

NON PROTEGE



**MINISTÈRE
DES ARMÉES**

*Liberté
Égalité
Fraternité*

**Direction générale
de l'armement**

DOCUMENTATION 4

DÉPLOYER AVEC AAP ET STOCKER SES SECRETS SUR GDS

1/30

DGA Maîtrise de l'information

BP 7 – 35998 Rennes CEDEX 9

Téléphone : + 33 (0) 2 99 42 90 11 - Télécopie : + 33 (0) 2 99 42 91 01

NON PROTEGE

Ce document est la propriété de l'État, il ne peut être reproduit, copié ou utilisé sans autorisation écrite.

SOMMAIRE

Historique	3
Acronymes.....	4
Introduction	5
1. Mettre en place sa première machine virtuelle	7
1.1 Créer une machine virtuelle manuellement	7
2. Paramétrer son déploiement avec Ansible Automation Platform.....	10
2.1 Lier l'authentification Galaxy.....	10
2.2 Faire le lien avec la machine virtuelle	10
2.3 Projet exemple pour tester la configuration AAP sur PICSEL	11
2.4 Créer son projet AAP	12
2.5 Relier son inventaire	14
2.6 Créer son modèle de job à lancer	16
17	
2.7 Exécuter son modèle.....	17
3. Configurer ses secrets au sein de Vault GDS	18
3.1 Se connecter à GDS	18
3.2 Stocker ses informations	19
3.3 Utiliser les secrets Vault	22
4. Spécificités des environnements de la plateforme C1DR	23
4.1 La création des machines virtuelles sur C1DR	23
4.2 Configurer son déploiement AAP sur C1DR	23
4.3 La gestion des secrets et des certificats sur C1DR.....	24
Conclusion.....	30

Historique

Tableau 1 : Historique

Version	Date	Auteur	Commentaire
1.0.0	08/2024	Cindy PEREIRA	
1.0.1	11/09/2024	Chloé PAGENEL	Relecture
1.0.2	12/09/2024	Aurélie GRABAS-DOREMUS	
1.0.3	12/09/2024	Chloé PAGENEL	Relecture
1.0.4	27/09/2024	Cindy PEREIRA	Correction d'une information
1.0.5	07/10/2024	Cindy PEREIRA	Correction de coquilles
1.0.6	16/10/2024	Aurélie GRABAS	Changement information d'identification GDS
1.0.7	05/12/2024	Cindy PEREIRA	Retirer mention d'annexe
1.0.8	16/01/2025	Cindy PEREIRA	Modification authentification Galaxy
1.0.9	29/01/2025	Cindy PEREIRA	Ajout précision chemin projet Git sur AAP
1.2.0	04/03/2025	Aurélie GRABAS-DOREMUS	Ajout des informations sur C1DR

Acronymes

Tableau 2 : Acronymes

Acronyme	Description
AAP	Ansible Automation Platform
API	Aplication Programming Interface
GDS	Gestion Des Secrets
OIDC	OpenID Connect
RHEL	Red Hat Enterprise Linux
SSH	Protocol Secure Shell
VM	Machine Virtuelle

Introduction

Ce **chapitre 4** a pour objectifs de vous accompagner sur le déploiement de votre première machine virtuelle, dans votre configuration d'AAP et pour le stockage de secrets dans le gestionnaire Hashicorp Vault GDS.

Ce chapitre commence par présenter comment déployer des machines virtuelles sur la plateforme PICSEL de manière manuelle à l'aide de VRA.

Vous configurerez ensuite l'outil Ansible Automation Platform (AAP) à partir d'un projet exemple sur Gitlab. À l'aide des connaissances acquises grâce à ce tutoriel vous serez en mesure de jouer vos propres playbooks depuis cette plateforme.

La partie suivante décrit la mise en place des secrets dans le gestionnaire de secret Hashicorp Vault GDS.

Enfin, le dernier chapitre vous donne quelques précisions concernant les spécificités de mise en œuvre de cette documentation dans le cadre des environnements de la plateforme C1DR.

Attention : Ces spécificités ne sont à prendre en compte que lors d'un déploiement effectué sur la plateforme C1DR à l'issue de la génération de votre package grâce au passage du pipeline de promotion PICSEL. C'est-à-dire lorsque vous aurez terminé la mise en œuvre des 7 chapitres de la documentation M2C et qu'il vous sera indiqué de vous référer à nouveau à ce présent document.

En résumé, voici les **trois parties majeures** abordées par ce document :

Charge	Titre de la partie	Description
20%	Mettre en place sa première machine virtuelle	Déployer une machine virtuelle avec VRA
60%	Paramétrer son déploiement avec Ansible Automation Platform	Comment utiliser AAP ? Comment lier son projet à cet outil ? Comment jouer ses playbooks ?
20%	Configurer ses secrets au sein de Vault GDS	Comment gérer ses secrets sur Hashicorp Vault GDS ?

Tout au long de la documentation, les encadrés bleus correspondent aux NFR (Non Functional Requirements) liés aux parties traitées. Ces NFR ne sont pas obligatoires mais sont de bonnes pratiques qu'il est conseillé de suivre.

1. Mettre en place sa première machine virtuelle

L'objectif de cette première partie est de vous permettre de créer facilement des machines virtuelles à l'aide du catalogue de services de VRA pour ensuite exécuter vos playbooks dessus.

Cette partie couvre les NFR suivantes :

NFR_SYSTM_03 : Gestion des données sur les VMs : les VMs Linux possèdent un disque de 50 Go et les VM Windows un disque de 100 Go. Ces disques hébergent l'OS et les piles logicielles. Le partitionnement du disque est fixé par les règles de durcissement. Le stockage des données (base de données, fichiers de données) doit être obligatoirement réalisé sur un second disque et pas sur le disque hébergeant l'OS.

Attention : Dans le cadre d'un déploiement à destination de C1DR reportez-vous aux spécifications indiquées [La création des machines virtuelles sur C1DR](#).

1.1 Créer une machine virtuelle manuellement

Le Service Broker (URL : <https://cloud.picsel.defense.gouv.fr/>) permet de déployer facilement une VM à travers une interface graphique.

Pour cela, il faut suivre les étapes suivantes :

- Accéder au **Catalogue** du Service Broker (URL : <https://cloud.picsel.defense.gouv.fr/automation/>)
- Choisir la carte **RHEL 8** (les **informations de connexion** à la machine virtuelle sont indiquées sur la carte).

Attention : Les cartes RHEL 8.x sont obsolètes ; par conséquent quelle que soit la sous-version RHEL 8.x souhaitée, la carte à utiliser est la **RHEL 8**.

- Dans le champ « **Zone projet** » indiquer la ZP sur laquelle vous souhaitez déployer vos VMs.
- Renseigner le « **nom du déploiement** », celui-ci correspond au nom qui sera affiché dans la liste des déploiements par le Service Broker.
- Le « **langage** » par défaut de la VM ; « **en_US** », ne doit **pas être changé**.

- La case relative au « **durcissement** » doit rester cochée.

Attention : L'ajout d'un **disque supplémentaire** est **obligatoire**. En effet, le stockage de données doit être réalisé sur un second disque et pas sur le disque hébergeant l'OS. L'absence de disque générera une erreur au moment de l'exécution des playbooks de déploiement.

Une fois la demande lancée, la création de la machine virtuelle commence et **dure quelques minutes** (en moyenne 15 minutes).

Service Broker [CHANGE](#) ▾

Consume Inbox

Overview

Projects

Showing all

Search projects

Catalog

Deployments ▾

Deployments

Resources

Virtual Machines

Volumes

Networking & Security

RHEL 8 Version [Q 1.5.0](#) [✕](#)

[Documentation RHEL 8](#)

Zone projet * [ZP_MOVE2CLOUD](#) ▾

Nom du déploiement * [VM Test AAP](#)

Nom de la machine virtuelle * [VMTEST](#) ⓘ

Segment réseau * [Développement](#) ⓘ

Gabarit * [2 vCPU / 2 Go RAM](#) ⓘ

Disques ⓘ

☐	Nom du disque ▾	Taille du disque (en Go) ▾
☐	sdb	30

[Manage Columns](#) 1 - 1 of 1

Version * [8.8](#) ⓘ

Langage * [en_US](#) ⓘ

Durcissement ☒ ⓘ

Déploiement rapide ☐ ⓘ

[SUBMIT](#) [CANCEL](#)

Figure 1 : Options recommandées pour la création manuelle d'une machine virtuelle

2. Paramétrer son déploiement avec Ansible Automation Platform

L'objectif de cette partie est de vous accompagner dans le **paramétrage de votre déploiement** à l'aide d'Ansible Automation Platform (**AAP**).

La plateforme AAP simplifie l'exécution des **playbooks Ansible** de votre projet gitlab à travers une interface graphique. Elle est disponible sous PICSEL à l'URL suivante :

<https://aap.picisel.defense.gouv.fr/#/login>.

Ainsi vous pourrez tester vos playbooks et vous assurer de leur bon fonctionnement tout au long de leur développement.

Il est **impératif** d'utiliser AAP pour **valider vos playbooks** dès le début du processus de migration, et ce jusqu'à la mise en place des pipelines Gitlab.

2.1 Lier l'authentification Galaxy

En premier lieu, vous devez ajouter une **authentification Galaxy** à la Zone Projet. Cela permet à AAP d'utiliser les **collections RUCHE** (prérequis obligatoire à la migration sur les C1).

Voici les étapes à suivre :

- Aller dans l'onglet **Organisations** sous le menu **Accès** de AAP.
- Choisir l'organisation au nom de la Zone Projet en question et cliquer sur **Modifier**.
- Dans le champ « Informations d'identification Galaxy », ajouter le jeton créé automatiquement et nommé « Automation Hub Ruche Published Repository ».

Attention : Pour un déploiement sur C1DR à l'issue de la génération de votre package grâce au passage de la pipeline PICSEL reportez-vous aux spécifications indiquées [L'authentification Galaxy sur C1DR](#).

2.2 Faire le lien avec la machine virtuelle

Pour qu'AAP puisse communiquer avec les machines virtuelles de la Zone Projet, il faut créer une **information d'identification**.

Pour cela, il faut suivre les étapes suivantes :

- Aller dans l'onglet **Informations d'identification** sous le menu **Ressources** d'AAP. Le bouton « **Ajouter** » permet de créer une nouvelle information d'identification.
- Choisir la **Zone Projet** souhaitée dans l'« **Organisation** ».

- Pour « **Type d'informations d'identification** », choisir *Machine*. Cela fait apparaître de nouveaux champs à compléter.
- Les champs « **Nom d'utilisateur** » et « **Mot de passe** » correspondent aux informations utilisées pour la connexion aux machines virtuelles. Ces informations sont indiquées sur la **carte** de « création de la machine virtuelle » du **catalogue** du Service Broker.

Attention : Pour utiliser le **mode administrateur** lors du lancement de vos playbooks il est nécessaire de remplir le champ « **Mot de passe pour l'élévation des privilèges** » en utilisant le même **mot de passe** que celui renseigné précédemment.

Attention : Pour un déploiement sur C1DR à l'issue de la génération de votre package grâce au passage de la pipeline PICSEL reportez-vous aux spécifications indiquées [Faire le lien avec la machine virtuelle sur C1DR](#).

2.3 Projet exemple pour tester la configuration AAP sur PICSEL

Pour tester votre configuration d'AAP un playbook d'exemple est mis à votre disposition. Il est disponible sur la **ZS_CONTINUITY** dans le groupe **samples** sous le nom d'« **Exemple playbook AAP** » (URL : https://scm-intredef.picisel.defense.gouv.fr/ZS_CONTINUITY/samples/exemple-playbook-aap).

Pour commencer, vous devez **copier le projet** dans votre Zone Projet de Gitlab afin d'y apporter les modifications nécessaires à son fonctionnement, notamment :

- Le fichier **/inventories/hosts.yml** contient une IP **fictive** qu'il faut remplacer par le **FQDN** de la machine virtuelle créée précédemment. Cette information est la concaténation du nom de la VM avec la zone, comme ceci :
nomVM.picisel.defense.gouv.fr.
Par exemple : **DRMTSTVA123V.picisel.defense.gouv.fr**

Vous pouvez retrouver le nom de votre machine virtuelle sur Service Broker, dans la section « **General** » sous « **Resource name** ».

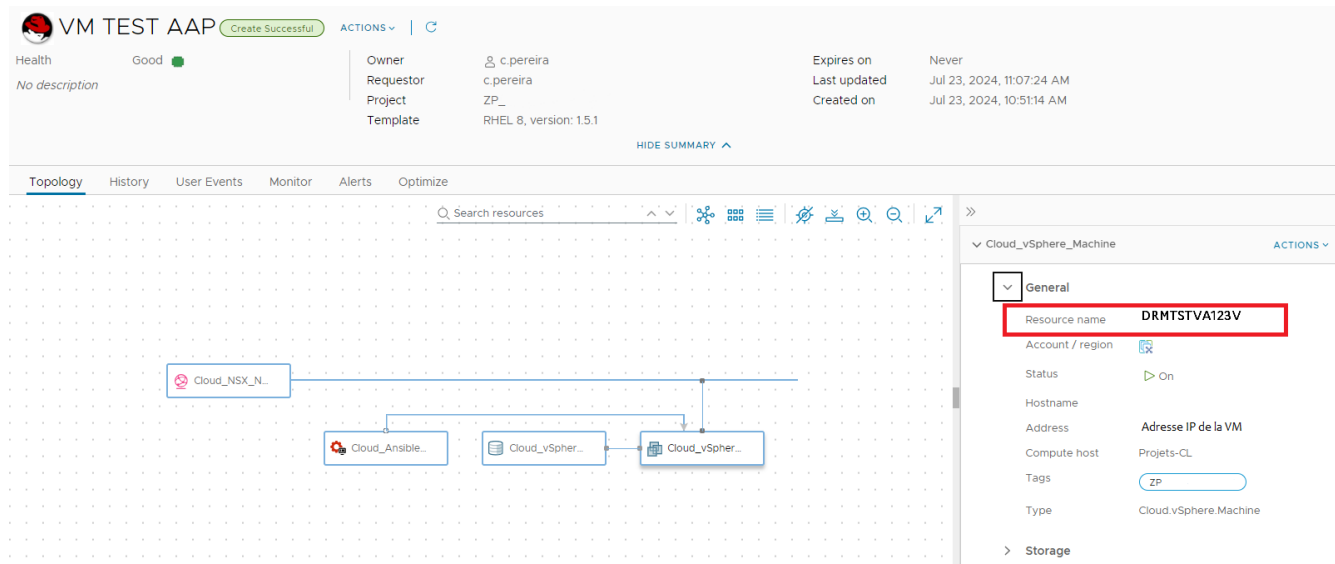


Figure 2 : Récupération du nom de la machine virtuelle

2.4 Créer son projet AAP

Les playbooks et inventaires à exécuter doivent être contenus dans le **dépôt Gitlab** de votre ZP. Voici les étapes à suivre pour lier ce dépôt à un projet AAP :

- Aller dans l'onglet « **Projets** » sous le menu « **Ressources** ».
Le bouton « **Ajouter** » permet de créer un nouveau projet.
- Choisir votre Zone Projet dans le champ « **Organisation** ».
- Sélectionner « **Git** » dans le champ « **Type de contrôle de la source** ».
Cela va ouvrir de nouveaux champs à renseigner.
- Dans le champ « **URL Contrôle de la source** », renseigner l'URL du dépôt Git contenant les playbooks et inventaires, sans renseigner la branche git du projet.
La base de l'URL est **<https://scm-intradef.picsel.defense.gouv.fr>**
- Pour le champ « **Identifiant Contrôle de la source** » sélectionner parmi les options « *Gitlab group access token* » (AAP se charge automatiquement de sa génération lors de la création de la ZP).
- Par défaut, AAP se base sur la branche *main*. Toutefois il est possible de **spécifier la branche Git** à utiliser dans le champ « **Branche** ».
- Dans « **Options** » il faut cocher **toutes les cases sauf** « *Autoriser le remplacement de la branche* ».

The screenshot shows the 'Création d'un projet dans AAP' form. It includes fields for 'Nom' (ZP_MYAPP-git), 'Description', 'Organisation' (ZP_MYAPP), 'Environnement d'exécution', 'Type de Contrôle de la source' (Git), and 'Certificat de validation de la signature du contenu'. Below these is a 'Détails sur le type' section with 'URL Contrôle de la source', 'Branche/ Balise / Commit du Contrôle de la source' (dev), and 'Refspec Contrôle de la source'. There is also an 'Identifiant Contrôle de la source' field with a GitLab token. The 'Options' section has checkboxes for 'Nettoyer', 'Supprimer', 'Suivi des sous-modules', 'Mettre à jour Révision au lancement' (checked), and 'Autoriser le remplacement de la branche'. At the bottom, there is an 'Expiration Délai d'attente du cache' field set to 0.

Figure 3 : Création d'un projet dans AAP

Attention : Même si l'option « Mettre à jour » est cochée, AAP ne se synchronise pas toujours correctement. Il est donc conseillé, après un commit dans le dépôt Git, de **synchroniser à la main le projet dans AAP** (voir Figure 4).

*Figure 4 : Synchronisation manuelle du projet dans AAP*

Attention : Pour un déploiement sur C1DR à l'issue de la génération de votre package grâce au passage de la pipeline PICSEL reportez-vous aux spécifications indiquées [Créer son projet AAP sur C1DR](#)

2.5 Relier son inventaire

Attention : Lors de la création de la ZP, **un inventaire est automatiquement créé** au nom de ZP_NAME où NAME est le nom de votre Zone Projet.

Cet inventaire contient les **informations d'identification Medusa**.

Ne surtout **pas renommer, modifier ni supprimer** cet inventaire au risque de perdre ces informations. Il faut **impérativement** créer un nouvel inventaire pour lancer un playbook.

L'inventaire permet d'indiquer à AAP sur quels **hôtes** exécuter les playbooks ainsi que les variables associées. Pour cela, il faut indiquer à AAP le chemin du fichier couramment nommé « *hosts.yml* » dans le dépôt Gitlab.

Pour cela, il faut suivre les étapes suivantes :

- Dans l'onglet **Inventaires**, sous le menu **Ressources** de AAP, le bouton « **Ajouter un inventaire** » permet de créer un nouvel inventaire.
- Seuls les champs « **Nom** » et « **Organisation** » doivent être renseignés ; les autres champs seront automatiquement mis à jour lors de la synchronisation avec le fichier « *hosts.yml* ».
- Une fois cet inventaire créé et enregistré, il faut cliquer dessus depuis la liste des inventaires et aller dans l'onglet **Sources** de l'inventaire (voir Figure 5).

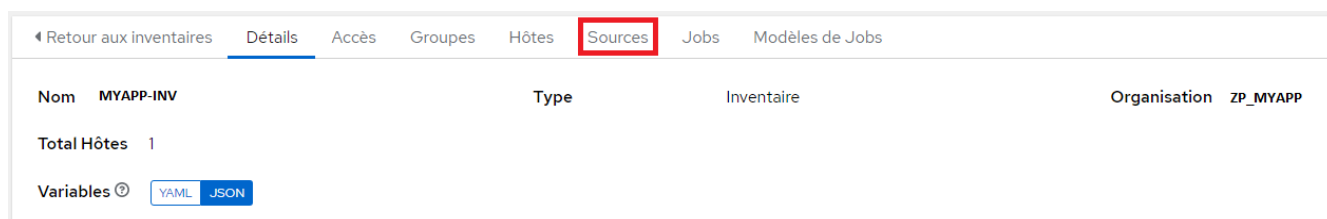


Figure 5 : Exemple d'inventaire AAP

- Le bouton **Ajouter** permet de créer une nouvelle source.
- Dans le champ « **Source** » sélectionner « *Provenance d'un projet* ».
- La localisation du fichier « *hosts.yml* » dans l'architecture du dépôt Git doit être renseignée dans le champ « **Fichiers d'inventaire** ».
- Cocher les trois options de « **Mettre à jour les options** ».

The screenshot shows the 'Ajout de la source à l'inventaire' form. At the top, there are fields for 'Nom' (filled with 'myapp-inv-src'), 'Description', and 'Environnement d'exécution'. Below these is a 'Source' dropdown menu set to 'Provenance d'un projet'. The main section, 'Détails de la source', contains several sub-sections: 'Information d'identification' with a search field; 'Projet' with a dropdown set to 'ZP_MYAPP-git'; 'Fichier d'inventaire' with a dropdown set to 'inventories/review/hosts.yml'; 'Verbosité' with a dropdown set to '1 (Info)'; 'Filtre d'hôte'; 'Variable activée'; 'Valeur activée'; 'Mettre à jour les options' with checkboxes for 'Remplacer', 'Remplacer les variables', and 'Mettre à jour au lancement'; 'Expiration du délai d'attente du cache (secondes)' set to '0'; and 'Variables de la source' with tabs for 'YAML' and 'JSON'. A pagination bar at the bottom shows '1' of 1 items.

Figure 6 : Ajout de la source à l'inventaire

Après la synchronisation de l'inventaire, les hosts VM importés doivent s'afficher dans l'onglet **Hôtes** de l'inventaire (voir Figure 7).

The screenshot shows the 'Hôtes' tab in the Gitlab interface. The top navigation bar includes links for 'Retour aux inventaires', 'Détails', 'Accès', 'Groupes', 'Hôtes' (highlighted), 'Sources', 'Jobs', and 'Modèles de Jobs'. Below the navigation bar, there is a search bar and buttons for 'Ajouter', 'Exécuter Commande', and 'Supprimer'. The table below has columns for 'Nom', 'Description', 'Groupes liés', and 'Actions'. The first row shows a host named 'front1' with description 'imported', linked to the 'front' group, and an 'Actions' column with a toggle switch and a pencil icon.

Nom	Description	Groupes liés	Actions
front1	imported	front	☒ Le

Figure 7 : Liste des hôtes associés à l'inventaire

Attention : Lors de la synchronisation d'un inventaire AAP, le champ **Variables** se remplit automatiquement avec les variables des inventaires indiquées dans le dossier **/inventories/group_vars/** du projet Gitlab. Ce champ **Variables** ne met pas à jour les valeurs des variables présentes auparavant, même si elles ont été modifiées sur Gitlab. Par conséquent, il est nécessaire de vider ce champ avant de synchroniser l'inventaire avec le projet Gitlab.

2.6 Créer son modèle de job à lancer

En dernière étape, il faut créer le modèle de job qui servira de point d'entrée pour exécuter les playbooks Ansible de la Zone Projet.

Pour cela, il faut suivre les étapes suivantes :

- Aller dans l'onglet **Modèles** sous le menu **Ressources** de AAP. Le bouton **Ajouter un modèle de job** permet de créer un nouveau modèle.
- Dans le champ « **Type de Job** » sélectionner Exécuter.
- Renseigner l'inventaire configuré précédemment dans le champ « **Inventaire** ».
- Renseigner le projet créé précédemment dans le champ « **Projet** ».
- Dans le champ « **Playbook** » il faut renseigner le nom du playbook, qui se trouve généralement à la racine du dépôt git, qui lance l'exécution de toutes les tâches (communément appelé *site.yml*).
- Enfin, deux informations sont à renseigner dans le champ « **Informations d'identification** » :
 - La **connexion SSH** de catégorie *Machine* permettant la connexion avec la machine virtuelle, créée dans la partie *Créer une machine virtuelle manuellement* de ce document.
 - La **connexion cloud** de catégorie *GDS DEV ARN Hashicorp Vault AppRole authentication auto* ou *GDS Hashicorp Vault AppRole authentication auto* (selon celui disponible sur votre Zone Projet) permettant la connexion entre AAP et la Gestion des Secrets. Cette information d'authentification est créée automatiquement lors de la création de la Zone Projet.

The screenshot shows the 'Ajouter un modèle de job' form in AAP. The form is organized into several sections. The 'Nom' field is filled with 'ZP_MYAPP-job'. The 'Description' field is empty. The 'Type de Job' dropdown is set to 'Exécuter'. The 'Inventaire' field is filled with 'MYAPP-INV'. The 'Projet' field is filled with 'ZP_MYAPP-git'. The 'Playbook' dropdown is set to 'site.yml'. The 'Informations d'identification' section contains two entries: 'SSH: ZP_MYAPP_SSH;' and 'Cloud: Vault-AppRole...'. The 'Libellés' field is empty. The 'Variables' section has tabs for 'YAML' and 'JSON'. Each field has a search icon and a 'Me le demander au lancement' checkbox.

Figure 8 : Création d'un modèle de job dans AAP

Attention : Pour un déploiement sur C1DR à l'issue de la génération de votre package grâce au passage de la pipeline PICSEL reportez-vous aux spécifications indiquées [Créer son modèle de job à lancer sur C1DR](#).

2.7 Exécuter son modèle

Finalement, pour exécuter l'entière des fichiers il faut lancer un job depuis le modèle que nous venons de configurer. Pour cela, il faut aller dans l'onglet **Modèles** et cliquer sur l'icône **Lancer le modèle** (voir Figure 9). Cette exécution sera ensuite visible dans l'onglet **Jobs**.

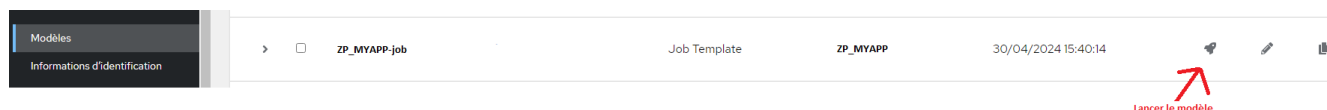


Figure 9 : Exécution d'un modèle

Si vous avez suivi ces étapes avec le playbook donné en exemple, le message suivant devrait s'afficher dans la console de sortie : « **Félicitations, AAP est bien configuré.** »

3. Configurer ses secrets au sein de Vault GDS

Attention : Dans le cadre d'un déploiement à destination de C1DR reportez-vous aux spécifications indiquées [La gestion des secrets et des certificats sur C1DR](#).

L'objectif de cette partie est de comprendre comment utiliser l'outil **Hashicorp Vault GDS** (URL : <https://gds.picisel.defense.gouv.fr/>). Cet outil permet de gérer les secrets pour la migration d'un SI, et est accessible à partir d'une **VDI PICSEL**.

Est considéré comme secret tout token ou mot de passe permettant l'accès aux outils utilisés pour le déploiement de l'application.

Cette partie couvre les NFR suivantes :

NFR_SECRET_01 : Les secrets utilisés par les scripts ANSIBLE doivent être stockés dans GDS. La récupération du secret dans le GDS doit être gérée par le code ANSIBLE du playbook (ou des rôles/collections sous-jacentes) via l'usage d'un rôle fourni par l'équipe GDS.

3.1 Se connecter à GDS

Le **namespace** à renseigner pour la connexion à GDS est « **administration/ZP_NAME** » où NAME est le nom de votre Zone Projet. Votre instance GDS est créée automatiquement à la création de votre Zone Projet.

La méthode de connexion se fait par OIDC en utilisant les identifiants PICSEL. Le rôle doit être laissé vide.

Attention : Veillez à autoriser les **pop-up** dans votre navigateur, car Vault tentera d'en ouvrir une pour vous authentifier.

Sign in to Vault

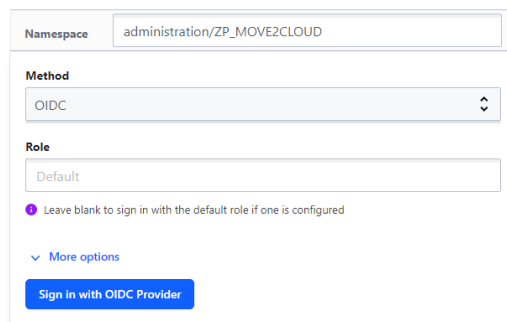


Figure 10 : Connexion à Vault GDS

3.2 Stocker ses informations

Dans GDS, les secrets sont stockés dans des zones appelées secrets engines (ou « moteurs de secrets » en français). Ces zones ont été créées automatiquement lors de la création de votre Zone Projet.

Les secrets doivent être stockés dans le moteur de secrets nommé **se-kv-v2**.

Pour cela, il faut suivre les étapes suivantes :

Cliquer sur **Secrets Engines** dans le menu Vault. Choisir le secret engine **se-kv-v2** dans la liste.

Les secrets doivent être rangés dans des dossiers distincts. Le bouton **Create secret** permet de créer un nouveau dossier de secrets (voir Figure 11).

secrets / se-kv-v2

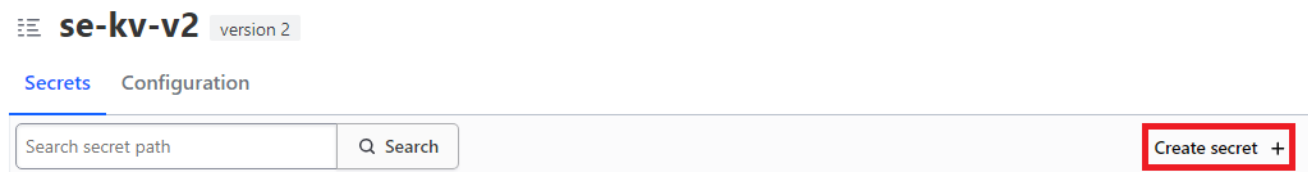


Figure 11 : Création d'un dossier de secrets

3.2.1 Ajouter les secrets Ansible

Cliquer sur **Create secret** et renseigner « ansible » dans le champ **Path for this secret**.

Dans le champ **Secret data**, renseigner la clé « artifactory_password ». La valeur de cette clé est trouvable sur AAP.

Voici les étapes pour récupérer la valeur de la clé « artifactory_password » :

- Dans la liste des inventaires sur AAP, choisir l'inventaire nommé ZP_NAME où NAME est le nom de votre Zone Projet. Cet inventaire est créé et rempli automatiquement lors de la création de votre Zone Projet. Cet inventaire ne doit en aucun cas être modifié ou supprimé.
- Dans les détails de l'inventaire, le champ **Variables** contient les informations d'identification à Medusa (voir Figure 12).
- Copier le token correspondant à la section « medusa_service_build ». Ce token possède, contrairement à medusa_service_dev, les **droits de lecture et d'écriture**.

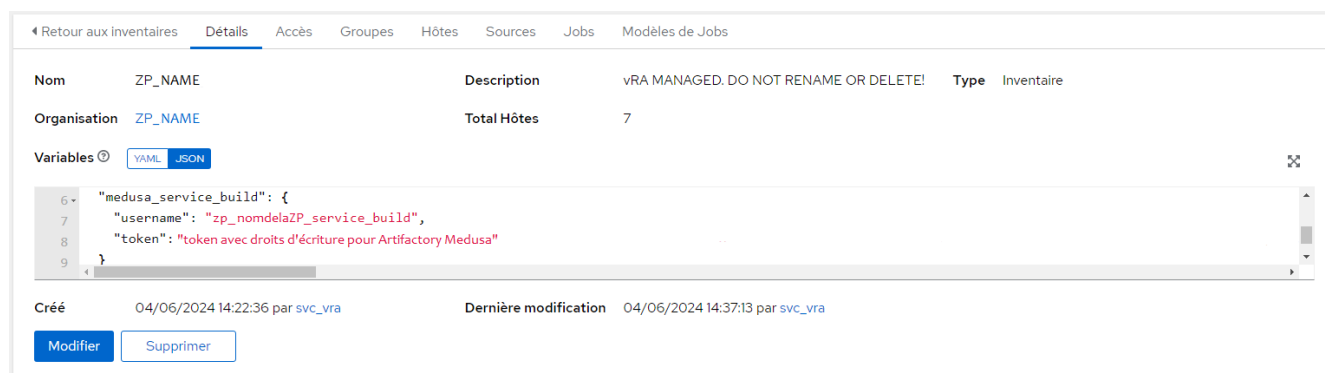


Figure 12 : Récupération du token Medusa

Ce token correspond à la valeur de la clé « artifactory_password » et doit être renseigné sur GDS.

secrets / se-kv-v2 / create

Create Secret

☐ JSONThis secret will be created in the `administration/ZP_NAME` namespace.

Path for this secret

`ansible`

Secret data

`artifactory_password`

TOKEN PRESENT DANS L'INVENTAIRE ZP_NAME DE AAP



Add

[Show secret metadata](#)

Save

Cancel

Figure 13 : Ajout des secrets dans Vault

3.2.2 Ajouter les secrets d'observabilité

La Zone Projet dans laquelle la promotion du SI va avoir lieu doit disposer d'un tenant Observabilité et avoir pris connaissance de ses identifiants (login / token), avec des droits en **lecture et en écriture**.

Aujourd'hui, la création de ce tenant et de ces identifiants n'étant pas automatisée, ils doivent être demandés via le Service Desk à l'URL suivante :

<https://forge.intradef.gouv.fr/plugins/tracker/?tracker=11745&func=new-artifact>

Ce token doit ensuite être intégré dans GDS, dans un dossier « observability » sous le nom « obs_apikey_elasticsearch_tenant ». La Figure 14 illustre l'ajout de ce secret.

secrets / se-kv-v2 / create

Create Secret

☐ JSONThis secret will be created in the `administration/ZP_NAME` namespace.

Path for this secret

Secret data

**Add**[Show secret metadata](#)**Save**

Cancel

Figure 14 : Ajout des secrets Observabilité

Pour modifier un secret dans l'outil GDS, il est nécessaire de cliquer sur le bouton « **Create new version** » (voir Figure 15).

secrets / se-kv-v2 / observability

observability

[Secret](#) [Metadata](#) [Paths](#) [Version History](#)☐ JSON

Delete

Destroy

Copy ▾

Version 3 ▾

Create new version +

Figure 15 : Modification de secret dans GDS Vault

3.3 Utiliser les secrets Vault

Une fois ajoutés dans l'outil GDS, les secrets peuvent être récupérés et utilisés par Ansible à travers le rôle `vault_get_secret` de la collection `minarm.gds`. Cette partie est détaillée plus loin dans la documentation dédiée à la création des playbooks Ansible (chapitre 5 « Déployer son projet grâce à Ansible »).

4. Spécificités des environnements de la plateforme C1DR

4.1 La création des machines virtuelles sur C1DR

Sur C1DR les DA/DP ne sont pas pourvues de droits permettant de créer ou de détruire des VMs. Celles-ci seront mises à votre disposition à partir des informations que vous aurez renseignées dans votre demande d'hébergement.

Il vous sera tout de même possible de faire des snapshots, de restaurer vos VM ainsi que de les éteindre ou de les redémarrer via VRA : <https://portail-vra.intradef.gouv.fr>.

4.2 Configurer son déploiement AAP sur C1DR

4.2.1 L'authentification Galaxy sur C1DR

Lors de la configuration de votre modèle de déploiement sur AAP il n'est pas nécessaire de créer une authentification Galaxy, celle-ci sera automatiquement générée pour vous.

Vos droits ne vous permettront pas de vérifier sa présence au sein de votre organisation, toutefois si la task suivante :

```
TASK [Fetch galaxy collections from collections/requirements.(yaml/yaml)]
```

tombe en erreur il est très probable que l'association entre votre organisation et le jeton Galaxy ai été mise en échec, vous pouvez alors faire un ticket à notre équipe support via le lien suivant : <https://forge.intradef.gouv.fr/plugins/tracker/?tracker=8599&func=new-artifact>.

Ceci est une première différence notable comparativement à la configuration nécessaire sur PICSEL ([cf. 2.1 Lier l'authentification Galaxy](#)).

4.2.2 Faire le lien avec la machine virtuelle sur C1DR

De la même manière que l'authentification Galaxy est ajoutée automatiquement à votre organisation, c'est également le cas pour la configuration de la communication SSH vers vos VMs. L'« informations d'identification » type **Machine** aura été créée au préalable pour vous et prend la forme **SVC_C1DR_P_VA_<TRI>_<ENV1L>**. Il ne vous restera plus qu'à l'ajouter manuellement à votre modèle. ([cf. faire le lien avec la machine virtuelle](#)).

4.2.3 Créer son projet AAP sur C1DR

Lors de la création de votre projet AAP en vue d'un déploiement sur C1DR le processus est très similaire à celui appliqué pour un déploiement PICSEL ([cf. 2.4 Créer son projet AAP](#)). La seule différence notable réside dans le choix des "Options" du projet : il vous faudra cocher toutes les cases à l'exception de « **Autoriser le remplacement de la branche** » ET de «

Supprimer ». Si la case de « **Supprimer** » est cochée la synchronisation du projet sortira en erreur.

4.2.4 Créer son modèle de job à lancer sur C1DR

Les différences connues pour le paramétrage d'un modèle AAP dans les environnements de la plateforme C1DR concernant les 2 « **Informations d'identification** » ([cf. 2.6 Créer son modèle de job à lancer](#)) :

- La **connexion SSH** est générée par défaut à la création de votre Zone Projet, il ne vous reste qu'à l'ajouter à votre modèle.
- La **connexion cloud** de catégorie *GDS DEV ARN Hashicorp Vault AppRole authentication auto* OU *GDS Hashicorp Vault AppRole authentication auto* **n'existe pas**. En effet, aucun service de Gestion des Secrets n'est actuellement mis à disposition des DA/DP sur les environnements de C1DR (nous évoquerons des manières d'y pallier dans le chapitre suivant).

Attention : Lorsque le service de GDS aura été intégré à la plateforme C1DR cette documentation sera mise à jour afin de vous permettre de vous aligner avec les attendus de la plateforme en termes de sécurité.

4.3 La gestion des secrets et des certificats sur C1DR

A l'heure actuelle la plateforme C1DR ne dispose pas d'un service de Gestion des Secrets équivalent à celui proposé sur PICSEL.

Dans l'attente de son arrivée il convient que les DA/DP organise par elle-même leur gestion des secrets. Vous pouvez vous renseigner sur les différentes solutions existantes (par exemple Ansible Vault la solution de GDS intégrée à Ansible, ou encore votre propre serveur HashiCorp Vault).

Concernant la gestion des certificats une méthode de contournement est proposée pour une « installation manuelle », voir ci-après :

- Pour passer la Promotion du SI (PICSEL), vous pouvez jouer le playbook **certificats_selfsigned.yml** (cf. ci-après).

Ce playbook utilise la collection : community.crypto version 2.5.0 (rôles : openssl_privatekey, openssl_privatekey, x509_certificate) pour générer des certificats ssl auto-signés.

Ou vous pouvez dès le départ utiliser le playbook **install_certificats.yml** (cf. ci-après) pour une installation manuelle.

L'utilisation d'une méthode plutôt que l'autre peut être indiqué via la variable : **certificats_selfsigned** (booléen true/false).

- Une fois arrivé sur C1DR, il faut FORCEMENT utiliser la méthode **mode manuel** via le playbook **install_certificats.yml**. Ce playbook ne s'exécute que si la variable **certificats_selfsigned** est déclarée à false.

*Sur C1DR il n'est pas possible d'auto-signer des certificats. C'est le POSA (**Autorité de certification IGCG-NG**) qui fournit des certificats, à demander lors de la demande d'hébergement, et il faut ensuite les intégrer manuellement au SI pour qu'il se déploie en https.*

Les certificats sont renseignés via les variables **content_cert_ssl** et **content_key_ssl**.

- Pour utiliser le mode manuel des certificats sur PICSEL, voici la démarche à suivre :

1 - Renseigner **certificats_selfsigned: false** dans le Modèle AAP utilisé.

2 - Sur la VM front de l'application, taper la commande suivante :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /tmp/<SI_NAME>.key -out /tmp/<SI_NAME>-cert.crt
```

3 - Remplir les informations demandées par openssl : La valeur de **Common Name** est importante à renseigner, celle-ci doit correspondre à l'ensemble **{{ nom_si }}.{{ zone_url }}**. Sachant que la valeur de la variable **{{ nom_si }}** correspond à l'alias DNS relié à notre VM front.

4 - Sur la VM front de l'application, taper la commande suivante :

```
cat /tmp/<SI_NAME>-cert.crt
```

5 - Copier le contenu du fichier **<SI_NAME>-cert.crt** et le coller dans la variable **content_cert_ssl** (à renseigner dans l'inventaire AAP).

6 - Sur la VM front de l'application, taper la commande suivante :

```
cat /tmp/<SI_NAME>.key
```

7 - Copier le contenu du fichier <SI_NAME>.key et le coller dans la variable **content_key_ssl** (à renseigner dans l'inventaire AAP).

A la fin, la partie **Variables** du Modèle AAP doit ressembler à ceci :

```
artifactory_password: "VALEUR"
db_password: "VALEUR"
certificats_selfsigned: false
content crt_ssl : |
-----BEGIN CERTIFICATE-----
VALEUR

content_key_ssl : |
-----BEGIN PRIVATE KEY-----
VALEUR
```

Attention : Attention à bien conserver l'indentation des fichiers .crt et .key en utilisant le `|` en début de chaîne ; cela peut causer des problèmes lors du restart d'Apache.

- Selfsigned_certificats.yml

```
---
- name: Configuration selfsigned certificats Apache

  block:

    - name: Ensure folder certs exist
```

```
ansible.builtin.file:

  path: /etc/pki/tls/certs

  state: directory

  recurse: true

- name: Ensure folder private exist

ansible.builtin.file:

  path: /etc/pki/tls/private

  state: directory

  recurse: true

- name: Create private key

community.crypto.openssl_privatekey:

  path: "{{ apache_key_path }}"

- name: Create content certificate

community.crypto.openssl_csr_pipe:

  privatekey_path: "{{ apache_key_path }}"

  common_name: "{{ nom_si }}.{{ zone_url }}"

  organization_name: "DSO"

  register: csr

- name: Create certificat

community.crypto.x509_certificate:

  path: "{{ apache_cert_path }}"
```

```
csr_content: "{{ csr.csr }}"
privatekey_path: "{{ apache_key_path }}"
provider: selfsigned
...
```

- Certificats.yml

```
---

- name: Configuration certificats Apache mode manuel
  block:

    - name: Ensure folder certs exist
      ansible.builtin.file:
        path: /etc/pki/tls/certs
        state: directory
        recurse: true

    - name: Ensure folder private exist
      ansible.builtin.file:
        path: /etc/pki/tls/private
        state: directory
        recurse: true

    - name: Certificats crt
      ansible.builtin.copy:
        content: "{{ content_crt_ssl }}"
```

```
dest: "{{ apache_cert_path }}"

owner: root

group: root

mode: '0644'

- name: Certificats key

  ansible.builtin.copy:

    content: "{{ content_key_ssl }}"

    dest: "{{ apache_key_path }}"

    owner: root

    group: root

    mode: '0644'
```

Attention : Lorsque le service de GDS aura été intégré à la plateforme C1DR cette documentation sera mise à jour afin de vous permettre de vous aligner avec les attendus de la plateforme en termes de sécurité.

Conclusion

Vous avez terminé le chapitre 4 « Déployer avec AAP et stocker ses secrets sur GDS ». Grâce à celui-ci, vous avez pu remplir les objectifs suivants :

Progression	Titre de la partie	Description
✓	Mettre en place sa première machine virtuelle	Déployer une machine virtuelle avec VRA
✓	Paramétrer son déploiement avec Ansible Automation Platform	Comment utiliser AAP ? Comment lier son projet à cet outil ? Comment jouer ses playbooks ?
✓	Configurer ses secrets au sein de Vault GDS	Comment gérer ses secrets sur Vault GDS ?