

NON PROTEGE



**Direction générale
de l'armement**

DOCUMENTATION 6

TESTER ET PROMOUVOIR SON SI

1/24

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. Valider son projet	6
1.1 Se connecter au SI	6
1.2 Accéder au dashboard Kibana	6
1.3 Tester son SI	7
2. Vérifier le respect des prérequis à la promotion	8
2.1 Récupérer les informations nécessaires à la promotion	8
2.2 Déployer son infrastructure avec Terraform.....	10
2.3 Créer un Gitlab Runner	11
3. Promouvoir son SI	13
3.1 Comprendre Gitlab CI	13
3.2 Récupérer le fichier .gitlab-ci.yml.....	14
3.3 Mettre en place la cartographie des rôles et collections	14
3.4 Variables obligatoires.....	15
3.5 Lancer la promotion du SI	18
4. Vérifier le bon déroulement du pipeline et générer le package applicatif	19
4.1 Vérifier le manifest	19
4.2 Vérifier le rapport qualité.....	19
4.3 Pipeline de packaging.....	20
5. Et après ?	21
5.1 Livrables à fournir	21
5.2 Après la livraison	22
Conclusion.....	24

Historique

Tableau 1 : Historique

Version	Date	Auteur	Commentaire
1.0.0	10/09/2024	Cindy PEREIRA	
1.0.1	11/09/2024	Chloé PAGENEL	Relecture
1.0.2	12/09/2024	Cindy PEREIRA	Relecture
1.0.3	07/10/2024	Cindy PEREIRA	Correction de coquilles

Acronymes

Tableau 2 : Acronymes

Acronyme	Description
CLI	Command Line Interface
GDS	Gestion Des Secrets
IHM	Interface Homme Machine
VM	Machine Virtuelle
ZP	Zone Projet

Introduction

L'objectif de ce **chapitre 6** est de parcourir un projet exemple pour appréhender les différentes étapes de la « Promotion de votre SI », dernière étape clé pour valider votre déploiement sur PICSEL.

Ce chapitre va donc commencer par quelques étapes de vérification pour s'assurer que votre SI est bien déployé avec Ansible et qu'il a le comportement attendu.

Ensuite, vous vérifierez que tous les prérequis pour la Promotion de votre SI ont été respectés.

Puis, vous verrez comment promouvoir votre SI au travers de l'utilisation d'un pipeline Gitlab CI/CD.

Enfin, vous vous assurerez que la Promotion de votre SI a été correctement réalisée de bout en bout.

Et finalement, ce chapitre évoquera rapidement les événements qui suivront après la validation de la Promotion de votre SI.

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

Charge	Titre de la partie	Description
10%	Valider son projet	Est-ce que son projet est correctement déployé ?
10%	Vérifier le respect des prérequis à la Promotion	Est-ce que son projet respecte les prérequis permettant de jouer la Promotion ?
60%	Promouvoir son SI	On fait la Promotion du SI
10%	Vérifier la Promotion	On vérifie que la promotion a bien fonctionné
10%	Et après ?	Quels sont les livrables à fournir et quelle est la suite des événements ?

1. Valider son projet

L'objectif de cette partie est de vérifier le bon fonctionnement du SI déployé grâce à AAP.

1.1 Se connecter au SI

La première étape est de se connecter au SI pour vérifier que les pages s'affichent tel qu'attendu.

Pour cela :

- Connectez-vous à la Zone Projet du SI puis entrez l'URL suivante afin d'accéder au SI :

`https://<dns_name_vm_front>.picsel.defense.gouv.fr`

- Naviguez entre les pages pour vérifier que tout fonctionne correctement.
- Vérifiez que les pages de connexion sont bien redirigées vers MinDefConnect et qu'il est possible de vous y connecter.

1.2 Accéder au dashboard Kibana

La validation du déploiement du SI se fait également par les logs Kibana. L'interface Kibana est accessible sur PICSEL au lien suivant :

<https://kibana.observ.picsel.defense.gouv.fr/login?next=%2F>

Pour cela :

- Aller sur la Zone Projet spécifique au SI. Cliquer sur l'onglet **Analytics** puis sur **Discover**.
- Choisir la durée sur laquelle les logs doivent s'afficher (voir Figure 1).
- Vous devriez voir un graphique de logs s'afficher.

Il faut également vérifier les informations de l'API Observabilité :

- Dans Kibana, choisir l'espace « **Commun** ».
- Cliquer sur Analytics puis Dashboard.
- Chercher le Dashboard **[API Observabilité] Vue d'ensemble d'un simple Système d'Information**.
- Si l'enrôlement est bien effectué, vous devriez alors pouvoir choisir votre projet dans la section « Nom du projet ».
- Vous pourrez ainsi vérifier que votre SI est « UP ».
- Si aucune information ne s'affiche et que le service indique « Unknown », c'est que l'API Observabilité n'a pas été mise en place correctement.



Figure 1 : Vérification des logs Kibana

1.3 Tester son SI

Si vous avez mis en place des protocoles de tests fonctionnels, vous pouvez les jouer pour voir si toutes les briques de votre SI ont bien été déployées. Vous pouvez également vérifier que votre base de données est complétée avec les données attendues.

2. Vérifier le respect des prérequis à la promotion

L'objectif de cette partie est de mettre en place toutes les étapes utiles pour pouvoir faire la Promotion de votre SI.

Il vous sera donc d'abord indiqué comment obtenir tous les **tokens** utiles, comment déployer votre infrastructure avec **Terraform** et enfin où obtenir votre **Gitlab Runner**.

2.1 Récupérer les informations nécessaires à la promotion

Plusieurs tokens sont nécessaires pour lancer le processus de promotion du SI. Cette partie liste les informations à récupérer avant de pouvoir lancer les pipelines.

- **LOGIN_BRM** (username) et **PWD_BRM** (token) : Ces deux valeurs (identifiants d'accès à Medusa) sont initialisées automatiquement lors de la création de la ZP, elles sont référencées dans un inventaire AAP portant le nom de la ZP (cf Figure 2).

```
"medusa_service_build": {  
  "username": "zp_..._service_build",  
  "token": ...  
},
```

Figure 2 : Récupération des identifiants Medusa

Plusieurs identifiants sont présents dans l'inventaire. **Il est nécessaire d'utiliser le compte de service build**, afin de posséder des accès en lecture et en écriture.

- **VRA_USERNAME** : Le username d'un compte nominatif ayant accès à la ZP.
- **VRA_REFRESH_TOKEN** : Sa durée de vie est de 90 jours. Pour l'obtenir ou le mettre à jour, lancer la commande suivante depuis une VM de la ZP concernée, en remplaçant les username et password avec les identifiants d'un compte nominatif ayant accès à la Zone Projet :

```
api_token=`curl -k -X POST \  
"https://cloud.picsel.defense.gouv.fr/csp/gateway/am/api/login?access_token"  
\  
-H 'Content-Type: application/json' \  
-d '{  
  "username": "p.nom",  
  "password": "xxxxx"  
}' | jq -r .refresh_token`
```

Après avoir lancé la commande, taper `echo $api_token` pour obtenir le token.

- **RUCHE_API_KEY** : à récupérer au lien suivant : <https://portail-hebergement.intradef.gouv.fr/fr/normes-minarm/automatisation-deploiement>. Cette variable permet d'accéder aux collections RUCHE.
- **XRAY_TOKEN** : Un token Xray doit être demandé par ticket sur le Service Desk PICSEL (URL : <https://forge.intradef.gouv.fr/plugins/tracker/?tracker=10282&func=new-artifact>).
- **MANIFEST_GITLAB_TOKEN** : Jeton d'accès au projet GitLab, il doit avoir des droits en écriture (= rôle au moins 'developer' et scope 'write_repository'). Ce jeton peut être créé dans la section « Access Token » des Settings de Gitlab.
- **VM_USER** et **VM_PASSWORD** : Identifiants d'accès aux VM. Ces identifiants peuvent être trouvés sur la page suivante : [https://portail-ct-rns.intradef.gouv.fr/sites/Portail%20du%20Cloud/KB_PICSEL/Pages%20Wiki%20PICSEL/Service%20de%20gestion%20des%20Machines%20Virtuelles%20\(VM\).aspx](https://portail-ct-rns.intradef.gouv.fr/sites/Portail%20du%20Cloud/KB_PICSEL/Pages%20Wiki%20PICSEL/Service%20de%20gestion%20des%20Machines%20Virtuelles%20(VM).aspx)
- **QUALITY_SERVICE_TOKEN** : Jeton d'accès à Observabilité avec des droits en **lecture ET en écriture**. La demande d'un jeton d'accès s'effectue par ticket sur le Service Desk OBSERVABILITE (URL : <https://forge.intradef.gouv.fr/plugins/tracker/?tracker=11748&func=new-artifact>). Correspond à la valeur de **obs_apikey_elasticsearch_tenant** dans la réponse du ticket.
- **VAULT_ROLE_ID** et **VAULT_SECRET_ID** : Token d'accès à Vault. Pour les obtenir, il faut se connecter sur l'IHM de Vault accessible par VDI (<https://gds.picisel.defense.gouv.fr/>) avec un compte administrateur de ZP, et ouvrir la CLI (avec l'icône en haut à gauche). Dans la CLI, rentrer les commandes suivantes :
`vault read /auth/approle/role/tower-admin/role-id` pour obtenir le **role_id**
`vault write -force /auth/approle/role/tower-admin/secret-id` pour obtenir le **secret_id**

Il est également nécessaire de vérifier votre version du **VAULT_PKI_ENGINE**. Pour ce faire, suivre les étapes suivantes :

- Aller sur l'IHM Vault : <https://gds.picisel.defense.gouv.fr/> et se connecter sur la Zone Projet (administration/ZP_NAME)
- Aller dans l'onglet "Secrets Engines" et récupérer l'AC émettrice Vault la plus récente (**se-pki-ace-202x-x**). Cette valeur correspond à la variable **VAULT_PKI_ENGINE**.

2.2 Déployer son infrastructure avec Terraform

Terraform est un outil permettant de définir, provisionner et gérer des infrastructures. Cet outil est utilisé lors du service de Promotion du SI.

Un ensemble de scripts template Terraform assurant le déploiement d'une infrastructure sur PICSEL est disponible dans l'exemple de notre déploiement Drupal, au lien suivant :

https://scm-intradef.picisel.defense.gouv.fr/ZS_CONTINUITY/samples/exemples-documentation-4-move2cloud/exemple-partie-promotion-du-si

Vous trouverez tous les fichiers nécessaires dans le dossier racine **/terraform**.

Il est adaptable à tous les SI et ne nécessite aucune compétence spécifique sur Terraform.

Seule l'affectation des variables est à considérer par l'utilisateur.

Pour ce faire, des indications sont directement intégrées en commentaires dans les fichiers à modifier.

Les fichiers à récupérer sont les suivants :

- **common.auto.tfvars** : modifier la variable **project_zone** avec le nom de votre Zone Projet.
- **infrastructure.tf**
- **terraform.tf**
- **variables.tf**
- **review.env.tfvars** : le nom de ce fichier doit correspondre à l'environnement dans lequel l'infrastructure sera déployée (integration, review, staging...).
- **hosts.tftpl**

Attention : Les fichiers sont donnés à titre d'exemple. Il est nécessaire de modifier le fichier **review.env.tfvars** avec les valeurs des déploiements spécifiques à votre SI. De même, vous devez absolument modifier la variable **project_zone** avec le nom de votre Zone Projet.

Il vous est demandé de mettre au minimum le trigramme de votre SI dans les noms de vos déploiements (variable **deployment_name**). Par exemple, ZP_M2C_VM_BACK.

2.3 Créer un Gitlab Runner

GitLab Runner est une application qui fonctionne avec GitLab CI/CD pour exécuter des tâches dans un pipeline. Avoir un GitLab Runner en fonctionnement dans sa Zone Projet est nécessaire pour la Promotion du SI.

La création d'un GitLab Runner se fait dans le catalogue VRA avec la carte « GitLab-Runner » : <https://cloud.picsel.defense.gouv.fr/automation/#/service/catalog/consume/library/1935bdb8-23a4-3510-b6d6-e80d92b6a2b9>

Les informations rentrées par défaut ne doivent pas être modifiées (voir Figure 3). Notamment, il est nécessaire de laisser l'executor en « docker ».

Une fois le Runner créé, il s'affiche automatiquement dans la section Runners de CI/CD dans votre projet Gitlab (URL de la forme : https://scm-intredef.picsel.defense.gouv.fr/groups/ZP_NAME/-/runners).

GitLab-Runner Version Q 2.2.1

GITLAB RUNNER

[Documentation](#)

Attention, il est nécessaire d'avoir un tenant dans GitLab avant d'utiliser cette carte pour que le runner y soit rattaché.

Informations pour vRA

Project * ZP_MOVE2CLOUD

Deployment Name * Gitlab runner

Options pour le runner

Executor docker

Taille du disque en Go 32

Tags ci, vRAGenerated

SUBMIT CANCEL

Figure 3 : Création d'un Gitlab runner

3. Promouvoir son SI

L'objectif de cette partie est de faire concrètement la Promotion du SI.

Ainsi, une fois que vos playbooks ont été testés et validés sur AAP, que les fichiers Terraform permettant de créer des machines virtuelles automatiquement ont été ajoutés dans votre projet Ansible, il est temps de passer à la Promotion de votre SI.

La Promotion du SI s'effectue par Gitlab CI en utilisant des templates de pipelines mis à disposition sur PICSEL.

3.1 Comprendre Gitlab CI

Gitlab CI (Continuous Integration) est une fonctionnalité de Gitlab qui permet d'automatiser le processus d'Intégration Continue.

La Promotion du SI s'effectue donc par Gitlab CI en utilisant des templates de pipelines mis à disposition dans l'exemple de notre déploiement Drupal, au lien suivant : **https://scm-intradef.picisel.defense.gouv.fr/ZS_CONTINUITY/samples/exemples-documentation-4-move2cloud/exemple-partie-promotion-du-si**

Un fichier nommé **.gitlab-ci.yml** doit être présent à la racine du projet Git. Ce fichier décrit les étapes et les jobs à exécuter, ainsi que les conditions et les environnements nécessaires.

Dans notre cas, il permettra d'importer les différents projets qui constituent le pipeline de la Promotion du SI.

Un **pipeline** est une séquence de stages qui décrit tout le processus d'intégration et de déploiement continu.

Un **stage** (ou *étape* en français) est une phase du pipeline dans laquelle des jobs sont exécutés. Au sein d'un stage, les jobs s'exécutent en parallèle. Les stages s'exécutent dans l'ordre défini par le fichier **.gitlab-ci.yml**.

Un **job** (ou *tâche* en français) est une tâche à effectuer dans le pipeline. Chaque job contient un ensemble de commandes que Gitlab CI doit exécuter.

Le mot-clé **include** permet d'inclure des projets de pipelines déjà créés. Ce mot-clé doit être présent une fois, avant la liste de projets à importer.

3.2 Récupérer le fichier .gitlab-ci.yml

Un fichier **.gitlab-ci.yml** contenant tous les imports de templates et stages nécessaires est mis à votre disposition dans la ZS_CONTINUITY à l'URL suivante : **https://scm-intradef.picsele.defense.gouv.fr/ZS_CONTINUITY/samples/exemples-documentation-4-move2cloud/exemple-partie-promotion-du-si**

Ce fichier référence également les variables à ajouter (voir Tableau 3 : Variables obligatoires pour la promotion du SI).

Il est nécessaire de faire une copie de ce fichier à la racine de votre projet Gitlab et de modifier les valeurs de ces variables, ainsi que d'ajouter en variables d'environnement celles qui doivent l'être.

3.3 Mettre en place la cartographie des rôles et collections

Une étape du pipeline de Promotion est de récupérer et publier des métriques concernant le SI. Il est nécessaire de fournir un rapport qualité en version PDF. Dans ce rapport se trouve notamment la cartographie des rôles qui permet de lister tous les rôles utilisés dans le projet.

Quelques étapes sont à suivre afin d'obtenir la cartographie des rôles :

- Récupérer le fichier **/callback_plugins/dependency.py** du projet exemple et le placer dans le dossier **/callback_plugins** de votre projet.
- Récupérer le fichier **extra_info.yml** du projet exemple et le placer à la racine du projet. Remplir les informations nécessaires.
- Remplir le fichier **ansible.cfg** : Ce fichier doit être placé à la racine de votre projet et doit contenir la ligne suivante :

```
[defaults]
callbacks_enabled = dependency
```
- Ajouter les variables suivantes dans la section **ANSIBLE_SI_EXTRA_VARS** du fichier **.gitlab-ci.yml** (voir Figure 4) :
 - o **quality_service_token** = Jeton d'accès à Observabilité avec des droits en lecture ET en écriture.
 - o **quality_service_tenant** = Nom de la ZP sous la forme ZP_NAME
 - o **quality_service_si** = Nom du SI
 - o **quality_service_si_version** = Version du SI
 - o **quality_service_team** = Nom de l'équipe
 - o **send_data_elastic** = true

Astuce : Mettre les variables à cet endroit permet de ne déclarer l'information qu'une seule fois et ne pas avoir à la surcharger.

```
ANSIBLE_SI_EXTRA_VARS: |
  quality_service_token=$QUALITY_SERVICE_TOKEN
  quality_service_si_version=$VERSION_SI
  quality_service_tenant=$QUALITY_SERVICE_TENANT
  quality_service_si=$QUALITY_SERVICE_SI
  quality_service_team=$QUALITY_SERVICE_TEAM
  send_data_elastic=true
```

Figure 4 : Variables de cartographie des rôles

3.4 Variables obligatoires

Les variables suivantes doivent obligatoirement être ajoutées dans votre projet Ansible sur Gitlab.

Celles dont la valeur de la case « Masqué ? » est « Non » doivent être placées dans le fichier **.gitlab-ci.yml**.

Celles dont la valeur de la case « Masqué ? » est « Oui » doivent être placées dans les variables d'environnement CI/CD.

Pour ce faire :

- Aller dans l'onglet **CI/CD** de **Settings** de Gitlab.
- Dans la section Variables, cliquer sur **Expand**.
- Cliquer sur **Add variable** pour ajouter une nouvelle variable.

Ces variables étant des tokens, il est important de cocher l'option « **Mask variable** » pour qu'ils ne s'affichent pas dans les logs.

Tableau 3 : Variables obligatoires pour la promotion du SI

Nom de la variable	Description	Exemple de valeur	Masqué
ADAPTIVE_PIPELINE_DISABLED	Permet d'activer le stage quality dans le pipeline	true	Non

TENANT_LOWER	Nom de la ZP en minuscules	zp_move2cloud	Non
MANIFEST_GITLAB_TOKEN	Jeton d'accès Gitlab	xxxxxxxxxxxx	Oui
PWD_BRM	Token Medusa	xxxxxxxxxxxx	Oui
LOGIN_BRM	Username Medusa	zp_m2c_service_build	Non
VAULT_PKI_ENGINE	Nom du moteur "se-pki-ace" le plus récent sur HashiCorp Vault GDS	se-pki-ace-2024-2	Non
VAULT_ROLE_ID	Identifiant d'accès à HashiCorp Vault GDS	xxxxxxxxxxxx	Oui
VAULT_SECRET_ID	Mot de passe HashiCorp Vault GDS	xxxxxxxxxxxx	Oui
RUCHE_API_KEY	Token d'accès à l'Automation Hub Ruche pour récupération des collections	xxxxxxxxxxxx	Oui
VERSION_SI	Version du SI, à incrémenter à chaque relivraison	1.0.0.0	Non
VM_USER	Username de la machine virtuelle	sysadm	Non
VM_PASSWORD	Mot de passe de la machine virtuelle	xxxxxxxxxxxx	Oui
VRA_USERNAME	Username d'un compte ayant accès à la ZP	c.pereira	Non

SI_TF_VARS_FILE	Nom du fichier de déploiement Terraform	review.env.tfvars	Non
VRA_REFRESH_TOKEN	Token VRA (possédant une date d'expiration)	xxxxxxxxxxxx	Oui
VRA_HOSTNAME	FQDN de VRA	cloud.picisel.defense.gov.fr	Non
QUALITY_SERVICE_TOKEN	Jeton d'observabilité	xxxxxxxxxxxx	Oui
QUALITY_SERVICE_TEAM	Équipe en charge du SI	Move2Cloud	Non
QUALITY_SERVICE_SI	Nom du SI	Move2Cloud	Non
QUALITY_SERVICE_SI_VERSION	Version du SI	\$VERSION_SI	Non
FQDN_BRM_SERVER	URL vers Medusa – à mettre dans l'inventaire all.yml	"medusa.{{ zone_url }}"	Non
XRAY_TOKEN	Token XRAY	xxxxxxxxxxxx	Oui
PLATFORME_DESTINATION	Plateforme sur laquelle le SI sera déployé	C1NP	Non
CI_USER_MAIL	Email de l'utilisateur qui apparaîtra comme ayant fait la livraison	adresse@email.fr	Non
CI_USER_NAME	Nom de l'utilisateur qui apparaîtra comme ayant fait la livraison	Prénom NOM	Non

INVENTAIRE_SI_VMS	Chemin vers le fichier hosts.yml	inventories/review/hosts.yml	Non
-------------------	----------------------------------	------------------------------	-----

3.5 Lancer la promotion du SI

3.5.1 Pipeline de génération du manifest

Pour lancer le pipeline de génération du manifest, allez dans l'onglet CI/CD de Gitlab.

Cliquer sur le bouton « **Run pipeline** » et choisir la branche à partir de laquelle le pipeline doit être lancé. Cliquer sur Run pipeline sans renseigner de variables supplémentaires (voir Figure 5).



Figure 5 : Lancer un pipeline sur Gitlab

3.5.2 Que faire si le pipeline est en échec ?

En cas d'erreur du pipeline sur une tâche située entre le stage « infra » et le stage « infra-destroy », il est nécessaire de supprimer manuellement toutes les VM qui ont été créées par Terraform avant de relancer le pipeline. Pour ce faire, aller dans la section Deployments du Service Broker :

<https://cloud.picsel.defense.gouv.fr/automation/#/service/catalog/consume/deployment>.

Cliquer sur les trois points verticaux au niveau des VM concernées et cliquer sur Delete. Cette action peut prendre quelques minutes avant d'être effective.

4. Vérifier le bon déroulement du pipeline et générer le package applicatif

L'objectif de cette partie est de vérifier que la Promotion du SI a fonctionné correctement de bout en bout. Nous allons donc nous assurer que le manifest et le PDF qualité générés par le pipeline sont complets.

4.1 Vérifier le manifest

Le pipeline de génération du manifest contient une tâche permettant d'intégrer un fichier **manifest.yml** à votre projet Ansible sur Gitlab. Lorsque celui-ci apparaît, il est important de vérifier qu'il contient bien les informations nécessaires.

Attention : Le job **push_manifest** est autorisé à fail si le fichier **manifest.yml** qu'il essaie de push est identique à celui déjà présent dans votre projet. Cela n'impacte pas la suite du pipeline.

Ce fichier doit contenir les champs suivants :

- version : version de votre SI.
- gitlab : peut être vide, cela correspond aux rôles externes au projet éventuellement utilisés.
- artifactory : liste de tous les fichiers utiles au déploiement contenus sur Medusa
- decos : liste des fichiers contenus dans Decos

4.2 Vérifier le rapport qualité

Une fois le pipeline de génération du manifest passé, le job **create-final-report** devrait avoir créé un rapport qualité en PDF.

Pour le récupérer, cliquer sur le job en question. Dans la section **Job artifacts**, le bouton Download permet de télécharger l'artefact.

Attention : Les artefacts s'autodétruisent au bout de 24h. Veillez à télécharger le rapport PDF dès que le pipeline a tourné.

Ce rapport doit contenir les choses suivantes :

- Informations générales sur votre SI
- Liste de toutes les collections utilisées
- Vulnérabilités XRAY de votre SI (liste et graphiques)

4.3 Pipeline de packaging

Une fois que le manifest généré est complet, il est nécessaire de faire tourner le pipeline qui permettra de créer l'archive de votre SI et de la déposer sur Medusa.

Pour ce faire, suivre les étapes suivantes :

- Créer un tag à partir de la branche sur laquelle le pipeline a tourné.
- Aller dans Repository > Tags et cliquer sur **New tag**.
- Nommer le tag avec la version du SI (par exemple 1.0.0.0) et choisir la branche en question dans « Create from ».
- Aller dans l'onglet CI/CD et cliquer sur **Run pipeline**. Choisir à ce moment-là le tag nouvellement créé et cliquer sur Run pipeline sans renseigner de variables supplémentaires.

Lorsque le pipeline de packaging tourne, il crée une archive contenant tous les fichiers listés dans le manifest. Cette archive est déposée sur Medusa, dans le dossier **export_livraison** du répertoire zp_name-generic.

[2FDSI%2DSocle%2DNumerique%5FMove%2Dto%2DCloud%2FRessourcesdocumentaires%2FDocumentation%2DPr%C3%A9requis%2FPICSEL%2FPROMOTION%2DSI&FolderCTID=0x0120009E74175D7CF3B44DB1B7051D91BE8697&View=%7B58521187%2D2BFF%2D4A23%2D8FC6%2DA15B0BB26636%7D](https://portail-hebergement.intradef.gouv.fr/fr/documentaires%2FDocumentation%2DPr%C3%A9requis%2FPICSEL%2FPROMOTION%2DSI&FolderCTID=0x0120009E74175D7CF3B44DB1B7051D91BE8697&View=%7B58521187%2D2BFF%2D4A23%2D8FC6%2DA15B0BB26636%7D)).

5.1.4 Manuel d'exploitation

Le manuel d'exploitation doit décrire les procédures d'exploitation de votre SI. Il doit lister les différentes commandes à effectuer pour gérer votre SI.

Un template de Manuel d'exploitation est mis à votre disposition au lien suivant : <https://portail-hebergement.intradef.gouv.fr/fr/documentation>.

5.1.5 Document d'Architecture Technique

Le DAT contient ce qu'il faut mettre en place pour réussir la mise en œuvre du SI, et notamment pour les exploitant : les règles de flux inter-VM et la liste des ressources consommé sur Internet.

Un template de DAT est à télécharger sur le lien suivant : http://portail-hebergement.intradef.gouv.fr/sites/default/files/migrated_files/Word/DAT_Dossier_Architecture_Technique_du_SI-XXX-version-minimaliste.docx

5.2 Après la livraison

5.2.1 Installation sur C1NP

Une fois les livrables fournis à la TME, celle-ci va pouvoir déployer votre SI sur C1NP.

Pour ce faire, elle va devoir opérer des changements spécifiques à l'environnement dans le dossier **/inventories** de votre projet. C'est pourquoi il est nécessaire que votre projet suive la structure indiquée dans cette documentation, afin que l'équipe d'exploitation puisse facilement retrouver les variables à modifier.

5.2.2 Fourniture des certificats pour l'exposition publique

Il ne vous reste plus qu'à fournir aux exploitant les certificats SSL qu'ils devront appliquer lors de l'exposition en ligne de votre SI.

Pour les obtenir, si ce n'est pas déjà fait, vous devez en faire la demande.
Il s'agit d'une demande de type DSC « Site WEB – Acquisition de certificat SSL » à faire au POC concerné.

Plus d'informations sur ce lien :

https://catalogue-services-dirisi.intradef.gouv.fr/sp?sys_kb_id=508cf2e65d5a8e90d025c7b7b258bd55&id=kb_article_vie&sysparm_rank=1&sysparm_tsqueryId=73e45bd669ecde1085f38acb96e95a81

Conclusion

Vous avez terminé le chapitre 6 « Tester et promouvoir son SI ». Grâce à celui-ci, vous avez pu remplir les objectifs suivants :

Avancement	Titre de la partie	Description
✓	Valider son projet	Est-ce que mon projet est correctement déployer ?
✓	Vérifier le respect des prérequis à la promotion	Est-ce que mon projet respecte les prérequis permettant de jouer la promotion ?
✓	Promouvoir son SI	On fait la promotion de SI
✓	Vérifier la promotion	On vérifie que la promotion a bien fonctionné
✓	Et après ?	Quels sont les livrables à fournir et quelle est la suite des évènements ?