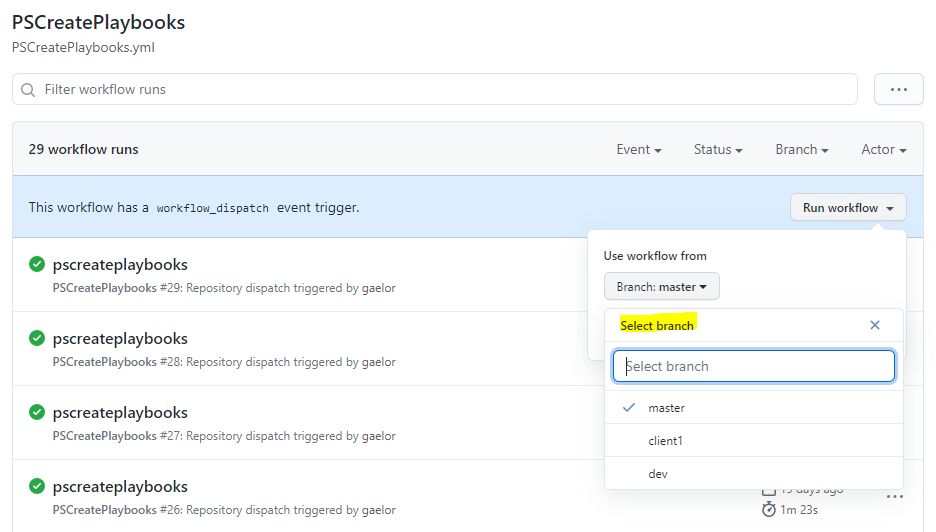
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

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

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



### 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": {

}

}

-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

-Editer le fichier « *.github\workflows\PSCreatePlaybooks.yml* » et modifier la section « Create Playbooks on Client Tenant » :

**Exemple d’Intégration :**

Scripts/CreatePlaybooks.ps1 -OnboardingFile Onboard\onboarding.json - PlaybooksFolder Playbooks\Restrict-NSGIPAddress.json -PlaybooksParamsFile Playbooks\Playbooks.params

-Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Playbooks Workflow and Parameters"

[dev acad074] Modify Playbooks Workflow and Parameters

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

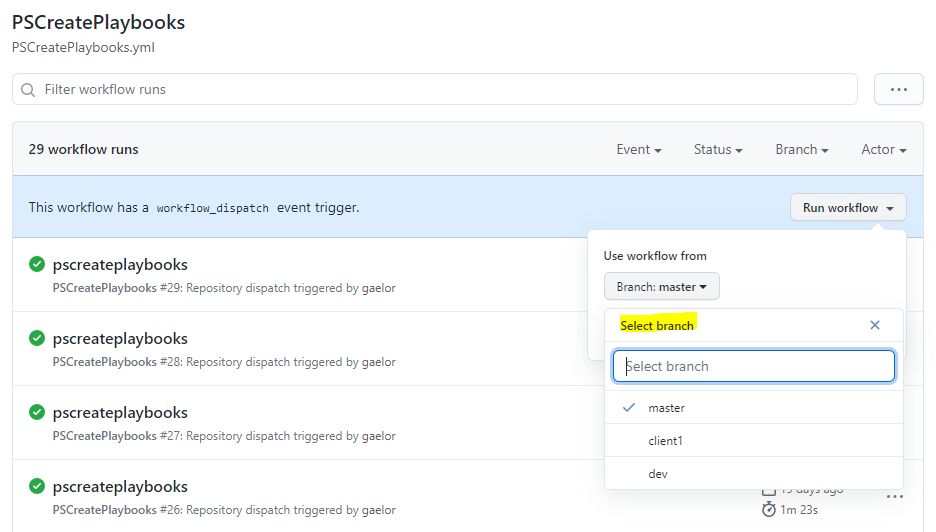
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

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

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



### 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": {

}

}

-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

-Editer le fichier « *.github\workflows\PSCreatePlaybooks.yml* » et modifier la section « Create Playbooks on Client Tenant » :

**Exemple d’Intégration :**

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

-Pousser sur le repo les modifications :

**Exemple d’Intégration :**

#git commit -a -m "Modify Playbooks Workflow and Parameters"

[dev acad074] Modify Playbooks Workflow and Parameters

1 file changed, 0 insertions(+), 0 deletions(-)

#git push

Enumerating objects: 7, done.

Counting objects: 100% (7/7), done.

Delta compression using up to 8 threads

Compressing objects: 100% (4/4), done.

Writing objects: 100% (4/4), 594 bytes | 594.00 KiB/s, done.

Total 4 (delta 2), reused 0 (delta 0), pack-reused 0

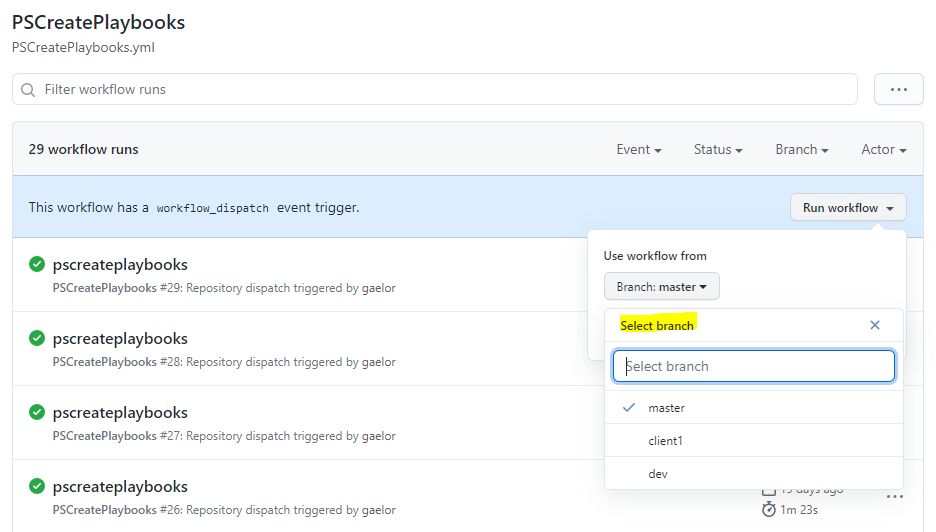
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.

To https://github.com/gaelor/SentinelAsCode.git

a03ba79..acad074 dev -> dev

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

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



### 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 Manuelle

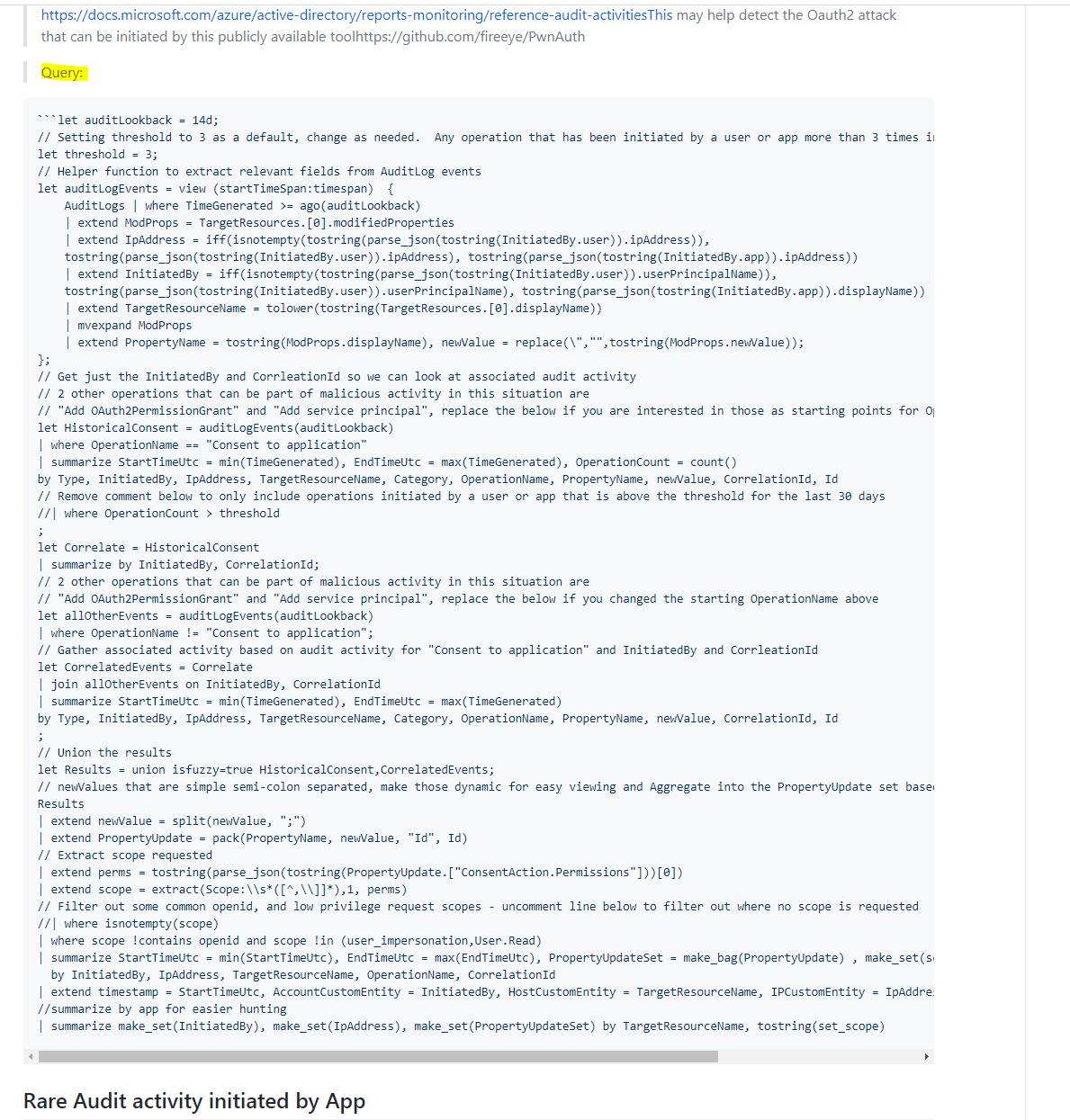
## Déploiement des hunting rules

* Se rendre à cette adresse :

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

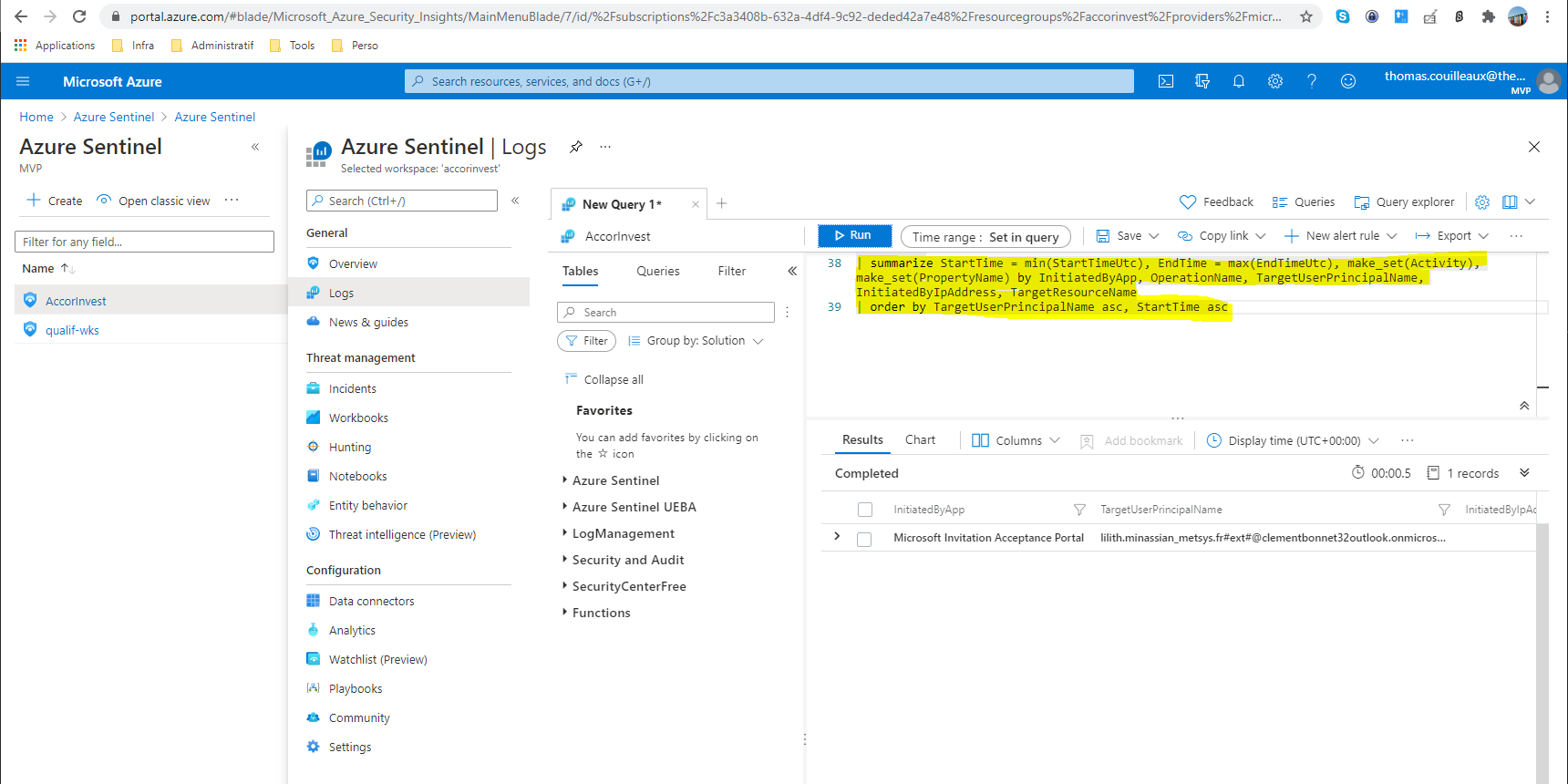
[https://github.com/gaelor/SentinelAsCode/tree/<clientid>/HuntingRules](https://github.com/gaelor/SentinelAsCode/tree/%3cclientid%3e/HuntingRules)

* 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



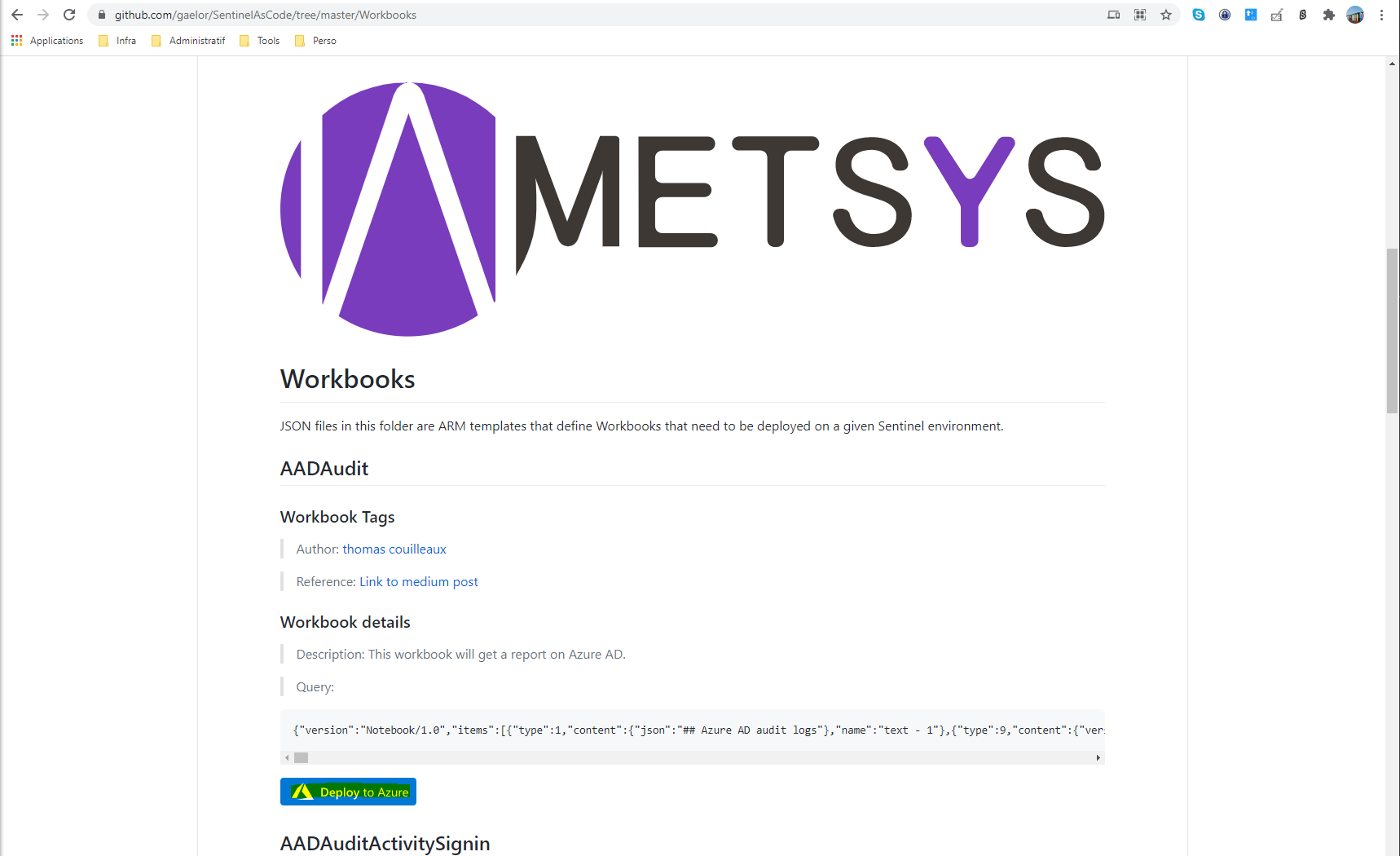
## Déploiement des workbooks

* Se rendre à cette adresse :

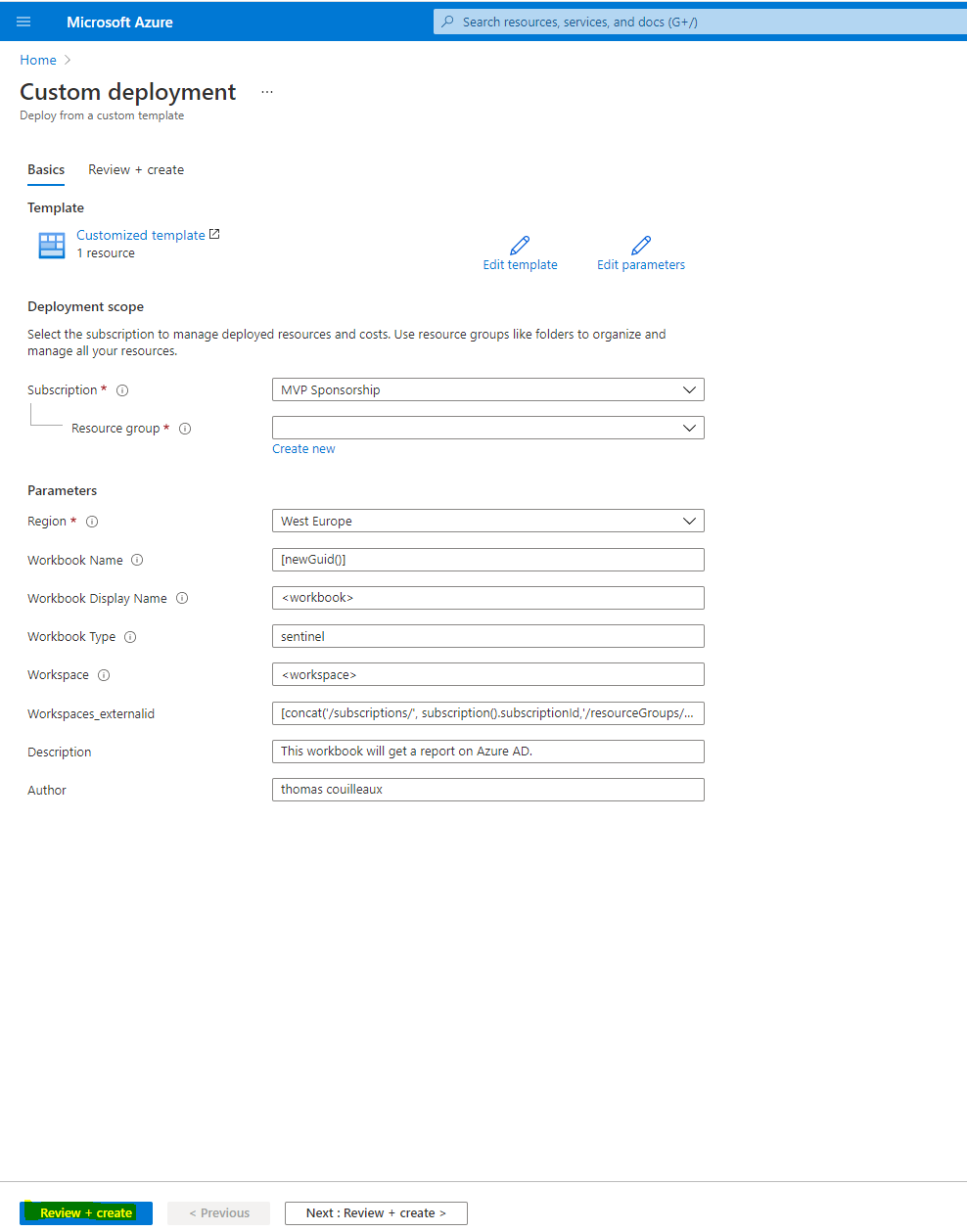
|  |  |  |
| --- | --- | --- |
| **Champ** | **Définition** | **Exemple d’Intégration** |
| **ClientID** | Identifiant du client chez Metsys. | client1 |

[https://github.com/gaelor/SentinelAsCode/tree/<clientid>/Workbooks](https://github.com/gaelor/SentinelAsCode/tree/master/Workbooks)

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



* Configurer les paramètres et déployer :



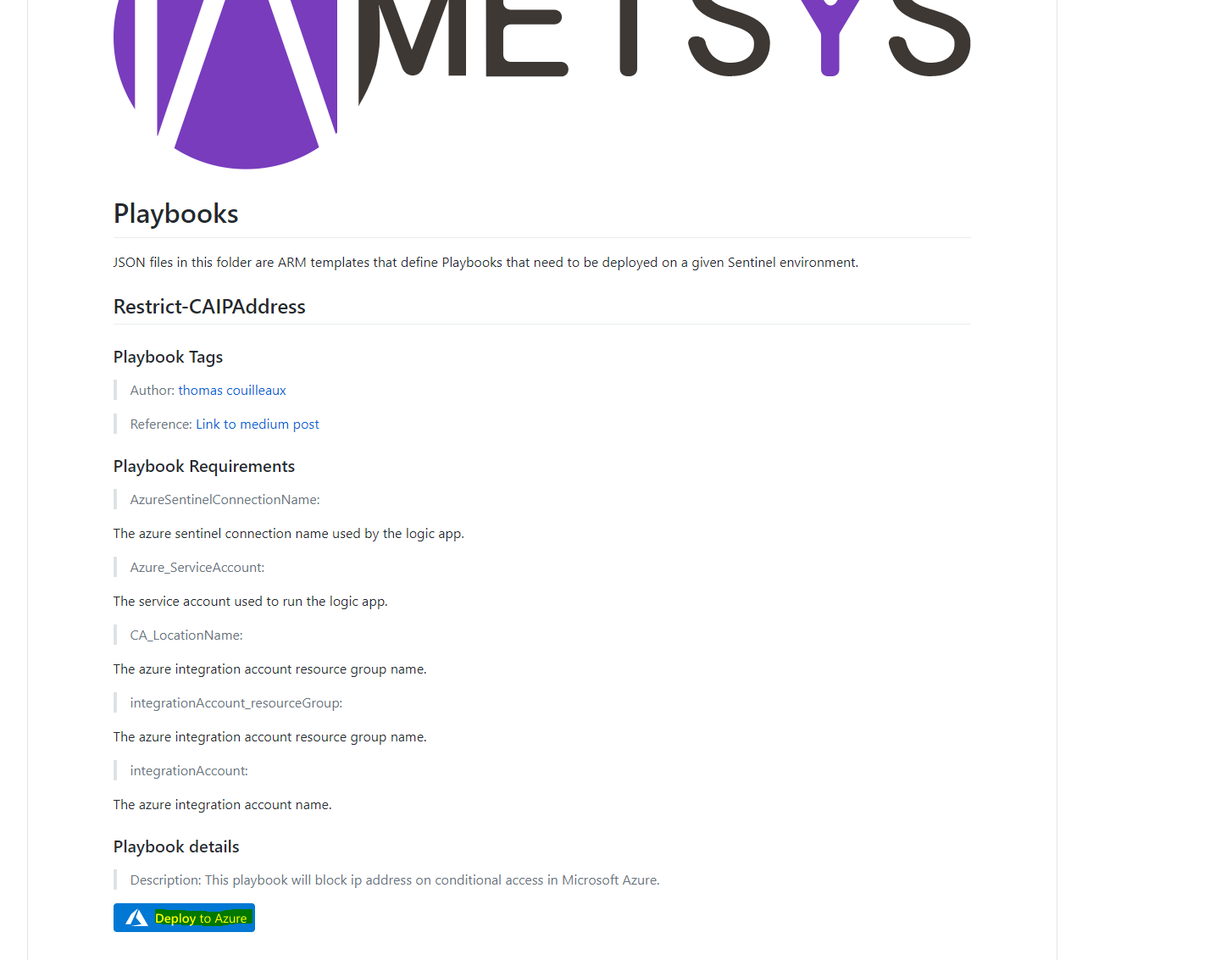
## Déploiement des playbooks

* Se rendre à cette adresse :

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

[https://github.com/gaelor/SentinelAsCode/tree/<clientid>/Playbooks](https://github.com/gaelor/SentinelAsCode/tree/master/Playbooks)

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



* Configurer les paramètres et déployer :

