

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence de l'ouvrage **EP2ANS** dans la zone de recherche
et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

- 1. D'où vient le terme DevOps ? 21
- 2. Les premiers produits DevOps. 22
- 3. Cibles et objectifs de l'ouvrage 23
- 4. Prérequis techniques et ressources documentaires. 23
 - 4.1 Prérequis techniques. 23
 - 4.2 Ressources documentaires 23
- 5. Présentation générale 24
 - 5.1 Les prérequis 25
 - 5.2 L'utilisation d'Ansible 25
 - 5.3 La personnalisation d'Ansible 26
 - 5.4 Conventions employées 26

Chapitre 1

Démarrer avec Ansible

- 1. Objectifs du chapitre et prérequis 27
 - 1.1 Contexte et prérequis. 27
 - 1.2 Version de Python 28
 - 1.3 Environnement de travail. 28
 - 1.4 Fichiers téléchargeables 28
- 2. Installation d'Ansible 29
 - 2.1 Contexte 29
 - 2.2 Installation derrière un proxy 29

2 **Ansible**

Configuration des serveurs, déploiement des applications

2.3	Installation via les packages système	30
2.3.1	Installation sous Debian/Ubuntu	30
2.3.2	Installation sur RHEL, CentOS ou Fedora	30
2.4	Installation via pip	30
2.5	Utilisation des packages releases	31
2.6	Utilisation de virtualenv	33
2.7	Vérification de la version d'Ansible	34
3.	Le protocole SSH	35
3.1	À propos de SSH	35
3.2	Clé publique et clé privée	35
3.3	Génération de la clé	36
3.4	Étapes de l'authentification	37
3.5	Parc important de machines ou hébergement dans le cloud	38
3.6	Échange de clé par mot de passe	39
3.7	Échange de clé sans mot de passe	40
3.8	Gestion d'une passphrase avec Ansible	41

Chapitre 2

Utilisation d'Ansible

1.	Objectifs du chapitre et prérequis	43
1.1	Contexte et prérequis	43
1.2	Fichiers téléchargeables	43
2.	Ansible en mode ad hoc	44
2.1	Création d'un fichier d'inventaire	44
2.2	Utilisateur non root	45
2.3	Utilisateur SSH non root et mécanisme sudo	47
3.	Les autres outils d'Ansible : playbook et doc	50
4.	Quelques notions sur le format YAML	54
4.1	Déclaration de variables simples	55
4.2	Les tableaux en YAML	56
4.3	Les structures clé/valeur	57

- 4.4 Tableau de tables de hachage..... 58
- 4.5 Inventaire au format YAML 59
- 5. Introduction de la notion de playbook 60
 - 5.1 Structure d’un playbook 60
 - 5.2 Lancement d’un playbook 62
- 6. Combinaison avec Git 63
 - 6.1 Création du repository 63
 - 6.2 Commandes de base 64
 - 6.2.1 Récupération d’un repository Git 64
 - 6.2.2 Branche de travail 65
 - 6.2.3 État de votre repository 66
 - 6.2.4 Ajout d’un fichier 66
 - 6.2.5 Modification d’un fichier 67
 - 6.2.6 Mise à jour du repository distant..... 68
 - 6.2.7 Récupération des modifications distantes..... 68
 - 6.3 Mécanisme de hook 69
 - 6.3.1 Vérification des playbooks avec ansible-lint 70
 - 6.3.2 Création du hook pre-commit 70
 - 6.3.3 Test du fonctionnement du hook 71

Chapitre 3

Découverte de l’inventaire

- 1. Objectifs du chapitre et prérequis 73
 - 1.1 Contexte et prérequis..... 73
 - 1.2 Fichiers téléchargeables 73
- 2. Structure d’un inventaire 74
 - 2.1 Groupes par défaut..... 75
 - 2.2 Mode de connexion aux machines 76
 - 2.3 Regroupement de machines..... 77
 - 2.4 Variables d’inventaire..... 80
 - 2.5 Hiérarchie des variables 81

2.6	Gestion des différents inventaires	83
2.7	Création de groupes temporaires.	85
3.	Fusion d'inventaires	88

Chapitre 4 **Inventaires : notions avancées**

1.	Objectifs du chapitre et prérequis	89
1.1	Contexte et prérequis.	89
1.2	Fichiers téléchargeables	90
2.	Chiffrement de fichiers	90
2.1	Contexte	90
2.2	Stockage d'identifiants de connexion	90
2.3	Chiffrement du fichier entier.	91
2.3.1	Par mot de passe	91
2.3.2	Utilisation d'un fichier	92
2.3.3	Déchiffrement d'un fichier.	93
2.3.4	Changement du mot de passe de chiffrement	94
2.4	Chiffrement d'un champ	95
3.	Les inventaires dynamiques	96
3.1	Contexte	96
3.2	Machines AWS	97
3.2.1	Prérequis	97
3.2.2	Récupération du script ec2.py	97
3.2.3	Configuration	98
3.2.4	Test du script	98
3.3	Communication avec Docker	99
3.3.1	Récupération du script d'inventaire.	99
3.3.2	Configuration de la communication avec Docker	100
3.3.3	Test du script d'inventaire	100
3.3.4	Communication avec les conteneurs Docker	101
3.3.5	Un petit mot sur le choix de l'image Docker	102

- 3.4 Extraction d'informations en provenance d'ESX 103
- 3.5 Extraction d'informations de Nagios/Naemon/Shinken 106
- 4. Fonctionnement des plugins d'inventaires génériques 108
 - 4.1 Contexte 108
 - 4.2 Connecteur supporté par le format auto 109
 - 4.3 Exemples de fichiers d'inventaire au format auto 110
 - 4.4 Utilisation de la commande ansible-inventory 111
 - 4.5 Inventaire des instances ec2 112
- 5. Écrire son propre inventaire dynamique 113
 - 5.1 Contexte 113
 - 5.2 Format en entrée 114
 - 5.3 Structure du programme 114
 - 5.3.1 En-tête du programme 114
 - 5.3.2 Chargement du fichier 115
 - 5.3.3 Transformation et alimentation de l'inventaire 116
 - 5.3.4 Test du script d'inventaire 118
 - 5.3.5 Test de l'inventaire dynamique avec Ansible 119

Chapitre 5
Administration Windows

- 1. Objectifs du chapitre et prérequis 121
 - 1.1 Contexte et prérequis 121
 - 1.2 Fichiers téléchargeables 121
- 2. Ansible et les machines sous Windows 122
 - 2.1 Contexte 122
 - 2.2 Prérequis 122
 - 2.3 Configuration de la machine 123
 - 2.4 Création du fichier d'inventaire 124
 - 2.5 Test de la communication Windows 125
 - 2.6 Installation d'un package 126

3.	Gestion de service Windows	128
3.1	Contexte	128
3.2	Installation de Prometheus	128
3.3	Création du playbook.	129
3.4	Lancement de l'installation	131
3.5	Consultation de la console Prometheus	132
4.	Création de service	133
4.1	Installation de Grafana.	133
4.1.1	Présentation de Grafana	133
4.1.2	Création du playbook.	134
4.1.3	Lancement de l'installation	135
4.2	Création d'un service Windows.	136
4.2.1	Recherche du programme de lancement de Grafana	136
4.2.2	Gestionnaire de services NSSM	137
4.2.3	Création du service Grafana.	137
4.2.4	Playbook complet d'installation de Grafana	138
4.3	Mécanismes de secours	140
4.4	Consultation de l'interface Grafana	144

Chapitre 6

Fonctionnement d'un playbook

1.	Objectifs du chapitre et prérequis	147
1.1	Contexte et prérequis.	147
1.2	Fichiers téléchargeables	148
2.	Le moteur de template Jinja : principe de fonctionnement	148
3.	Template Jinja	152
3.1	Mécanisme de boucle sur tableau avec Jinja.	153
3.2	À la recherche de la bonne variable Ansible	154
3.3	Affichage des interfaces réseau	155

- 4. Délégation de tâche 156
 - 4.1 Changement du type de connexion au niveau d'une tâche. . . 157
 - 4.2 Délégation d'une tâche. 158
 - 4.3 Tâche à exécuter une seule fois 161
- 5. Gestion d'un serveur Apache 162
- 6. Réduction des opérations impactantes 165
 - 6.1 Mécanisme des tags 165
 - 6.1.1 Déclaration d'un tag 165
 - 6.1.2 Comment lister les tags d'un playbook ? 166
 - 6.1.3 Sélection ou exclusion d'un tag 166
 - 6.1.4 Gather facts et le tag always 167
 - 6.2 Utilisation d'une variable sur une tâche 168
 - 6.3 Utilisation d'un handler 172
 - 6.3.1 Pourquoi utiliser un handler ? 172
 - 6.3.2 Déclaration d'un handler 173
 - 6.3.3 Exemple de lancement 174

Chapitre 7
Introduction à la notion de rôle

- 1. Objectifs du chapitre et prérequis 177
 - 1.1 Contexte et prérequis. 177
 - 1.2 Fichiers téléchargeables 177
- 2. Introduction aux rôles Ansible 178
 - 2.1 Structure d'un rôle 178
 - 2.2 Votre premier rôle : installation d'Apache 179
 - 2.3 Configuration du serveur Apache 181
- 3. Installation de MediaWiki 183
 - 3.1 Filtres Jinja et installation d'Apache 183
 - 3.2 Gestion des variables dans un rôle. 188
 - 3.3 Installation de MariaDB/MySQL 190
 - 3.4 Installation de MediaWiki 192

3.5	Configuration de MediaWiki	196
3.5.1	Décompression de l'archive	196
3.5.2	Mutualisation de la base MariaDB et du serveur Apache (SELinux)	198
3.5.3	Commande de configuration de MediaWiki	199
3.5.4	Dépendances et variables manquantes	200
3.5.5	Lancement de l'installation et configuration de MediaWiki	203
3.5.6	Changement de version de PHP	205
3.5.7	Boucle et affectation des accès distants vers la base MariaDB	208
4.	Finalisation de l'installation de MediaWiki	212
4.1	Réentrance et script shell	212
4.2	Problème de redirection	214

Chapitre 8

Playbooks, rôles et notions avancées

1.	Objectifs du chapitre et prérequis	215
1.1	Contexte et prérequis	215
1.2	Fichiers téléchargeables	216
2.	Gestion de Python sur les machines distantes	216
2.1	Introduction	216
2.2	Préparation de l'inventaire	217
2.3	Gestion d'une machine sans Python	218
2.4	Détection automatique de l'interpréteur	220
2.5	Positionnement de l'emplacement de l'interpréteur Python	221
2.6	Python3 et Ansible	223
3.	Scalabilité et répartition de charge	223
3.1	Contexte	223
3.2	Origine du besoin	224
3.3	Ajout de nouveaux serveurs Apache	225
3.4	Lancement d'une opération de manière séquentielle	227

- 4. Ajout d'un répartiteur de charge 230
 - 4.1 Pourquoi ajouter un répartiteur de charge ? 230
 - 4.2 Différence entre Ubuntu/Debian et Centos/Red Hat 231
 - 4.3 Installation du serveur Haproxy 231
 - 4.3.1 Configuration du serveur Haproxy 232
 - 4.3.2 Mise au point du rôle 235
- 5. Quelques bonnes pratiques à suivre 239
 - 5.1 Contrôler le lancement des handlers 239
 - 5.2 Réduire le temps d'indisponibilité d'un service 241
- 6. Mise à jour et réentrance de script 241
 - 6.1 Contexte 241
 - 6.2 Gestion de la montée de version du schéma 242
- 7. Mise à jour par roulement (rolling update) 245
 - 7.1 Contexte 245
 - 7.2 Présentation du mécanisme 246
 - 7.3 Rolling update sur MediaWiki 247
 - 7.4 Désactivation des machines avant mise à jour 252
 - 7.4.1 Configuration du Haproxy 253
 - 7.4.2 Désactivation de l'instance Apache avant mise à jour 255
 - 7.4.3 Utilisation de l'instruction include_role 260
 - 7.4.4 Pollution de la trace Apache 261
- 8. Inclusion et réutilisation 264
 - 8.1 Contexte 264
 - 8.2 Inclusion statique de tâches 265
 - 8.3 Inclusion dynamique de tâches 266
 - 8.4 Gestion du comportement de l'instruction de boucle 267
 - 8.4.1 Contrôle du nom de la variable de boucle 267
 - 8.4.2 Autres mots-clés intéressants 268
- 9. Ansible Galaxy 269
 - 9.1 Présentation du site 269
 - 9.2 Recherche d'un rôle 271
 - 9.3 Utilisation d'un rôle en provenance d'Ansible Galaxy 273

9.4	Utilisation d'un fichier de prérequis	274
9.5	Comment lister les rôles d'Ansible Galaxy ?	275
9.6	Suppression d'un rôle	275
9.7	Collections Ansible	276
9.7.1	Origine du besoin	276
9.7.2	Recherche de collections	277
9.7.3	Installation d'une collection	277
9.7.4	Consultation de la liste des collections	278
9.7.5	Utilisation d'une version spécifique	278
9.7.6	Utilisation d'un fichier de dépendances	279
9.7.7	Gestion des datasources Grafana	280

Chapitre 9

Stratégie d'exécution et optimisation

1.	Objectifs du chapitre et prérequis	283
2.	Stratégie d'exécution	283
2.1	Contexte	283
2.2	Gestion du nombre de tâches lancées en parallèle	284
2.3	Stratégie d'exécution : free	284
3.	Débogage avec Ansible	288
3.1	Contexte	288
3.2	Consultation d'informations	290
3.3	Modification d'un élément	290
3.4	Relance d'une tâche	291
3.5	Définition d'une variable	292
4.	Étude du fonctionnement d'Ansible	293
4.1	Contexte	293
4.2	Playbook de test	293
4.3	Opérations lancées par Ansible	295

- 5. Présentation de Mitogen 297
 - 5.1 Contexte 297
 - 5.2 Présentation de Mitogen 298
 - 5.3 Principe de fonctionnement de Mitogen. 298
 - 5.4 Activation de Mitogen 299
 - 5.5 Gains potentiels et limitations 300

Chapitre 10
Tests Ansible

- 1. Objectifs du chapitre et prérequis 301
- 2. Linter Ansible 301
 - 2.1 Contexte 301
 - 2.2 Lancement de l'analyse 302
 - 2.3 Règles disponibles 302
 - 2.4 Désactivation de règles 303
 - 2.5 Création de nouvelles règles 304
- 3. Présentation de Molecule 306
 - 3.1 Contexte 306
 - 3.2 Prérequis et installation de Molecule 306
 - 3.3 Installation de Docker 307
 - 3.3.1 Installation de Docker Community Edition sur Ubuntu 307
 - 3.3.2 Installation alternative 308
 - 3.3.3 Configuration des accès à Docker 308
 - 3.3.4 Vérification de l'installation de Docker 309
- 4. Utilisation de Molecule 310
 - 4.1 Fichier de configuration molecule.yml 310
 - 4.2 Création d'un rôle Apache 310
 - 4.2.1 Initialisation du rôle 310
 - 4.2.2 Structure du rôle 311
 - 4.2.3 Intégration du rôle Apache 312
 - 4.2.4 Playbook de convergence 312

4.2.5	Vérification de l'installation	313
4.3	Lancement des tests	314
4.4	Débogage du problème	317
4.4.1	Lancement de Molecule en mode debug	317
4.4.2	Connexion au container de tests	317
4.5	Configuration pour le fonctionnement avec System D	319
4.5.1	Modification de la définition de la plateforme (champ platforms)	319
4.5.2	Relance des tests	319
4.6	Gestion des scénarios de Molecule	320

Chapitre 11

Sortie Ansible et centralisation

1.	Objectifs du chapitre et prérequis	323
1.1	Contexte et prérequis	323
1.2	Fichiers téléchargeables	324
2.	Gestion de la sortie standard d'Ansible	324
2.1	Contexte	324
2.2	Fichier de configuration et cowsay	324
2.3	Gestion de la colorisation de la sortie	327
3.	Gestion du callback d'affichage	329
3.1	Contexte	329
3.2	Quelques plugins d'affichage alternatifs	330
3.3	Profiling des opérations	331
4.	Centralisation des résultats d'exécution	333
4.1	Contexte	333
4.2	Centralisation via Syslog	334
4.3	Centralisation via Logstash (Elasticsearch)	335
4.4	Centralisation des appels à l'aide d'Ara	339
4.4.1	Origine de la création d'Ara	339
4.4.2	Installation d'Ara	340
4.4.3	Activation d'Ara avec Ansible	340

- 4.4.4 Test de lancement 340
- 4.4.5 Lancement du serveur de consultation Ara 341
- 5. Écriture de son propre callback d’affichage 343
 - 5.1 Contexte 343
 - 5.2 Structure du programme 343
 - 5.3 Exposition du callback à Ansible 344
 - 5.4 Nomenclature des fonctions de callback 346
 - 5.5 Exemple de surcharge de la sortie en cas d’erreur 347

Chapitre 12
Écriture de modules

- 1. Objectifs du chapitre et prérequis 351
 - 1.1 Contexte et prérequis 351
 - 1.2 Fichiers téléchargeables 352
- 2. Mécanisme d’appel des modules Python 352
 - 2.1 Principe de fonctionnement 352
 - 2.2 Activation des traces d’exécution 353
 - 2.3 Contenu du fichier temporaire 354
 - 2.4 Extraction du programme autoporteur 355
- 3. Pourquoi écrire un module ? 356
 - 3.1 Contexte 356
 - 3.2 Consultation du schéma 357
- 4. Création d’un module 359
 - 4.1 Emplacement du programme 359
 - 4.2 En-tête du programme 360
 - 4.3 Spécification des arguments du module 360
 - 4.4 Gestion des arguments du module 361
 - 4.5 Récupération de la valeur des arguments 362
 - 4.6 Connexion à la base 363
 - 4.7 Transmission du résultat à Ansible 364
 - 4.8 Test du module 365

4.9	Temps d'exécution	366
4.10	Gestion de l'aide	367
4.10.1	Variable DOCUMENTATION	368
4.10.2	Variable EXAMPLES	369
4.10.3	Variable RETURN	369
4.10.4	Variable ANSIBLE METADATA	370
4.10.5	Test d'affichage	370
4.11	Cas particulier des modules facts	371
5.	Réentrance sur la création d'un schéma de base	373
5.1	Contexte	373
5.2	Limitations du module	374
5.3	Dépendances Python	374
5.4	En-tête du module	374
5.5	Gestion de la base de données	375
5.6	Python 2 et 3	376
5.7	Algorithme du module	378
5.8	Création d'un playbook de test	379
6.	Gestion de l'option diff d'Ansible	382
6.1	Contexte	382
6.2	Principe de fonctionnement du diff	382
6.3	Implémentation du diff	383
7.	Gestion de l'option test (check mode)	385
7.1	Activation du check mode	385
7.2	Prise en charge du mode check	386

Chapitre 13

Écriture de filtres Jinja et mécanisme de lookup

1.	Objectifs du chapitre et prérequis	389
1.1	Contexte et prérequis	389
1.2	Fichiers téléchargeables	390

2. Les filtres Jinja	390
2.1 Retour sur les filtres Jinja	390
2.2 Quelques exemples d'utilisation des filtres Jinja	391
2.2.1 Comment tester un filtre ?	391
2.2.2 Valeur par défaut	391
2.2.3 Conversion de type simple	391
2.2.4 Chargement de YAML/JSON	392
2.2.5 Gestion de listes	393
2.2.6 Calcul de somme de hachage	394
2.2.7 Combinaison de table de hachage	395
3. Test de comparaison de mots de passe salés	396
3.1 Quelques mots sur le stockage des mots de passe	396
3.2 Réentrance sur un mot de passe	397
3.3 Récupération du hash du mot de passe	398
3.4 Écriture du filtre de comparaison	399
3.4.1 Emplacement du programme	399
3.4.2 Fonction de comparaison des mots de passe	400
3.4.3 Mapping Ansible	400
3.4.4 Tests du filtre	401
3.5 Utilisation du filtre dans le playbook	403
4. Génération de mot de passe	405
4.1 Origine du besoin	405
4.2 Principe de fonctionnement du générateur de mots de passe	405
4.2.1 Écriture du générateur de mots de passe	406
4.3 Playbook de test	407
5. Récupération d'informations (lookup/query)	408
5.1 Principe de fonctionnement	408
5.2 Présentation des lookups Ansible natifs	409
5.2.1 Lecture de fichiers : file	409
5.2.2 Récupération du résultat d'un template	409
5.2.3 Réalisation d'une requête DNS : dig	410
5.3 Le mécanisme de lookup et les boucles	410

6. Écriture de son propre lookup : KeePass	412
6.1 Contexte	412
6.2 Dépendance Python	412
6.3 Récupération du mot de passe	412
6.4 Intégration avec Ansible	414
6.4.1 Emplacement du fichier Python	414
6.4.2 Exposition de l'objet lookup	414
6.4.3 Code du plugin	415
6.5 Test du plugin	415
6.5.1 Contenu base KeePass et présentation de KeePassX . .	416
6.5.2 Création du test	416

Chapitre 14

Les actions Ansible

1. Objectifs du chapitre et prérequis	419
1.1 Contexte et prérequis	419
1.2 Fichiers téléchargeables	419
2. Découverte des actions Ansible	420
2.1 Fonctionnement des actions Ansible	420
2.2 Certificats SSL avec Java	420
2.3 Importation d'un certificat avec Ansible	421
3. Création du module d'action java_cert	424
3.1 Emplacement du fichier d'action	424
3.2 Structure du module d'action java_cert	425
3.3 En-tête du module	426
3.4 Contenu de la méthode run	426
3.5 Test du module action	428

Chapitre 15

Ansible : virtualisation et cloud

- 1. Objectifs du chapitre et prérequis 431
 - 1.1 Contexte et prérequis 431
 - 1.2 Fichiers téléchargeables 431
- 2. Gestion de machines virtuelles sous ESX/VMware 432
 - 2.1 Contexte et prérequis 432
 - 2.2 Installation des bibliothèques requises 432
 - 2.3 Caractéristiques de l'ESX 432
 - 2.4 Création de la VM 433
 - 2.5 Customisation des machines 435
 - 2.6 Gestion du démarrage de la machine 437
 - 2.7 Suppression de la VM 440
- 3. Gestion cloud AWS 441
 - 3.1 Contexte 441
 - 3.2 Configuration des accès à l'API Amazon 443
 - 3.3 Création d'une instance EC2 448
 - 3.4 Comment taguer ses instances ? 451
 - 3.5 Suppression d'une instance 453
 - 3.6 Accès réseau aux machines 454
 - 3.7 Configuration de la clé SSH 456
 - 3.8 Modification de la déclaration de l'instance ec2 457

Chapitre 16

Tester Ansible avec Docker

- 1. Objectifs du chapitre et prérequis 461
 - 1.1 Contexte et prérequis 461
 - 1.2 Fichiers téléchargeables 462
- 2. Ansible, Docker et Systemd 462
 - 2.1 Comment faire cohabiter Docker et Systemd ? 463
 - 2.2 Choix des images à lancer 463

2.3	Alimentation du fichier d'inventaire	464
2.4	Playbook de création des conteneurs	465
2.5	Playbook de suppression des conteneurs	468
3.	Les images Docker	469
3.1	Présence de l'interpréteur Python	469
3.2	Pilotage de Docker à l'aide d'Ansible	470
3.2.1	Avant propos	470
3.2.2	Idempotence et immutabilité	470
3.2.3	Image à déployer	471
3.2.4	Compilation de l'image Docker	472
3.2.5	Utilisation de l'image Docker	473

Chapitre 17

Pilotage de Kubernetes à l'aide d'Ansible

1.	Objectifs du chapitre et prérequis	477
2.	Introduction au déploiement d'éléments dans Kubernetes	478
2.1	Containers et pods	478
2.2	Mécanismes de déploiement dans Kubernetes	478
2.3	Déploiement de pods dans Kubernetes	478
2.3.1	À propos de Keycloak	478
2.3.2	Configuration de Keycloak	479
2.3.3	Définition d'un objet StatefulSet	479
2.3.4	Création d'une entrée de service	480
2.3.5	Playbook de déploiement	481
2.3.6	Test du playbook de déploiement	482
2.4	Création d'un opérateur pour Keycloak	483
2.4.1	Contexte	483
2.4.2	Installation des prérequis	483
2.4.3	Initialisation de l'opérateur	484
2.4.4	Description de la structure de l'opérateur	485

- 2.5 Création de l'opérateur 486
 - 2.5.1 Patron de l'objet StatefulSet 486
 - 2.5.2 Variables par défaut 488
 - 2.5.3 Patron de l'objet Service 488
 - 2.5.4 Création des objets 489
- 3. Test de l'opérateur 489
 - 3.1 Contexte 489
 - 3.2 Définition du nouveau type d'objet 490
 - 3.3 Installation des prérequis 491
 - 3.4 Création d'un playbook de test 492
 - 3.5 Vérification et suppression des objets de test 492
 - 3.6 Lancement de l'opérateur depuis l'extérieur du cluster 493
 - 3.7 Test de création d'un objet Keycloak 494
 - 3.8 Vérification des objets créés 495
- 4. Déploiement de l'opérateur 497
 - 4.1 Contexte 497
 - 4.2 Configuration du registre Docker 497
 - 4.3 Construction de l'image de l'opérateur 498
 - 4.4 Déploiement de l'opérateur 499
 - 4.5 Vérification de l'installation 502
 - 4.6 Test de l'opérateur 503
 - 4.7 Journaux d'activité de l'opérateur 503
- Index 505

